Internal transport control in pot plant production

Promotor: Dr. P. van Beek
Hoogleraar in de Operationele Analyse

Co-promotoren: Dr. Ing. H.W.J. Donkers
Senior wetenschappelijk onderzoeker, IMAG-DLO
Dr. Ir. D. Goense
Hoofd van de afdeling Technologie Open Teelten, IMAG-DLO

Engelbertus Annevelink

Internal transport control in pot plant production

Proefschrift

ter verkrijging van de graad van doctor
op gezag van de rector magnificus
van de Landbouwuniversiteit Wageningen,
dr. C.M. Karssen,
in het openbaar te verdedigen
op maandag 28 juni 1999
des namiddags te 13.30 uur in de Aula.

This research was carried out at the DLO-Institute of Agricultural and Environmental Engineering (IMAG-DLO) in Wageningen. It was initiated under the Research Programme 'Farm development and effects of changes in technique and environment for farms and agricultural sectors' and was financed by the Ministry of Agriculture, Nature Management and Fisheries. This contribution is gratefully acknowledged.

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Annevelink, E.

Internal transport control in pot plant production / E. Annevelink - [S.I. : s.n.]
Thesis Wageningen Agricultural University. - With ref. - With summary in Dutch and English
ISBN 90-5808-059-5

Subject headings: operational planning, simulation, sequencing, simulated annealing, genetic algorithm

Bibliographic Abstract:
Drawing up internal transport schedules in pot plant production is a very complex task. Scheduling internal transport at the operational level and providing control on a day-to-day or even hour-to-hour basis in particular requires a new approach. A hierarchical planning approach based on Anthony's framework was presented in this study to decompose the internal transport planning problem of pot plant nurseries into three planning levels with differentiating characteristics: strategic, tactical and operational planning. Emphasis was put on the operational planning level within the constraints of the tactical and strategic planning level. A discrete transport simulation model TRANSIM was designed and developed to analyse and evaluate the rules of thumb on internal transport used by growers in everyday life. The operational planning problem with respect to internal transport was divided into three sub-problems that deal with planning the work sequence, internal transport sequence and parking positions. Four local search methods - Simulated Annealing (SA), Genetic Algorithm (GA), Tabu Search (TS) and Random Search (RS) - were described and applied to generate good internal transport sequences in combination with a chosen parking method. The performance results of the local search methods applied to real-scale test cases show that Simulated Annealing is the most effective and efficient method of finding good internal transport sequences in combination with a relatively simple parking method.

This thesis is also available as publication No. 99-03, ISBN 90-5406-175-8 of the DLO-Institute of Agricultural and Environmental Engineering (IMAG-DLO), P.O. Box 43, NL-6700 AA, Wageningen, The Netherlands

imag-dlo

Voor mijn ouders,
Sietske, Jan Dirk en Annelies

Voorwoord

Het schrijven van een proefschrift is weliswaar een eenzame activiteit, maar het is ook een onderneming die onmogelijk is zonder de hulp van een groot aantal mensen. Hun ondersteuning op het inhoudelijke en persoonlijke vlak is onmisbaar tijdens het tot stand komen van een boekwerk, zoals dat nu voor u ligt. Al deze mensen verdienen daarom stuk voor stuk een persoonlijk woord van dank voor hun specifieke bijdrage.

Allereerst wil ik mijn promotor Paul van Beek bedanken voor de prettige wijze waarop hij mij al die tijd heeft begeleid. Tijdens onze werkbesprekingen stelde hij strategische vragen die mij vaak op nieuwe sporen wezen. Een speciale rol vervulde Theo Hendriks als mijn dagelijks begeleider vanuit de vakgroep wiskunde. Vanuit zijn ruime ervaring op het gebied van planning in de glastuinbouw wist hij steeds opbouwende kritische vragen te stellen. Zijn opmerkingen stimuleerden mij steeds weer tot nieuwe gedachten en inzichten, waarvoor ik hem dankbaar ben. Mijn co-promotoren vanuit IMAG-DLO, Harry Donkers en Daan Goense, wil ik bedanken voor hun gedegen inhoudelijke ondersteuning en voor het in mij gestelde vertrouwen, dat ik dit project ooit nog wel eens een keer zou afronden. De directie van IMAG-DLO ben ik erkentelijk voor de mij geboden kans en voor alle ondersteuning die ik vanuit mijn werkomgeving heb mogen ontvangen. Jan Achten stimuleerde mij destijds om te beginnen aan dit promotie-onderzoek. Harry Donkers ondersteunde ons voorstel en zo kon ik in 1994 starten met mijn project. In de beginperiode leverde Annet Vink een zeer belangrijke bijdrage aan de ontwikkeling van het simulatiemodel TRANSIM. Het is aan haar inzet te danken, dat de animatie van het model zijn huidige, visueel aantrekkelijke vorm heeft bereikt. Op statistisch gebied heb ik waardevolle adviezen en hulp gekregen van onze huis-statistici Valentijn van den Berg en Margriet Hendriks, waarvoor mijn dank. Gedurende de afgelopen jaren mocht ik veelvuldig gebruik maken van de rekencapaciteit van de PC’s van mijn afdelingsgenoten Jan Achten, Gerrit Kroeze, Kees Lokhorst, Rudi de Mol, Annet Vink en Cor Wildenberg. Dit heeft mij snel verder geholpen en ik hoop maar dat ik jullie niet te veel overlast heb bezorgd.

Voor mijn onderzoek heb ik gebruik kunnen maken van geregistreerde gegevens van Kwekerij de Goede Hoop te Honselersdijk. Hiervoor wil ik Jos van der Knaap en Erik van der Voort hartelijk bedanken. Zij stonden altijd klaar om mijn vragen te beantwoorden en om mij steeds de meest recente gegevensbestanden toe te sturen. Erik heeft mij bovendien geholpen bij de analyse van de bedrijfsgegevens en hij speelde ook een belangrijke rol bij de oriënterende bedrijfsbezoeken in de eerste fase van mijn onderzoek. Op deze plaats wil ik ook de contactpersonen van de bezochte potplantenbedrijven nogmaals danken voor hun medewerking.

Niet onvermeld mag blijven, dat ook een aantal studenten een belangrijke bijdrage heeft geleverd aan mijn project. Hierbij wil ik Jan-Dries Luijks, Wilko van der Hoorn, Hans Rosier, Joost Huijbers en Anke Janssen bedanken voor hun waardevolle inbreng.

De inspiratie voor de keuze van mijn onderzoek ligt bij een aantal personen. Ton Saedt, destijds werkzaam bij IMAG-DLO, dank ik speciaal voor het feit dat hij mij heeft binnengeleid in de planningsproblematiek van glastuinbouwbedrijven. Peter van Weel is degene geweest die mij bij IMAG-DLO op het spoor heeft gezet van intern transport, waarvoor ik hem nog steeds dankbaar ben. Rob Broekmeulen, destijds werkzaam bij de vakgroep wiskunde, heeft mij als eerste in aanraking gebracht met lokale zoekmethoden en speciaal met het genetisch algoritme. Hij deed dit met veel enthousiasme en verwees mij ook door naar professor Aarts van de TU-Eindhoven. Deze gaf mij zeer waardevolle adviezen op het gebied van experimenteren met lokale zoekmethoden.

Bij de afronding van het proefschrift heeft Marilyn Minderhoud-Jones een belangrijke rol gespeeld door het Engelse correctiewerk voor haar rekening te nemen. Zij heeft vele uren van haar tijd opgeofferd om mijn kromme Engels weer recht te schrijven. Pieter Bosveld dank ik voor het werpen van een kritische blik op mijn Nederlandse samenvatting.

Mijn ouders wil ik bedanken voor de mogelijkheid, die ze me hebben geboden om te studeren wat ik leuk vond en voor alle steun die ze mij altijd hebben gegeven.

Sietske, tenslotte dank ik jou voor al het geduld en optimisme, dat je steeds hebt weten op te brengen in de soms moeilijke tijden gedurende de afgelopen jaren. Vaak heb je waarschijnlijk het gevoel gehad, dat ik nooit meer van onze zolder naar beneden zou komen. Misschien dat je daarom soms opperde, dat we er ooit twee kamers voor Jan Dirk en Annelies van zouden kunnen maken. Nu ziet het er dan eindelijk naar uit, dat we die plannen kunnen gaan realiseren.

Contents

1. Introduction

1.1 Problem description

The horticultural production process requires internal transport to bring products from one position in the company to another. Internal transport is especially important in pot plant nurseries. At the moment the amount of internal transport needed in pot plant nurseries is substantial and it will continue to grow in volume as developments internal and external to the industry continue to take place.

External developments that relate to the pot plant nursery include the change from a production driven push-market towards a customer driven pull-market and the necessity for individual pot plant nurseries to operate more and more as a part of a larger production chain (Van Oosten, 1998). Changing customer preferences and the growing purchasing power of the retailers have made growers realize that a customer-oriented attitude is necessary if they are to survive competition. The trend towards mass individualization implies that each customer will try to satisfy his or her individual needs. This has far-reaching implications for the way growers think about production. A shift has to be made from less expensive and standardized products towards custom-made products. However, these custom-made products will still have to be produced within an advanced and almost industrial production process, which will have to be highly flexible in order to cope with many different types of customer demands. Rapidly changing market specifications will determine which product types have to be produced by the grower. These external developments require a very flexible and responsive pot plant production system.

The production plan for a pot plant nursery usually involves a large number of crop batches each year. The fact that these crop batches have to be transported several times during their growing process makes the every day business of internal transport a complex and time-consuming process in most pot plant nurseries. Technical developments within the pot plant nursery itself makes the production and internal transport processes more complex but technical developments also offer possibilities and ways in which these processes can be supported. New Information and Communication Technology (ICT) allow better and faster communication with clients (Meuleman & Van Weel, 1997). Information and product flows can be controlled more efficiently with ICT. Data control techniques have a huge potential and can be of considerable benefit to crop management and product quality management. ICT can help generate more and better data on the internal transport process and has the potential to create more information for executing controlling tasks. Another internal development is the use of vision systems for the automated and easy identification of different quality classes. This results in smaller pot plant production units - with their specific characteristics - having to be traced and transported through the pot plant nursery. Developments in robot or related technology make it possible to treat small production units separately. These robots or automated machines require instructions from an electronic planning system. Another development is the continuous growth in the size of enterprises and the consequent need for more internal transport. According to Meuleman & Van Weel (1997), the conflict between the call for greater product differentiation (consumer pull) and the necessity of economy of scale in the form of larger production facilities can only be solved by developing flexible, modular systems for mechanization and automation.

How to control internal transport in pot plant nurseries in an effective and efficient way becomes an important operational planning problem in the face of these external and internal developments is. The control of internal transport is associated with frequent decision-making about work sequences, the internal transport sequences and the position of pot plants in the growing area. Effective internal transport means that crop batches arrive at the correct position at the right time, and efficient internal transport means that this is done with the least possible effort, that is with the least amount of transport movements over the shortest possible transport distance. Drawing up internal transport sequences in the context of pot plant production is a very complex task. Therefore, it is essential to study and develop new approaches and methods that can support the grower during the internal transport planning process. Sequencing internal transport at the operational level and providing control on a day-to-day or even hour-to-hour basis requires a new approach.

Internal transport control is an extremely complex problem and this is the main reason why there is no general, efficient technique that can offer a solution to this problem at the moment. Growers therefore are accustomed to schedule by ‘rule of thumb’ and in general do not develop exact internal transport sequences. Rules of thumb can provide a valuable basis for finding new ways of supporting the control process, because they can be used to guide heuristic search methods. Issues of combinatorial explosion (see Section 6.1) arise when real-scale problems have to be solved. The growing complexity of the control process and the explosive growth of the data available demand the development of new control systems that are based on operational research techniques, local search methods and Information and Communication Technology. Growers with automated internal transport devices and automated data recording systems are especially interested in a new control approach, because each day they confront the difficult problem of sequencing internal transport.

With the aid of a new control approach they will be able to increase the flexibility of the production process a necessary step in being able to satisfy fluctuating market demand.

This present research not only attempts to find new ways of supporting the internal transport control process it is also concerned with developing a new control approach. If a suitable support method is available growers will be able to control complex internal transport processes more effectively and efficiently, and be able to avoid bottlenecks in the production process. Internal transport devices will not become overloaded anymore during peak periods and, as a result the speed of harvesting and delivery to auctions and customers will be increased. This will enable the grower to react more responsively to market demand. A grower will also be able to use the control of internal transport to avoid incorrect allocation of pot plants to positions where sub-optimal growing conditions will lead to loss of quality, the possible waste of fertilizer and insufficient crop protection measures.

1.2 Design, planning and control of internal transport

The internal transport planning process is very complex. Therefore, some decomposition of the problem into sub-problems is necessary. Hax & Candea (1984) describe Anthony's framework in which Anthony classified decisions into three categories: strategic planning, tactical planning (or management control) and operational control. Anthony's framework is a hierarchical integrative planning approach. A hierarchical planning approach enables the grower to divide the internal transport planning problem into sub-problems, and thus lower the complexity of the planning process. The sub-problems can be solved separately, but they impose constraints on each other and these have to be taken into account. Activities at the three decision levels are defined in the terminology of Slack et al. (1998; Figure 3-1), as design, planning and control. Design, planning and control are needed to maximize effectiveness (achieve all targets) and to maximize efficiency (minimize production costs).

At the *strategic level* internal transport will be influenced for many years by the following *design* factors: the production system; the layout of the pot plant nursery; and internal transport devices. One *production system* that involves considerable internal transport is the transportable bench production system. Pot plants stand on transportable benches and these can be moved throughout the nursery. Transportable benches offer good opportunities for mechanizing the operations involved in pot plant cultivation. The *layout of a pot plant nursery* (Figure 2-1, 2-4 and 2-6) influences the accessibility of transportable benches in the rows. Poor accessibility leads to more internal transport movements. A growing number of pot plant nurseries have a separate working area, where the potting, sorting, spacing and harvesting operations are performed on pot plants by mechanized work stations. Separating the work involved in pot plant cultivation from the growing process itself leads to better working conditions in the working area and to a greater degree of space utilization in the growing compartments, than is the case in more traditional systems. However, it also increased the amount of internal transport needed between the growing area and the working area. The introduction of new *internal transport devices* such as Automatic Guided Vehicles (AGVs) in combination with the transportable bench production system has made internal transport less labour intensive and has lowered operating costs. However, it has also increased investment levels with respect to internal transport devices (hardware), systems for collecting and handling data on the transport process and systems to plan and control the movements of the AGV (software). An AGV has to be given transport assignments which requires additional labour. The manager must perform difficult control operations at the operational planning level.

Each crop of pot plants has its own standard production process with specific operations that give rise to internal transport. A crop with a strongly varying space requirement pattern will need more internal transport, than a crop whose space requirements remain unchanged throughout the whole growing period. On the *tactical level*, internal transport can be influenced for many months by the *production plan* developed. In this plan the grower makes decisions about the exact size of crop batches (specified amount of a specified crop) within the constraints set at the strategic planning level. The crop batches chosen determine the amount of internal transport required.

On the *operational level* control decisions are made on subjects that influence internal transport in a pot plant nursery in the short term. The internal transport control problem of a pot plant nursery consists of determining all transport movements on the operational planning level. The main control problem can be divided into three sub-problems (Figure 5-2). These deal with determining the *work sequence*; the *internal transport sequence*; and the *parking positions*. It is important to minimize the total number of transport movements, because each transport movement requires time and consequently costs money. The total number of transport movements can be influenced by choosing a good internal transport sequence. The choice of the exact rows and positions for parking transportable benches can also influence the amount of internal transport.

Decisions on the operational planning level will be influenced by decisions at the tactical and strategic planning level. The work sequence is directly influenced by the contents of the production plan, which in turn is influenced by the available size of the production system chosen at the strategic planning level. The *interaction* between these different planning levels has to be taken into account to enable an optimal integration of decisions made at the various planning levels. These levels will have to signal each other when decisions at one planning level lead to a non-feasible situation at another. A common solution can then be found to discrepancies between the planning levels. The work sequence, internal transport sequence and parking positions are interrelated at the operational planning level, and will, therefore, have to be adjusted to each other. Relations also exist between the internal transport sequence and other schedules that are not directly concerned with internal transport such as the climate control and the watering schedule.

1.3 Objectives of this study

The emphasis of this study is the operational planning level. At this level the *work sequence* determines the order of operations (potting, sorting, spacing and harvesting), that must be performed on certain crop batches during a specific period of time. These operations lead to the internal transport of transportable benches. These have to be collected from the growing area and brought to the working area. The *internal transport sequence* determines which transportable benches have to be transported and in what order. Finally, the current and future *positions of transportable benches* have a considerable influence on the amount of internal transport that can be expected. Different crops have different requirements as far as the climate, light, nutrients, water consumption and pest control required at their position in the growing compartment are concerned. This restricts the number of positions suitable for a transportable bench carrying a particular crop batch.

Relevant data about internal transport have to be recorded frequently at the operational planning level in order to support the control process. Recently, automatic data recording systems with sensors have become available and these can be installed on work stations and internal transport devices. These systems can replace manual data collection to a large extent. Collected data contain information valuable for the control process providing information about the exact space allocation of the growing area, the exact position of transportable benches, and the status of sequences being implemented. All this information is needed to generate new or modified sequences. Considering the complexity of the control problem and the high degree of uncertainty about the near future, it is very difficult to determine optimal sequences at the operational planning level. In most cases it will be sufficient to find an acceptable sequence within the given constraints. The short-term character of the operational planning process forces the manager to make frequent and rapid choices throughout the day. Once a sequence has been chosen, it has to be implemented immediately and it has to be monitored to decide when it has to be adjusted.

The main research objective was to find methods capable of supporting solutions to the problem of internal transport control at the operational planning level in pot plant nurseries using transportable benches. Strategic and tactical planning were only studied to determine their interaction with the operational planning level. The possibilities of simulation techniques and local search methods to support the grower during the control process at the operational planning level were also studied.

1.4 Overview of thesis structure

This thesis has the following structure. Chapter Two contains a detailed description of internal transport systems in pot plant nurseries. Chapter Three explains the different planning levels, that influence internal transport. An internal transport simulation model is presented in Chapter Four and this simulation model is used to discover and evaluate what are the best rules of thumb being used by growers today to solve the problem of internal transport control. Simulation experiments were performed using practical data taken from a pot plant nursery. Chapter Five describes a simplified internal transport control problem. A general framework is presented in which selected rules of thumb for control can be combined with methods for constructing an algorithm which can generate an acceptable solution to a simplified internal transport control problem. In Chapter Six four local search methods, the Genetic Algorithm, Simulated Annealing, Tabu Search and Random Search are described. These can be used for the approximation of good internal transport sequences. The implementation of these local search methods is described and some simple test cases are used to make a pre-selection of these methods. Chapter Seven analyses the performance of the local search methods that were applied to practical real-scale test cases of internal transport control problems. Final conclusions and recommendations are provided in Chapter Eight.

2. Components of internal transport in pot plant nurseries

2.1 Pot plant production and the importance of internal transport

This study will focus on internal transport in pot plant nurseries. The pot plant production of both flowering and green plants is one of the three main types of floricultural production carried out under glass in the Netherlands. The others being cut flowers and bedding plants. The areal development of these main floricultural production types in the Netherlands is given in Table 2-1. These data show that while the area of pot plant production is smaller than cut flower production it continues to grow.

Table 2-1. Area (in ha) with floricultural crops under glass in the Netherlands (LEI-DLO & CBS, 1998).

Year	Total	Cut flowers	Bedding plants	Pot plants flowering	Pot plants green	Others
1980	3 976	2 983	147	282	272	292
1985	4 275	3 221	170	301	385	198
1990	5 140	3 733	218	425	558	205
1995	5 518	3 832	345	561	550	231
1996	5 556	3 855	373	581	554	193
1997	5 541	3 806	422	597	547	170

Pot plant production is a growth sector within floricultural production. Since 1980, the area of both flowering and green pot plant production has more than doubled, although in area it has remained more or less stable in recent years (Table 2-1 and 2-2). The balance between flowering and green pot plants has shifted over the last twenty years. The percentage of green pot plants has increased from 49% in 1980 to a maximum of 57% in the period of 1988-1992. However, since 1993 the percentage of green pot plants gradually fell and, in 1997, it reached 48%. Some companies grow both flowering and green pot plants. In 1990 there were 355 such companies.

Data on the supply and the volume of sales of the ten most important pot plant species at Dutch auctions in the period 1994-1997 (Table 2-3) reflect how important the pot plant production sector is for the Dutch economy. Whilst the volume of sales of three pot plant species (Ficus, Begonia and Yucca) decreased in this period, the volume of sales of three other species (Kalanchoe, Hedera and Chrysanthemum) has increased. The volume of sales of the other four species has remained more or less stable.

Table 2-2. Development of the area of pot plant production under glass and the number of companies in the Netherlands between 1980 and 1997 (KWIN, 1997).

Year	Flowering pot plants			Green pot plants			Total
	ha	%	number	ha	%	number	ha
1980	282	51	1 250	271	49	1 057	553
1985	301	44	1 031	385	56	1 033	686
1986	333	45	1 044	406	55	1 041	739
1987	357	44	1 063	449	56	1 034	806
1988	375	43	1 026	497	57	1 083	872
1989	390	43	1 019	527	57	1 112	917
1990	425	43	1 037	558	57	1 083	983
1991	451	43	1 046	598	57	1 090	1 049
1992	465	43	1 031	618	57	1 041	1 083
1993	510	46	998	590	54	990	1 100
1994	525	46	1 026	613	54	975	1 138
1995	561	50	1 046	550	50	949	1 111
1996	581	51	1 041	554	49	964	1 135
1997	597	52	1 060	547	48	948	1 144

Table 2-3. Supply and volume of sales at Dutch auctions of 10 important pot plant species (KWIN, 1997; LEI-DLO & CBS, 1998). The species are ranked by the volume of sales in 1997.

Species	Supply ($x10^6$ pots)				Volume of sales ($x10^6$ NLG)			
	1994	1995	1996	1997	1994	1995	1996	1997
1. Ficus	33	28	29	29	125	110	110	114
2. Kalanchoe	48	46	46	49	52	52	57	63
3. Dracaena	16	16	15	17	62	58	59	62
4. Hedera	37	50	62	30	37	40	46	50
5. Chrysanthemum	22	26	28	28	38	42	43	46
6. Begonia	22	20	19	18	48	42	42	42
7. Spatiphyllum	10	10	10	12	33	31	32	36
8. Saintpaulia	35	32	31	31	36	33	34	34
9. Poinsettia	13	13	14	17	30	30	33	34
10. Yucca	6	4	4	4	33	28	28	26

Price developments related to the ten most important pot plant species in the period 1990-1996 are given in Table 2-4. Prices show a strong fluctuation over this seven year period. Fluctuating prices have economic consequences for individual pot plant nurseries. The production plan should therefore be flexible enough to allow the choice of economically attractive products. However, given price fluctuations it will be difficult to predict prices when making production plans for the coming year.

Table 2-4. Development of prices at Dutch auctions for 10 important pot plant species as a percentage of the price in 1990 (KWIN, 1997).

Species	1990	1991	1992	1993	1994	1995	1996
1. Ficus	100	92	93	95	90	93	89
2. Kalanchoe	100	115	102	100	102	100	114
3. Dracaena	100	101	103	105	101	94	98
4. Hedera	100	106	99	79	81	65	61
5. Chrysanthemum	100	107	107	100	119	110	106
6. Begonia	100	108	103	109	107	103	108
7. Spatiphyllum	100	86	89	106	92	82	88
8. Saintpaulia	100	110	89	94	103	102	109
9. Poinsettia	100	92	77	81	77	80	78
10. Yucca	100	96	102	105	88	103	100

Internal transport can be defined as the movement of products from one position within a company to another. In the case of pot plant nurseries, the aim is to get plants to the required destination (effectiveness) with the least amount of effort (efficiency). Internal transport consists of loading products, moving them over a certain transport distance and unloading them again. It can also involve relocation, that is moving the products from one transport device to another. Internal transport sometimes also includes the return movements of empty transport devices. Internal transport is an activity that has seen tremendous changes in recent years, including for example, a movement from manual to mechanized and automatic transport (Aldrich & Bartok, 1990; Fang, 1989; Hamrick, 1988; KWIN, 1997).

All floricultural production types require internal transport to move products to their designated positions in the greenhouse. The characteristics of the floricultural production type strongly influence the amount of internal transport required. Pot plant production needs more internal transport than other floricultural types of production because of the specific production process involved (Section 2.5) and because pot plants can be moved relatively easily within the company during and between all growing phases. In vegetable and cut flower production, only harvested products will be transported: producing plants remain in their original position in the greenhouse. Internal transport in cut flower and vegetable production is still in the earlier phases of development. In pot plant production, however, there are many internal transport devices, ranging from the simple to the highly sophisticated. Van Weel (1991) states that the level of mechanization and automation in internal transport and handling is greatest in the pot plant production sector. A pot plant nursery can produce as many as 1 000 different crop batches of pot plants each year. Each crop batch has 1-5 growing phases and must always be transported at the beginning and the end of each growing phase. Sometimes it also has to be moved during the growing phase. This means that internal transport in pot plant production is a complicated and time-consuming business. For this reason the present study has chosen to look at internal transport in pot plant nurseries.

This chapter describes the most significant elements influencing internal transport in pot plant production. Some of the components will be used for a longer period (selected during the strategic planning process) while others change continuously (determined by the operational planning level). At the strategic planning level, a company will decide on the configuration of the internal transport system. These choices include the layout of a pot plant nursery, the type of production system, the type of internal transport devices used and the organization of internal transport processes. The effects of the tactical production plan and the daily internal transport schedule on the amount of internal transport will be discussed in Chapter Three.

Descriptions provided in this chapter are based on a general literature survey combined with a small survey of eleven innovative pot plant nurseries (Appendix 1), conducted by Annevelink & Van der Voort (1995 & 1996). The object of the company survey was to describe the characteristics of the internal transport systems used in pot plant nurseries. The company survey consisted of a visit to each nursery and a study of the specific features of the internal transport system found there. This was followed by an interview with the manager guided by a detailed questionnaire. The aim was to outline each company individually, not to compare them. The pot plant nurseries included in the survey were specially selected because they were highly mechanized innovators. The company survey restricted itself to the transportable bench production system. Because of this, results cannot be considered representative for the pot plant nursery sector as a whole.

2.2 Layout of a pot plant nursery

The *layout* of a pot plant nursery (Figure 2-1) describes the size, position and arrangement of different components. These components include the working area, work stations, buffers, office, rest area, storage area, paths and the growing area with growing compartments and production system. The layout of a pot plant nursery determines internal transport routing and transport distances between different components (Janssen, 1987; Fang et al., 1992).

In modern pot plant production, layout is characterized by a strict separation of the growing area and working area (Van Weel, 1991). These two areas are linked together by paths which can be used by (automatic) internal transport systems and workers. Highly mechanized or fully automated, flexible work stations for all crop handling operations are located in the working area. An extensive data network can be used by computer models and operators for planning and control. A separation between growing compartments and working area was found in almost all of the companies surveyed by Annevelink & Van der Voort (1995 & 1996).

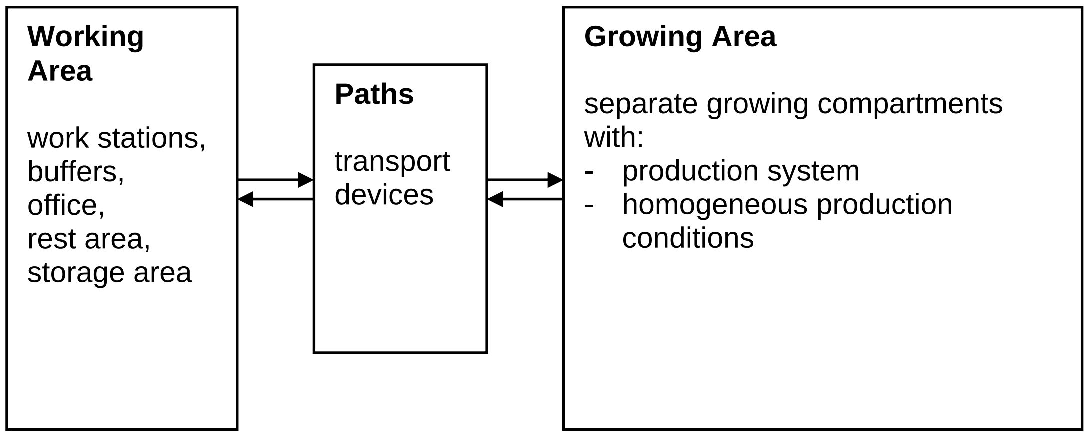


Figure 2-1. Schematic representation of the layout of a pot plant nursery showing the separation of working area and growing area. The connecting paths lie between these two areas.

There are two main reasons for the separation of the growing and working area. First, a separate working area offers better labour conditions for workers in the pot plant nursery. The climatic environment (light, temperature and humidity) in the working area can be adjusted to human requirements. At the same time the conditions in the growing area can be controlled according to the exact requirements of the growing pot plants. A second important advantage is that there are better possibilities for mechanizing pot plant operations. Work stations with a high capacity are needed to increase the flexibility of production. However, these work stations cannot be moved due to their size and therefore they can only be used in a separate area. Examples are highly advanced vision systems (Dijkstra, 1994), which are designed to sort pot plants (Section 2.5.3).

Transportable benches and a uniformity of plants within a crop are preconditions for the separation of processing and growing functions (Van Weel, 1991). Uniformity is needed to achieve a good balance between the costs made to transport a crop to a working area and the advantages of performing specific tasks in that working area. Harvesting work is especially critical in this respect, because it is one of the major tasks and uniformity tends to deteriorate at the end of the production cycle.

A *working area* includes work stations, buffers, an office, rest area and storage area. Input operations start in the working area and output operations are completed there. There are some general factors that have to be taken into consideration in a working area. These include lighting, temperature control, noise control, ease of movement, and the location of restrooms. An agreeable working climate means as little draught as possible, optimal air humidity, equable light (no direct radiation from the sun) and an optimal temperature. Other requirements are optimal working height and as little static load as possible on the employee (Van der Molen, 1992). These conditions can be controlled in a working area so that it gives the most comfortable environment for the employees. Productivity is increased in a separate working area through easier access to materials and equipment in both preparation and shipping (Reilly, 1981). One disadvantage of a separate working area is that pot plants have to be transported to the growing area in the greenhouse (Langhans, 1983). The size of the working area in the Annevelink & Van der Voort survey (1995 & 1996) varied from 450-5 000 m^2 (Appendix 1, Table 4), which was equal to 4-22% of the total area of the growing compartments (Appendix 1, Table 5).

A *work station* is a location where one or more of the operations required during pot plant production (potting, sorting, spacing and harvesting) are performed by one or more machines and by a number of employees. Important characteristics of a work station include: type of operation, capacity, operation time and capacity of buffers. A work station can be located either in the working area or in a growing compartment. This depends on the approach chosen: one can either bring pot plants to a work station or bring the work station to the pot plants. Stoffert & Rohlfing (1983) distinguish three different working positions for performing an operation: a fixed working position in the central working area, a temporarily fixed working position in the main path of a growing compartment or a completely free working position in the growing compartment, for example between benches. Large specialized machines, which cannot be moved can only be used in the working area. Concentrating the work in the working area allows the grower to make the most efficient use of potting machines, transplanting lines and other production equipment (Hamrick, 1988).

Most work stations have *buffers* to store pot plants before or after the operation. Buffers play an important role in the organization of internal transport. They ensure that machines and internal transport devices can be used efficiently. Machines should not have to stand idle because they are without supplies or because they cannot deliver processed pot plants. Required buffer capacity depends on such factors as the processing time involved at the work station and the transport speed of the internal transport device. When an output buffer has been filled completely, a work station cannot be unloaded after finishing the operation. It will, therefore, have to wait until enough pot plants have been removed from the buffer (Janssen, 1987).

The *growing area* in a pot plant nursery consists of greenhouses where the pot plants are placed while they grow. The growing area is often divided into growing compartments. Each *growing compartment* is a separate production location in the pot plant nursery with homogeneous production conditions for plant growth. A growing compartment is further divided into smaller sections such as bays and rows. The total area of the pot plant nurseries surveyed by Annevelink & Van der Voort (1996) varied between 6 300-60 000 m^2. The total area of the growing compartments varied from 5 200-56 000 m^2 (Appendix 1, Table 4). The growing area was divided into a number of growing compartments varying from one to ten (Appendix 1, Table 6). Von Zabeltitz (1986) mentions three possible ways of making optimal use of the growing area. Increasing the production of plants by adopting better growing techniques, speeding up the changing and throughput of plants by more precise production planning and the mechanization of internal transport, and finally enlarging the technical space utilization by reducing the area required for paths.

Paths enable transport devices and workers to access the area. Paths connect the working area with the growing area and growing compartments with each other. The width and the position of paths in a growing compartment depend on their function (main path or aisle between pot plants) and on the transport devices being used. To determine the position and size of the paths, data on the size of the area, transport loads, transport times and type of pot plants are needed (Von Zabeltitz, 1986). Small, long paths are less favourable for transport by hand. A frequently used path should be short if transport is by hand. The area of paths is usually kept as small as possible because, in a glasshouse, ground per square meter is very expensive. However, narrow paths make internal transport more difficult (Reuter, 1989). Aisles have a width of 0.40 to 0.50 m and main paths start at 1 m, depending on the size of the transport device used. Paths through the compartments vary between 2.5 and 3 m in width if they have to carry a vehicle (Von Zabeltitz, 1986). The total area of the paths in Annevelink & Van der Voort's survey (1995 & 1996) was 270-2 150 m^2 (Appendix 1, Table 4), which constituted 2-9% of the total growing compartment area (Appendix 1, Table 5).

A main path can be situated on one or both sides of a growing compartment (Figure 2-4 and 2-6) or it can be positioned somewhere in the middle. The position of the paths in the growing compartments and in the working area is very important if favourable transport possibilities are to be achieved (Von Zabeltitz, 1986). Often one central main path is better than two main paths at the side of the growing compartment. However, paths along the sides of the growing compartment are necessary when different crop batches stand on transportable benches set behind each other in rows (IKC, 1991). The position of the paths determines the accessibility of transportable benches in the rows of a growing compartment. Three access systems are described in Section 2.5.5.

2.3 Production systems

A *production system* is defined in this study as the type of supporting device in a growing compartment on which pot plants are distributed during the growing period. A production system can either be fixed, partly moveable or transportable. The production system strongly influences the amount of handling and internal transport involved. It also influences the possibilities for mechanizing handling and internal transport. The most important production systems available for pot plants at the moment are (Figure 2-2):

- ground (or tempex plates) production system;
- concrete floor production system;
- fixed bench production system;
- rolling bench production system;
- transportable bench production system.

These production systems will be described and compared in this section in terms of their most important characteristics. These include technical space utilization, crop characteristics (size of plants), ergonomic conditions (working height, required reach, climate conditions) and possibilities for the mechanization of operations.

Technical space utilization is defined as the percentage of the total surface of a growing compartment, which can actually be used for growing plants (KWIN, 1997). Technical space utilization is determined on the strategic planning level. Technical space utilization is always less than 100%, because support poles, main paths and aisles, the watering system and heating system also require space in a growing compartment. As far as benches are concerned technical space utilization is the ratio of bench space to the total floor area of the growing compartments expressed as a percentage. This is sometimes called benching efficiency (Langhans, 1983).

Organizational space utilization on the other hand is mainly determined on the tactical planning level by the quality of the production plan. Organizational space utilization gives the ratio between the surface which is actually and fully used for growing crops during each week of the year and the surface which is technically suitable and available for growing crops during each week of the year. Used and available space are given in week-square meters. Organizational space utilization will also be less than 100% because parts of the area which can be technically used for growing pot plants will remain unused in certain periods. Tongeren and Peelen (1984) distinguish two reasons for not fully using this area: transitions between crop batches and selective harvesting.

The description of production systems for pot plant production in this study is based on several handbooks and guidelines for horticulture (Aldrich & Bartok, 1990; ASEA, 1993; Ball, 1988; Bartok, 1984; Hamrick, 1988; Hanan et al., 1978; Hanan, 1998; Van der Hoeven, 1995; IKC, 1991; Janssen, 1987; Van der Kwaak, 1989; Van der Kwaak & Van Weel, 1990; Langhans, 1983; Nelson, 1991; Stein, 1987; Stein & Van Weel, 1980; Tongeren & Peelen, 1984; Van Weel, 1991; Von Zabeltitz, 1986).

2.3.1 Ground production system

In this production system pot plants are grown on the bare ground without any special supporting device although tempex plates may sometimes be used. This production system has the advantage of a high technical space utilization of more than 90%. Space can be used in a flexible way: all available space can be used and there are no pot-size restrictions. This production system is used for those large plants that cannot be grown on transportable benches. A poor working position, however, is one disadvantage of growing on the ground. Picking up and putting down pot plants requires manual labour and bending, which is exhausting and often inefficient. This production system should be used for plants that do not require the sort of attention that involves bending. It is difficult to mechanize the handling of pot plants on the ground, because the bare ground will not support heavy machinery.

2.3.2 Concrete floor production system

In this production system pot plants are grown on a concrete floor (Figure 2-2-a). Concrete floors make it possible to combine different factors such as internal transport, a heating system in the floor and an intermittent flooding system for watering and fertilizing. A growing compartment with a concrete floor is divided into basins or sections. Sections are separated by paths. This production system has the same advantage as the ground production system: a high technical space utilization of more than 90% and it is used for large plants. However, a poor working position is one of the disadvantages of growing on concrete floors. There are possibilities for mechanization within a concrete floor production systems and these have gradually been introduced since 1990. However, it is difficult to devise a good combination of transport for operations such as potting, picking up, sorting, putting down and harvesting. Many operations can be mechanized or automated on a concrete floor, but spacing and harvesting will probably remain a manual task for some time to come. Light vehicles may be driven over a concrete floor to set up and remove crops. However, it is not possible to transport carts between basins, because thresholds divide them. Pot plants which are at the end of a section on a concrete floor cannot be reached from the main path until obstructing pot plants have been removed so that the ones required can be reached. This is a disadvantage when maintenance operations have to be carried out on individual (small groups of) pot plants. In concrete floor production systems, pot plants can be transported above the crop by conveyor belts.

a)

b)

c)

Figure 2-2. The most important production systems available for pot plants at the moment: a) concrete floor, b) transportable bench and c) rolling bench.

2.3.3 Fixed bench production system

In the fixed bench production system the benches have the function of supporting growing pot plants at a comfortable working height between 60 and 90 cm. A height of about 80 cm is a comfortable height for working with pot plants. The benches have fixed positions in the growing compartment. An aisle is situated between two fixed benches. Bench arrangements depend on the dimensions of a growing compartment, paths, doors, materials for handling and on the heating system. The relatively high number of transport paths required reduces technical space utilization to a maximum of 70%. This makes the fixed bench production system economically unattractive. Fixed benches are used for heterogeneous crops, when individual plants require much handwork. Fixed benches provide a good working height, but the fact that almost all operations have to be performed in the growing compartments where climate conditions are adjusted to plants and not to workers is a distinct disadvantage. Benches improve labour efficiency, permit more effective display and inspection, and assist air circulation. However, it is difficult to mechanize the handling of pot plants within the fixed bench production system.

2.3.4 Rolling bench production system

The rolling bench production system (Figure 2-2-c) involves benches that remain in more or less fixed positions in the growing compartment. Rolling benches are supported on rollers and can be easily moved 45-60 cm to each side in order to create a working aisle. Benches are moved by turning one of the support rollers with a crank. Rolling benches make it possible to use the entire growing area except for one or two work aisles. The standard widths for rolling benches in the USA are 1.52 m, 1.68 m and 1.83 m (5 ft, 5.5 ft and 6 ft). Workers can reach the centre of the rolling bench from both sides. Only one side of a rolling bench can be worked on at a time. The width allows the rolling bench enough space for movement, so adequate aisle space is created without the problem of the bench tipping off the rollers. In practice rolling benches are built in lengths of between 40 and 60 metres. The length of a rolling bench depends primarily on the chosen capacity of the transport device. The greater the transport capacity of the device, the longer the rolling benches can be. Using a transport device like a cart which has several layers ('etagewagen'), a mono-rail cart or a roller conveyor in the aisle between the rolling benches offers advantages. When choosing the transport device, the width of the aisle should be sufficient to enable it to be moved quickly.

A combination of fixed benches and rolling benches can have a technical space utilization varying between 70 and 80%. The rolling bench production system on its own can have a technical space utilization that varies from 80 to 90%. This is a higher technical space utilization than in the fixed bench production system. The ability to perform a certain amount of work within a certain period in one bay depends on the available number of aisles. However, when the number of aisles is increased, the technical space utilization will decrease. Rolling benches are used for heterogeneous crops in the same way as fixed benches. Rolling benches are generally used for crops that remain on the benches for less than four months or crops that require frequent spacing. Because accessibility is restricted, rolling benches are not generally recommended for pot plant crops that require frequent selection or where there are regular retail sales. Rolling benches provide a good working height but they have the same disadvantage as fixed benches: all operations have to be carried out in the growing compartments. It is difficult to mechanize pot plant handling in the rolling bench production system.

2.3.5 Transportable bench production system

A Dutch grower, Teun Boekenstein, developed the concept of the transportable bench production system in 1976 and subsequently the system has seen enormous developments. A transportable bench (Figure 2-2-b) is a flat rectangular aluminium tray of limited size on small roller wheels. It can serve as a bench and it can be easily transported from the working area to the growing area and back. Each transportable bench is able to carry a large number of pots. A transportable bench has the double function of being both a production and a transport system. The total number of transportable benches in the growing compartments of the companies surveyed by Annevelink & Van der Voort (1995 & 1996) varied from 600 to 8 428 (Appendix 1, Table 1). The size of transportable benches was not standardized. The length of transportable benches was 3.00-6.20 m and was adjusted to greenhouse dimensions. The width of transportable benches varied less, at 1.56-1.80 m, so that half the width corresponded to the reach of a worker. The area of a single transportable bench was 4.80-11.16 m^2. Each growing compartment is divided into a number rows and these can hold a number of transportable benches in specified positions on supporting stationary transport rails. Transportable benches can still have aisles between the rows allowing crops to be inspected. All companies in the survey had different row lengths. The number of transportable benches per row varied from 7-113 (Appendix 1, Table 2). Most companies had more than one row length in the growing compartments. Optimal row length depends on transport costs and anticipated loss of returns due to growing area occupied by paths. In the working area, rows are available as input or output buffers attached to the available work stations to avoid waiting time. The possible number of transportable benches in the various buffer rows varied enormously (Appendix 1, Table 3). Growing pot plants on transportable benches has resulted in new ideas about nursery layout. The traditional main path in the middle has been replaced by one or two main paths along the side of the compartment.

The transportable bench production system can have a high technical space utilization, that is often more than 80%. To make the best technical space utilization possible, the length of a transportable bench (or several of them) should exactly fill a bay. In the transportable bench production system it is possible to regroup pot plants that are not yet ready for harvesting, by relocating them from several partly filled transportable benches to one new transportable bench. This increases organizational space utilization. All pot plant nurseries in Annevelink & Van der Voort's survey (1995 & 1996) used the transportable bench system. The total net growing area on transportable benches was 4 290-48 920 m^2 (Appendix 1, Table 4).

Three companies combined the transportable bench production system with the concrete floor production system. The companies had a high technical space utilization in the transportable bench compartments of 83-93%, mainly because there were no paths between the rows (Appendix 1, Table 5). The transportable bench production system is especially suitable for crops with uniform small plants and a high circulation speed. When compared to other production systems the transportable bench production system has the best ergonomic conditions for human workers. Workers no longer have to enter the growing compartments. Climate conditions in the working area (temperature, CO_2, humidity and light) can be adjusted to workers requirements and workers are able to work for the most part at correct working height. At the same time the climate in the growing compartments remains optimal for pot plants. Ergonomic improvements were made to avoid workers having to make heavy pushing and pulling movements. Transportable benches can be transported automatically (without human efforts). Hand controlled transport devices have been replaced by automatically controlled systems. Another advantage of the transportable bench production system is that it can be combined with large specialized machines for potting, sorting, spacing and harvesting in a separate working area, because transportable benches can be transported to these machines. In this way, mechanization and the automation of operations replace heavy and monotonous work. Considerable mechanization has already taken place in transportable bench systems and the process is continuing.

A disadvantage of separating growing from the handling of pot plants is that transportable benches must be transported between growing compartments and the working area. The constant flow of transportable benches has to be scheduled carefully to avoid delays in the production process and in deliveries to customers. The sequence of transportable benches in a row cannot be changed without taking transportable benches out of the row again. Transportable benches are forced to flow in rows according to the FIFO system (First In First Out) or the LIFO system (Last In First Out, see Section 2.5.5). This makes it difficult to access a specific transportable bench in the growing compartment if it is in the middle of a row. If the required bench is at the back of a row, all benches in front of it will have to be moved in the LIFO system. Another disadvantage of the transportable bench system is that it requires relatively high investments. The required size of the working area will be larger. Additional costs will also be incurred for mechanization and automation. A further disadvantage is that empty transportable benches must be handled (cleaned, stacked and stored) and this requires additional equipment and takes up valuable space. The stationary rails and tracks in position throughout the greenhouse which are used to move transportable benches also take up valuable space.

2.3.6 Comparison of production systems

The descriptions of the different production systems provided above have been summarized in Table 2-5.

Table 2-5. Summary of the characteristics of production systems for pot plant nurseries.

Characteristic	Ground	Concrete floor	Fixed bench	Rolling bench	Transportable bench
technical space utilization	>90%	>90%	<70%	70-90%	>80%
crop	uniform	uniform	hetero geneous	hetero geneous	uniform
size plants	large- small	large- small	small- large	small	small
working height	bad	bad	good	good	good
working position	growing area	growing area	growing area	growing area	working area
mechanization possibilities	difficult	many operations still have to be mechanized	difficult	difficult	many operations have been mechanized

Stein (1987) made a costs comparison of the transportable bench production system comparing it with two other systems: the concrete floor and the rolling bench production system (Table 2-6). He found the investment costs related to transportable benches in 1987 were 100.00 NLG/m^2 bench and included the intermittent flooding system. This was about 89.32 NLG/m^2 glasshouse with a technical space utilization of 89%. Higher costs occurred where the system was fully automated or when the working area had to be adjusted. However, when a more expensive type of transportable bench was chosen, its higher costs had to be met by higher product quality, because the advantages of larger quantities or higher labour productivity could already be achieved using cheaper transportable benches (Stein, 1987). The investment costs for concrete floors were 32.00 NLG/m^2 floor. This was about 30.77 NLG/m^2 glasshouse at a technical space utilization of 96%. The costs of demolishing a concrete floor were also about 32.00 NLG/m^2 floor. The price of rolling benches was 60.00 NLG/m^2. This was about 53.59 NLG/m^2 glasshouse at a technical space utilization of 89%. The year costs (depreciation, interest and maintenance) for a growing system with an intermittent flooding system were about 6.97-9.02 NLG/m^2 glasshouse for concrete floors, 6.85 NLG/m^2 glasshouse for rolling benches and 14.11 NLG/m^2 glasshouse for transportable benches (Stein, 1987). The concrete floor growing system has the lowest investment costs. This is mainly caused by relatively low construction costs per m^2 concrete floor, compared to the construction costs of rolling benches or transportable benches (Tongeren & Peelen, 1984).

Table 2-6. Level of investment costs of different production systems in 1987 (Stein, 1987).

Production system	Investment costs (NLG/m^2 production system)	Greenhouse costs (NLG/m2 growing area)	Year costs (NLG/m^2 growing area)
concrete floor	32.00	30.77	6.97-9.02
rolling bench	60.00	53.59	6.85
transportable bench	100.00	89.32	14.11

The transportable bench system involves high investment costs. This is because transportable benches, Automatic Guided Vehicles and roller conveyors are expensive, and working areas have to be larger because of the space needed around the work stations and their buffers (Tongeren & Peelen, 1984). The cost of installing a transportable bench system will vary considerably depending upon size, design and the degree of automation included in the system (Reilly, 1981). In the ten years after Stein's study (1987) the investment costs associated with the transportable bench system have continued to grow. According to KWIN (1997) the cost of new, transportable benches are now around 60 NLG/m^2. A new, fully-automated internal transport system with transportable benches, robots and AGVs costs 85-125 NLG/m^2. Both transportable benches and the total system have a depreciation of 10% per year and maintenance costs of 5% per year.

2.3.7 Incidence of production systems

Ploeger (1992) has provided information on the incidence of different production systems at pot plant nurseries in the Netherlands. The data refers to 1989 (Table 2-7). Unfortunately, no data are available on the situation in 1999 because his survey was never repeated. In 1989 about half the pot plant area in the Netherlands consisted of bench production systems and the other half of ground production systems (Ploeger, 1992). A clear relationship existed between the area of the pot plant nursery and the production system. Smaller companies had more fixed benches (39%), larger ones had more transportable benches (18%) and concrete floors (18%). Glasshouses with fixed benches were found, on average, to be four years older than the average age of all glasshouses, while glasshouses with concrete floors and transportable benches were about three years younger.

In comparison with 1983 the portion of production on ground beds in 1989 had decreased from 45% to 34% and the portion of production carried out on concrete floors had increased from 5% to 14%. Fixed benches decreased from 33% to 18% and rolling and transportable benches increased from 16% to 25%. Thus Ploeger

(1992) concluded that between 1983 and 1989 the transportable bench and concrete floor production systems expanded.

There are several reasons why the research on scheduling internal transport in pot plant nurseries discussed in this thesis will focus mainly on the transportable bench production system. First of all, this system has been expanding and will probably continue to grow. It is particularly popular in larger companies whose economic perspective is generally better than average. A second reason is that the transportable bench production system is a highly flexible system that allows the grower to respond adequately to rapidly changing market demands. Furthermore, it is a production system that complies with the strict environmental regulations that have to be taken into account by growers and it can be combined with intermittent flooding systems, that prevent the loss of fertilizers and chemicals. The transportable bench production system is an ergonomically friendly system because almost all operations are performed in a separate working area and at a comfortable working height. However, the main reason for this choice is the fact that the internal transport control problem is most significant in this production system. This is because of the constant flow of transportable benches between the working area and the growing area and because access to the rows in the growing area is difficult.

Table 2-7. Percentage of different production systems at pot plant nurseries in the Netherlands in 1989 (Ploeger, 1992).

Production system	Size class (m^2/company)			
	< 5 000	5-10 000	>=10 000	All companies
ground (& tempex plates)	35	45	39	40
concrete floor	5	12	18	14
fixed bench	39	19	9	18
rolling bench	15	15	13	14
transportable bench	2	5	18	11
unknown and other	4	4	3	3
Total in %	100	100	100	100
Total in ha	182	252	483	917

2.4 Internal transport devices for the transportable bench production system

There are several important reasons for the mechanization and automation of internal transport (Hendrix, 1975; IKC, 1991; Van der Molen, 1992; Ploeger, 1992). These include the need to:

- reduce labour;
- improve labour conditions (better work posture and less tiring work);
- lower the risk of damaging products (as there is less handling);
- increase technical space utilization (when paths can be eliminated);
- improve the quality of products;
- lower the cost price;
- have a mobile crop, enabling a more efficient use of fixed equipment;
- separate the crop from the ground, necessary to meet environmental requirements;
- separate the growing area and the working area.

Disadvantages of highly mechanized transport devices are that they require relatively high investments costs, they involve higher data collection costs and they often lead to much more complex control processes. However, labour costs will normally decrease when these devices are used (Reuter, 1989). In the period 1991-1996 profit margins were very low in pot plant production because the prices of many species had fallen (Table 2-4), while production costs remained steady. This made growers postpone new investments in new production systems and transport devices. In 1997, the lowest point of the profit margins seems to have been reached, and producers of horticultural equipment expect to sell more systems again in the near future.

The choice of a mechanization system for internal transport is influenced by crop (uniform product), data recording, mechanization possibilities within the production system, the carrying-power of the soil, length-width ratio and the shape of the greenhouse, the position and number of the main paths and the size of the working area (IKC, 1991). A rule of thumb for the capacity of a transport system (Van Weel, 1991) is that it needs to be greater than the capacity of the work stations in order to keep the growing area filled as much as possible. Slightly higher operating costs for a transport system are less important than a potential loss of production. Transport time is related to transport speed and transport distance (Giacomelli et al., 1991). Speed is the most variable factor. Speed is affected by the size and weight of a transport unit, the surface on which the transport device travels and whether transport is a powered or manually pushed movement. Transport times should be as short as possible (Von Zabeltitz, 1986).

Table 2-8. The percentage of mechanization for potting, spacing and internal transport operations at pot plant nurseries in 1989 (Ploeger, 1992).

Size class in m^2	unknown	manual	manual/ mechanical	mechanical	automatic	total
< 1 000	10	87	-	3	-	100
1-2 000	9	89	-	2	-	100
2-5 000	1	84	0	15	0	100
5-10 000	-	68	2	27	3	100
>= 10 000	1	49	4	40	6	100
Netherlands	3	77	1	17	2	100

In most of the smaller companies (Table 2-8) internal transport still involves heavy manual labour and little mechanization (Ploeger, 1992). The mechanization possibilities available were not yet being widely used in 1989. Automatic internal transport was not often found and was usually installed in companies larger than 5 000 m^2. The investment costs of these systems were probably too high for many (small) companies (Ploeger, 1992).

Table 2-9. Stoffert & Rohlfing's (1983) classification system applied to internal transport devices used in combination with the transportable bench production system.

Main characteristics		**Secondary characteristics**	**Internal transport device**
discontinuous	not restricted to tracks	on and above floor; vertical	- lift & stacking system
	restricted to tracks	on floor; horizontal	- transportable bench - Automatic Guided Vehicle (AGV)
		above floor; horizontal and vertical	- hanging cart with lifting device
continuous	not restricted to tracks	on floor; horizontal	- roller conveyor - belt conveyor

Stoffert & Rohlfing (1983) made a systematic classification of internal transport devices in floricultural production (Table 2-9). First they distinguished between discontinuous and continuous internal transport devices. A continuous transport device means that the transported unit moves on a fixed framework such as a belt conveyor. A discontinuous transport device such as a cart moves while the transport unit rests on it. The second main classification feature was deciding whether the internal transport device was restricted to tracks. An unrestricted transport device can move to any position in the company, while a restricted transport device has to stay on its tracks. Within the four major groups which emerged from this classification, Stoffert & Rohlfing distinguished between internal transport devices that travel above the floor or on the floor. They also made a distinction between horizontal and vertical movement. Other characteristics of internal transport devices which have not been taken into account in Stoffert & Rohlfing's classification (1983), are whether internal transport devices are self-propelled or pushed/pulled, whether they are controlled manually or automatically (for self-propelled devices) and the number of transport directions associated with them (Janssen, 1987).

Stoffert & Rohlfing's classification (1983) refers to all the different types of internal transport devices to be found in floricultural production. However, this section only deals with the transportable bench production system. Internal transport devices, that can be combined with the transportable bench production system are:

- lift & stacking system;
- transportable bench;
- Automatic Guided Vehicle (AGV);
- hanging cart with lifting device;
- roller conveyor;
- belt conveyor.

Lift & stacking system

When a nursery is growing or transporting pot plant on two levels, a lift is needed to move transportable benches from the lower level to the upper level and back. Stacking systems are used to store empty transportable benches. They take transportable benches automatically, stack them above each other and deliver them again when needed (Stoffert & Rohlfing, 1983).

Transportable bench

A transportable bench has a growing function, but also a transport function (Section 2.3.5). Strictly speaking transportable benches are only a transport device when they are moved on their own rollers in the row of a growing compartment. As soon as they leave a row, the transport direction changes (from side-ways to length-ways) and the transportable bench is then taken by another transport device such as an Automatic Guided Vehicle, for example (Stoffert & Rohlfing, 1983).

Automatic Guided Vehicle (AGV)

Improvements to the original transportable bench production system include the replacement of hand pushing on roller conveyors by Automatic Guided Vehicles (Figure 2-3). More and more pot plant nurseries in the Netherlands use one or more AGVs to move transportable benches loaded with pot plants between the working area and the growing compartments. AGVs are powered by electricity using a towing cable or an overhead line. Semi-automatic guided vehicles are operated by an employee, who rides with the AGV as it transports a bench. With more sophisticated, fully automatic vehicles, one can use a computer in the working area to instruct the AGV to obtain a specific transportable bench in the growing area and bring it back to the working area. The AGV runs on a metal track buried in the greenhouse path. Computer controlled AGVs that can continue working for hours have virtually eliminated the need for employees to go inside the growing area. Fully automated, they can steadily move hundreds of transportable benches. Control systems on a central computer can collect data from the individual AGVs. These control systems know the path layout of the company and take transport orders either directly from a terminal or out of a file prepared by the operator or a control model. Computers will eventually be able to decide on the shortest route in time or distance. An AGV should be able to co-operate with central work stations and pot lifting devices (Ball, 1988; Hamrick, 1988; Stoffert & Rohlfing, 1983; Van Weel, 1991). Most companies in Annevelink & Van der Voort's survey (1995 & 1996) used semi- or fully Automatic Guided Vehicles (Appendix 1, Table 8). There were between 1 and 3 AGVs per company, and the number of transportable benches that could be transported per AGV lay between 1 and 6 (Appendix 1, Table 9). The maximum length of the transport path of an AGV varied between 44 and 500 m (Appendix 1, Table 10).

Hanging cart with lifting device

A hanging cart with a lifting device makes it possible to access each position in a row of transportable benches. This type of accessibility is known as 'Random Access' (Section 2.5.5). In Denmark, a system was developed to carry transportable benches overhead on a monorail with a new automatic overhead bench carrier. This is known as the Kuli system and freed the space that transportable benches traditionally require for the AGV tracks needed to move them. It also answers the problem of accessing specific transportable benches (Ball, 1998; Hamrick, 1988; Van der Hoeven, 1995).

Roller conveyor

When no guided vehicle is available, transportable benches can also be moved on a roller conveyor that covers (part of) the main path. Transport is either powered (automatically) or hand driven. After the transportable benches have been moved on the roller conveyor, they have to be pushed in a row either automatically or by hand.

Belt conveyor

The use of belt conveyors is restricted to situations where a continuous flow of uniform products needs to be transported. As soon as the transport flow becomes intermittent, or when the transport distance is longer than 100 meters, or when transport carts have to use the same route as the belt conveyor, problems of cost and operational management will arise. Belt conveyors in combination with the transportable bench production system will not often be used inside the growing area. However, conveyor belts can be combined with work stations in the working area. The plants are usually loaded and unloaded by a robot. A belt conveyor can be used for transport before pot plants are put on or removed from a transportable bench. Belt conveyors are mobile and should always be moved within easy reach of the operator and near the transportable benches (Hendrix, 1975; Van Weel, 1991).

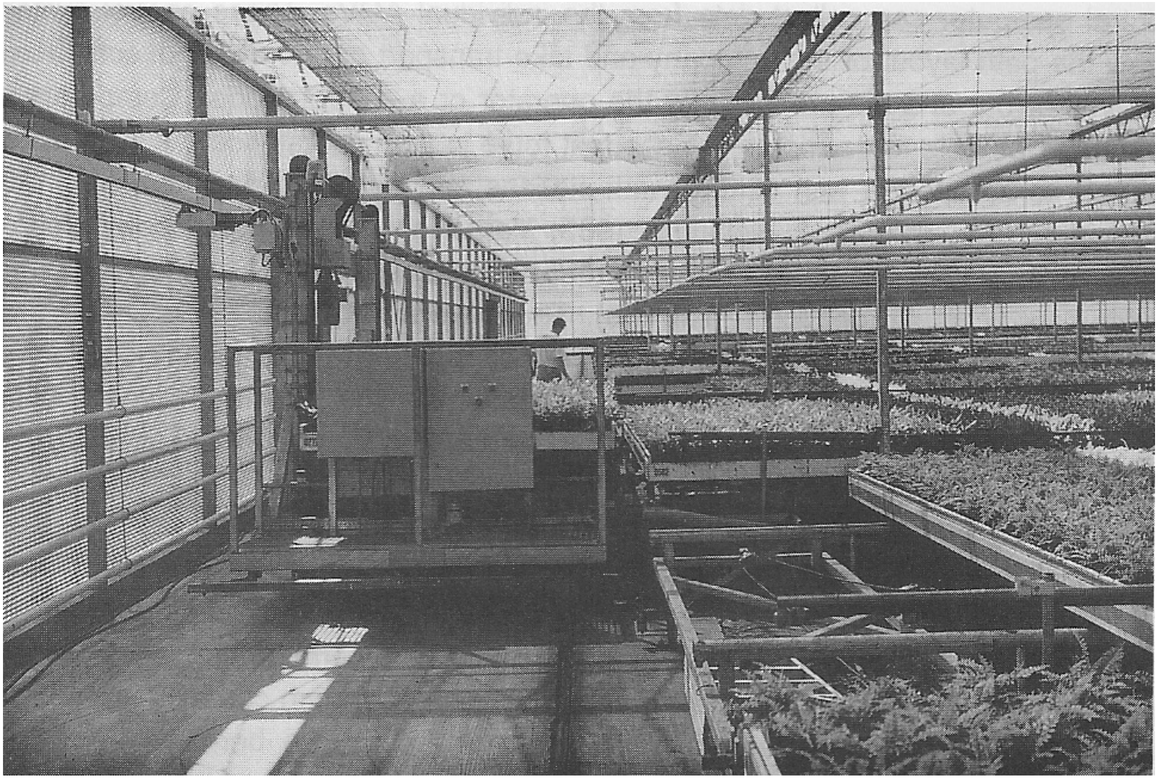


Figure 2-3. An Automatic Guided Vehicle (AGV) for moving transportable benches.

2.5 Internal transport process of transportable benches

A schematic representation of the internal transport process of transportable benches in a Last In First Out (LIFO) system (Section 2.5.5) is shown in Figure 2-4. Transportable benches have to be moved in a pot plant nursery for several reasons:

- to empty a work station buffer (transport from working area to growing area);
- to supply a work station with transportable benches, so that operations can be performed on pot plants (transport from growing area to working area);
- to reorganize the sequence or position of transportable benches in the rows (transport within the growing area).

The first two reasons are related to the operations of potting, sorting, spacing and harvesting crop batches that always require the transport of transportable benches between the growing area and a work station in the working area. An *operation* is any activity that changes some characteristics of the pot plants. A *crop* is a specified pot plant species, that has to be started during a specified period of the year. The number of pot plants is not yet specified for a crop. This is done in a *crop batch* which is a specific number of pot plants of a specified pot plant species started in a specified period of the year. The production process of a crop consists of different growing phases. A *growing phase* is the time between two operations (potting, spacing, sorting or harvesting). The production process of a crop is described by standard product data, which can be concluded from previously recorded data, from the advisory service or from literature. Standard product data specify the varying space and labour requirements for a unit of a pot plant species and the growth duration of a crop. A shorter crop duration usually requires more internal transport. Each transition from one growing phase to the next requires internal transport between a growing compartment and the working area. The operations, which require transport of transportable benches will be described in the following four sub-sections.

In the survey conducted by Annevelink & Van der Voort (1995 & 1996) both fast growing pot plants, with an average crop duration of less than 13 weeks, and very slow growing pot plants, with an average crop duration of more than 26 weeks, were grown in a transportable bench production system (Appendix 1, Table 11). The number of growing phases was between 2 and 5. The length of a growing phase was from 2 to 7 weeks in the group of fast growing pot plants. In the group of slow growing pot plants the length of a growing phase was much longer: 9 to 26 weeks.

The final reason for moving transportable benches during a growing phase is to reorganize their sequence in rows in a growing compartment. Reorganization is necessary to guarantee the *accessibility* of required transportable benches in a row in a growing compartment. When certain transportable benches in a row have to be transported, they might be obstructed by other transportable benches in the same row, that do not yet have to be transported. Additional transport movements will be needed at that moment to move the obstructing transportable benches to other rows. This will result in delays which could have been avoided by reorganizing the sequence of transportable benches in the row in advance. This way the transportable benches can be accessed directly at the front end of the row at the moment they are needed. Reorganization does not necessarily lower the total number of required transport movements, but it does lower the number of required transport movements at the exact moment of transportation. Therefore, reorganization can lower working station waiting time. Different layout related types of accessibility of transportable benches in a row will be described in the last section.

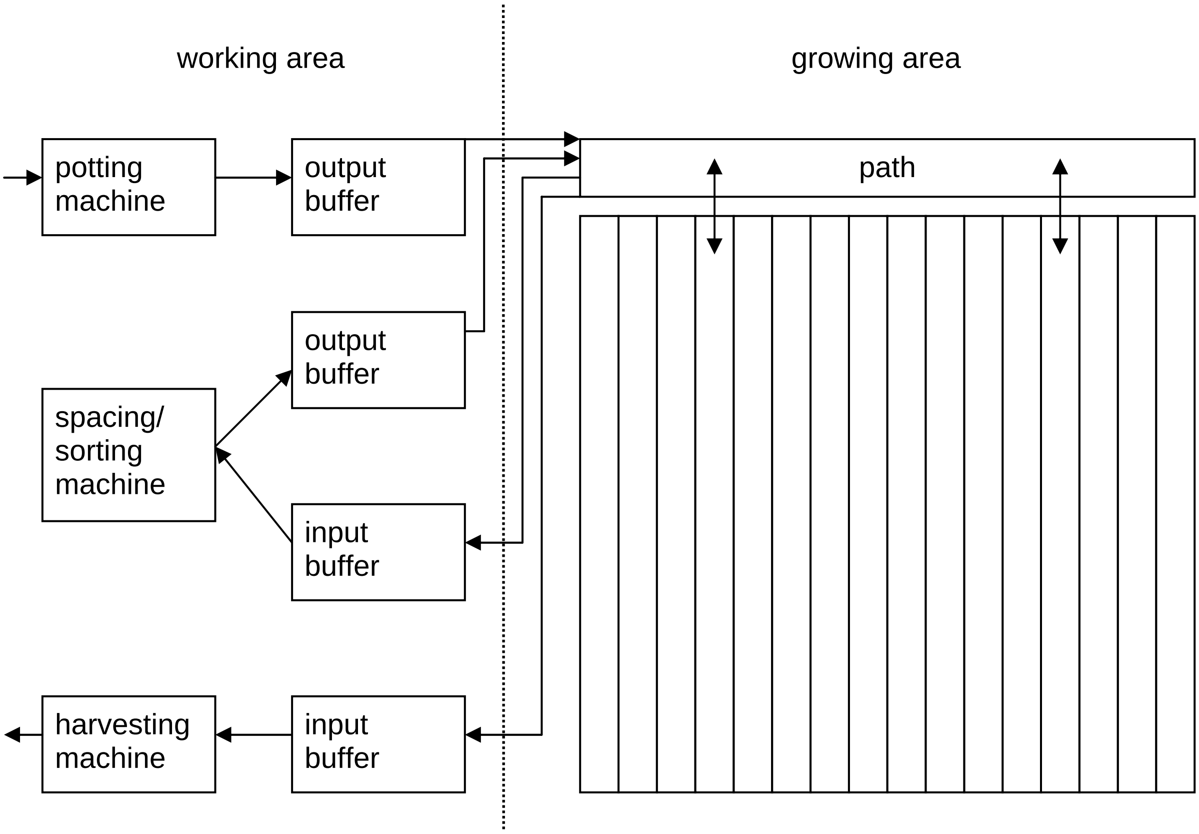


Figure 2-4. Schematic representation of the internal transport process of transportable benches in a LIFO system. Boxes indicate a position and arrows give the direction of internal transport.

2.5.1 Potting operation

Crop batches are potted year-round in a company. Pot plants of a crop batch enter the production process as seedlings or cuttings that are planted (semi)automatically by a potting machine in pots with soil. A potting machine and a robot to put the pots on the transportable bench are the most commonly used form of mechanization in pot plant nurseries (Table 2-10). More than half of the companies use a potting machine (Ploeger, 1992). The most common area for the potting operation is the working area. Automated equipment such as rollers, conveyors, automatic pot fillers, and stacking equipment can be permanently installed there (Langhans, 1983). Each company in Annevelink & Van der Voort's survey (1995 & 1996) had a potting machine. Some companies combined the potting and spacing/sorting operations at one work station in the working area. Where this was the case, only one operation could be performed at a time. This did not seem to give any problems because the operations were carried out on different days. The pot sizes used in the company survey were between 8.5 cm and 21 cm (Appendix 1, Table 12). When the transportable bench has been entirely filled with pot plants it is pushed into an output buffer row. At the end of the output buffer row a transport device will load one or more of the transportable benches and will then transport them along the path to a designated row in a growing compartment.

Table 2-10. The percentage of mechanization for the operations potting and spacing and internal transport at pot plant nurseries in 1989 (Ploeger, 1992).

Operation	Unknown	Manual	Manual/ mechanical	Mechanical	Not relevant	Total
potting	4	39	4	53	-	100
spacing	3	80	3	9	5	100

2.5.2 Spacing operation

At the end of the first growing phase the transportable bench with the crop batch will have to be collected from the growing area and brought to the working area, where an operation will be carried out on the pot plants standing on the transportable bench. The transport device will load one or more of the required transportable benches and will then transport them on the path to the input buffer row of the work station which has to perform the operation. In most cases the first operation after potting will be spacing. Each growing phase of a pot plant has different space requirements (Figure 2-5). At a certain point the leaves of adjacent pot plants will have grown close to each other and the plants no longer have sufficient growing space. The pot plants then have to be re-spaced so that there is a lower density of the pot plants on the transportable bench. Spacing means that pot plants are put at a greater distance from each other on the transportable bench. The spacing operation can be performed several times during the production process before the pot plants are finally ready for sale. When determining the proper spacing of a crop on the transportable bench, at least three factors must be considered: the labour needed to move the pots, the quality of the plants, and the cost of the bench per square meter per week. The ideal situation would be to always have the leaves from one pot just touching the leaves in the next pot. This way there would be no crowding or shading and at the same time no open space (Langhans, 1983). The spacing moment should be more or less accurately predicted for planning purposes. However, this is a problem because the length of a growing phase can vary depending on the actual growing circumstances of the crop batch. This may cause a certain delay or speeding up in the spacing moment, thus disturbing the production plan and the internal transport schedule. The number of pot plants per m^2 in the first growing phase is mainly determined by the pot size, because the pots are placed close to each other in that phase.

At 80% of the pot plant nurseries spacing is still carried out by hand. This is true also for the larger companies (Ploeger, 1992; Table 2-10). Pot spacing has only been fully automated in 9% of the pot plant nurseries. Pots are removed from a transportable bench by a robot with a pot spacing arm and put on a belt conveyor where they pass by an employee who selects out inferior plants. Another robot with a spacing arm picks up the pots and places them onto another transportable bench (Hamrick, 1988). The pot plants in Annevelink & Van der Voort's survey (1995 & 1996) were spaced between 0 and 4 times. Sometimes pot plants were also re-potted. The number of pot plants per m^2 differed from company to company, especially in growing Phase 2 and later. In these growing phases the density of the pots per m^2 depended on the growing process of the particular species of pot plant.

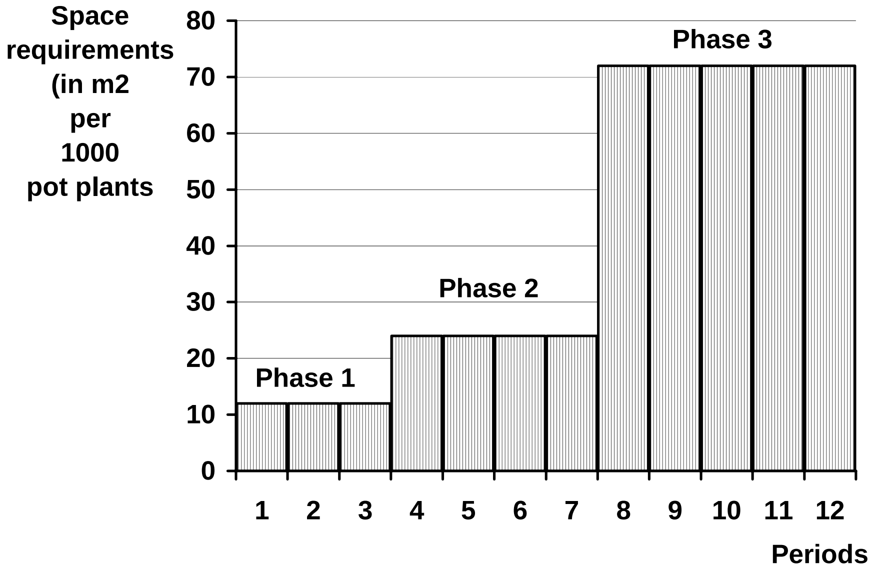


Figure 2-5. Example of the variable space requirements of a crop, specified in the standard product data (per 1 000 pot plants).

2.5.3 Sorting operation

There are several reasons why it is important to have crop batches with uniform pot plants. A uniform crop benefits from a uniform treatment. The main issue in harvesting is whether the crop batch can be sent to the working area on the transportable bench, and be harvested at one time, rather than simply picking out a few pot plants each day from a bench (Ball, 1988). Harvesting work will require less labour when there are uniform pot plants, because automated pot lifting devices can then be employed. Uniform pot plants will be ready for harvesting at the same time, so no filling in operation will be needed, and no transportable benches will have to be sent back to the growing compartment. Most crop batches can only be made uniform by sorting the pot plants from the original crop batch into sub-batches, using specific sorting criteria (height, potting date, leaf area, colour etc.). Vision systems can be used to automate the sorting process (Dijkstra, 1994). Sorting at an early stage in the production process is preferable to grading the final product as far as pot plants are concerned (Van Weel, 1991). Poor quality products can then be removed in time. However, sorting during the growing process influences the internal transport process. As a result of the sorting operation, sub-batches of the same crop batch have to be transported at different moments, the total amount of transport increases and the positions of all these sub-batches have to be planned carefully, in order to avoid additional transport movements. This means that the internal transport for each sub-batch has to be scheduled individually.

Most companies surveyed by Annevelink & Van der Voort (1995 & 1996) combined sorting with spacing. Eight companies had a robot to pick up pots from the transportable bench and ten companies had a robot to put the spaced and sometimes sorted pot plants back at a greater distance from each other on the transportable bench. Three companies sorted pot plants in two different quality classes and four companies sorted into three quality classes. One company in the survey used a recently developed computer vision system for sorting.

2.5.4 Harvesting operation

When pot plants have reached the required quality (size, flowering etc.) they have to be harvested for the market. During the harvesting operation - sometimes also called the selling operation - pot plants are taken from the transportable bench, packed and made ready for external transport to an auction or to an intermediate office working for the customers. Where pot plants are produced as uniform crop (sub)batches, it is possible to harvest the entire transportable bench at the same time. However, when a crop batch is not uniform and when not all pot plants are ready for harvesting, some additional growing time will be needed. The remaining pot plants are then put back on an empty transportable bench and they will have to be sent back to the growing area for a certain period of time. Harvesting is carried out throughout the year in pot plant nurseries.

In Annevelink & Van der Voort's survey (1995 & 1996) the number of pot plants harvested was between 0.5 and 6.5 million per year (Appendix 1, Table 13). This number depended on the size of the company, and also on the size of the pots. The length of the harvesting period of pot plants of the same crop batch was between 1 and 3 weeks. The harvesting process was only partly mechanized in the companies surveyed. This was probably because of the many different operations that had to be performed during the harvesting process (packing, mixing, sorting, etc.) and by the complicated visual selection task. Six companies used a robot to pick up pots from the transportable bench. Four companies had an automated packing machine.

2.5.5 Accessibility of transportable benches in the rows of a growing compartment

When studying the accessibility of transportable benches in a row three systems can be distinguished on the basis of the layout of the growing compartment and the rows themselves (see Section 2.3.5):

- Last In First Out (LIFO);
- First In First Out (FIFO);
- Random In Random Out (RIRO).

The first two entry systems to the rows in a growing compartment (FIFO and LIFO) were used in roughly equal numbers in the companies surveyed by Annevelink & Van der Voort (1995 & 1996; Appendix 1, Table 7). Three companies used both systems, but in separate compartments. None of the companies used the RIRO system. Each system has different advantages and disadvantages as far as internal transport is concerned.

Last In First Out (LIFO) system

In a LIFO system, the growing compartment has only one main path and therefore input and output of transportable benches is always at the same side of the row (Figure 2-4). An advantage of a LIFO system is that it needs less space for transport paths and thus less valuable growing area is lost. Another advantage is that less AGVs are needed in a LIFO system than in a FIFO system and thus the investment costs can be lower. The main problem of a LIFO system is that transportable benches of different crop batches may often obstruct each other in the same row when they have to be collected. This problem occurs when transportable benches have to be transported that are not situated directly at the output (= input) side of the row. When this happens transportable benches at the beginning of the row will obstruct the required transportable benches further on in the row. They will have to be moved to another row before the required transportable benches can be transported to the working area. Sometimes temporarily moved transportable benches have to be transported back to the original row, and sometimes they remain in the new row. Altogether many unnecessary additional transport movements will be needed in a LIFO system in order to constantly reorganize the sequence of transportable benches in rows.

First In First Out (FIFO) system

In a FIFO system the growing compartment has a main path on both sides of the rows (Figure 2-6). The input of transportable benches is always on one side of a row. During their stay in a row, transportable benches are slowly pushed through the row every time another new, transportable bench enters the row. Finally, the output of transportable benches is collected from the other side of the row. An advantage of the FIFO system is that the transportable benches can usually be accessed much better from two sides of the row. The main problem with a FIFO system is planning the sequence of transportable benches in a row in such a way that the pot plants on the transportable bench are exactly at the right stage of development when they reach the output side of the row. When a transportable bench is not yet ready to be transported when it reaches the end of the row, it will obstruct other transportable benches situated behind it that may well be ready for transportation. The transportable bench causing the obstruction will then have to be temporarily moved to another row. Another disadvantage of the FIFO system is that it requires more area for the transport paths and more AGV's.

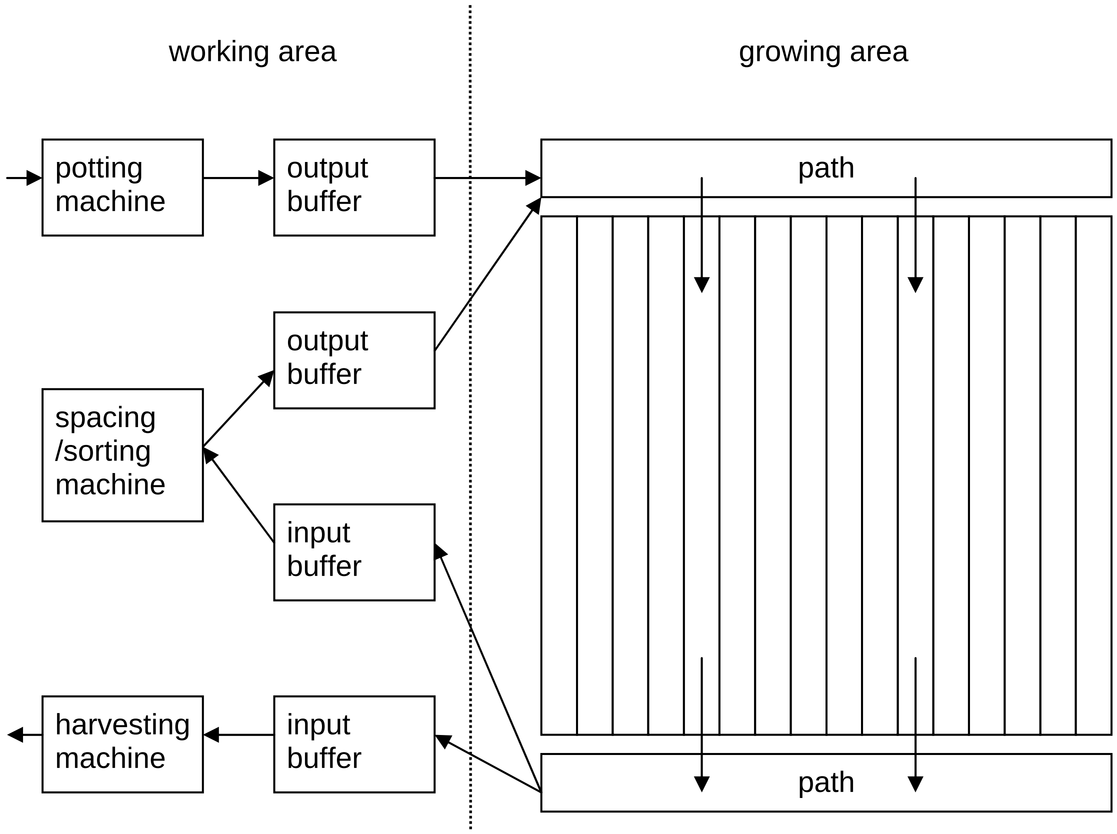


Figure 2-6. A schematic layout of a company with a FIFO system. Boxes indicate a position and arrows give the direction of internal transport.

Random In Random Out (RIRO) system

In a RIRO system, an overhead transport device hanging above the rows (see Section 2.4) is used during the input operation to move a transportable bench from the main path to any desired position in a row and let it down into position. It can also be used to pick up a transportable bench from any position in a row. In practice these overhead transport devices are not yet used on a large scale. They demand high investment costs because of the expensive equipment involved and because of adaptations which have to be made to the glasshouse construction, which has to be strong enough to support the overhead transport device. Adaptations, such as hanging tracks for example, may also cause a reduction of the light needed for the growth of the pot plants. A RIRO system does not have the problem of obstructing transportable benches, because each transportable bench can be accessed directly. However, controlling the sequence of collecting the transportable benches remains a problem.

2.6 Labour and planning costs of internal transport

2.6.1 Labour costs

According to Janssen (1987), a well-designed internal transport system should provide a system in which all operations can be executed without bottlenecks. The *labour costs* of such a system constitute one of the important cost factors in the economic evaluation of the system. When the capacity of the system, for example the number of transport devices, is overestimated, the system might be more workable, but it will also be much more expensive. Reducing labour costs should always be kept in mind when an internal transport system is designed. One of the most important objectives during the design of a pot plant nursery is to make the technical space utilization as large as possible. However, a large production area leaves insufficient area available for transport paths. More difficult transport circumstances will increase the labour time requirement for transport especially when workers are involved. To decrease the labour time involved in transportation, transportation distances or the transportation method must be changed. However, the latter leads to increasing other costs, because for example, more expensive transport devices such as AGVs are needed. To minimize the total cost of a production-transportation-system, all possible systems have to be compared. Cost calculations made by Stoffert & Rohlfing (1984) showed that it is useful to increase the plant production area, even when transportation paths become longer and smaller and labour time increased.

Differences in production and transport systems will lead to differences in labour costs (Stein, 1987). Labour costs are usually higher in systems other than the transportable bench production system. Time studies showed that as much as 35% of a worker's day was spent just walking from the greenhouse to the planting or shipping areas when the methods used before the introduction of the transportable bench system were being employed (Reilly, 1981). In the German situation, Reuter (1989) found that transport labour accounted for up to 60% of the total labour used in pot plant nurseries. High labour costs still occur in the transportable bench production system when workers are used to move transportable benches between the working area and growing area. When wages go up, the labour cost for internal transport will also increase in these companies. Therefore, manual internal transport of transportable benches should be avoided as much as possible. A manually controlled AGV still involves labour costs. Only when a fully automatic AGV is used can labour costs be avoided. In fact, in this case, hardly any labour costs are incurred. KWIN (1997) calculates labour data for pot plant production with a transport time for 100 m:

- 2.4 minutes/100 pots, when they are transported by a worker;
- 0.0 minutes/100 pots, when they are transported with AGV.

Labour needed to move transportable benches by hand constituted between about 5 and 10% of the total labour requirements of the companies surveyed by Annevelink & Van der Voort (1995 & 1996). When an AGV was used, transport labour was only between 1.6 and 3% of the total labour requirements. The number of workers in the pot plant nurseries in the companies surveyed varied from 3 to 34. In most of the companies only a few workers were involved in internal transport, most typically one manager and between 1 and 3 other workers.

2.6.2 Planning costs

Planning costs arise because the manager has to spend part of his or her time controlling the flow of transportable benches in the pot plant nursery. This time will increase in more difficult control situations, such as, for example, when AGVs have to be given detailed transport assignments or when the accessibility of the rows cause problems. Not only the manager's time, but also the additional costs of required planning tools (software) should be added to the planning costs. Finally, planning costs occur when the manager wants to avoid internal transport scheduling problems by lowering the organizational space utilization, for example, by keeping one row empty as an additional buffer or manoeuvring space. This can only be achieved by adjusting the production plan and by reducing the organizational space utilization and the profitability of the company. In one of the pot plant nurseries visited in the company survey, the costs of keeping one row empty (with 61 transportable benches) were calculated to be about 36 500 NLG/year. This was only the cost of the growing area and the calculation does not yet take into account the profit lost because no crop batches were grown in that row.

3. Design, planning and control of internal transport

3.1 Hierarchical integrative planning approach

The internal transport planning process is very complex. It is virtually impossible to find optimal solutions in an acceptable time. Therefore decomposition of growers planning problems into sub-problems is necessary. Hax & Candea (1984) describe Anthony's framework in which he classified decisions into three categories: strategic planning, tactical planning (or management control) and operational control. Decisions on the three planning levels cannot be made in isolation, because they interact strongly with one another. Therefore, a hierarchical integrated approach is suggested by Anthony to avoid the problem of sub-optimization. Furthermore, some kind of decomposition has to be made of the elements of the problem within the hierarchical system. Decisions at a higher level provide constraints for decisions at a lower level. In turn, detailed decisions at a lower level can be used to evaluate the decisions at the higher decision level.

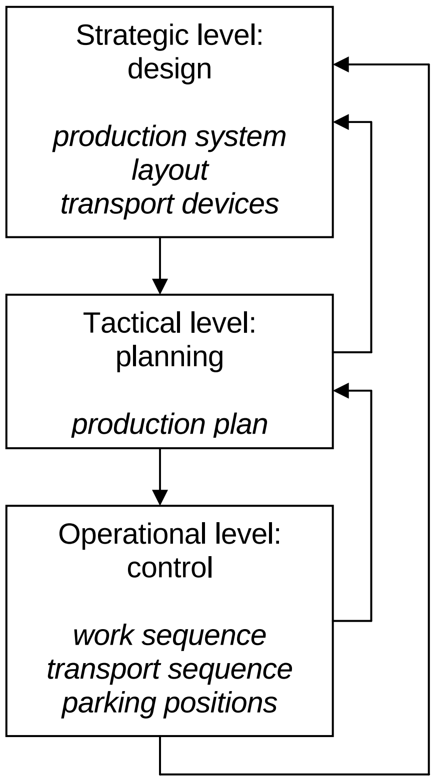


Figure 3-1. Interaction between different planning levels.

The activities at the three decision levels were defined as design, planning and control by Slack et al. (1998). Hax & Candea (1984) describe the three planning activities as facilities design, aggregate capacity planning and production scheduling. The first terminology is further applied in this chapter. The internal transport planning process can also be divided into these three hierarchical levels (Figure 3-1). A hierarchical planning approach enables the grower to decompose the internal transport planning problem into sub-problems, and thus lower the complexity of the planning process. The sub-problems can be solved separately. However, they impose constraints on each other, and these have to be taken into account.

Table 3-1. The differentiating characteristics of the three planning levels, related to internal transport (after Hax & Candea, 1985).

Characteristics	Strategic level: design	Tactical level: planning	Operational level: control
Decisions made on	- production system - layout - transport devices	- production plan	- work sequence - transport sequence - parking positions
Planning horizon	long: years	medium: year/months/weeks	short: weeks/days/hours
Scope	broad, company level	medium, compartment level	narrow, row level
Level of management involvement	top	middle	low
Frequency of re-planning	low	medium	high
Source of information	largely external	external and internal	largely internal
Level of aggregation of information	highly aggregated	moderately aggregated	detailed
Required accuracy	low	medium	high
Degree of uncertainty	high	medium	low
Degree of risk	high	medium	low
Required data related to internal transport:	- investment costs - global production plan	- available resources (space and labour) - gross margins - crop duration - duration of crop phases - space requirements - labour requirements	- expected handling and transport moment of crop batch (potting, spacing, sorting and harvesting) - actual space utilization - growing conditions per row (nutrients, climate, light) - average transport capacity

Design, planning and control of the internal transport process are activities performed by a grower and they are needed to maximize the effectiveness (achieve all targets) and efficiency (minimize production costs) of the entire production process. According to Hax & Candea (1984) each planning level has differentiating characteristics such as different decisions topics, a different planning horizon, scope of the decision, level of aggregation of the required information, degree of uncertainty and required data (Table 3-1). However, the planning levels remain mutually dependent and thus influence each other to a certain extent.

The *design* activity on the strategic level provides the form and the nature of an internal transport system of a pot plant nursery and imposes constraints on managing the ongoing internal transport process. The design activity considers only the broad, strategic objectives of quality, speed, dependability, flexibility and costs.

Planning and control is concerned with operating the internal transport system on a day-to-day basis (Slack et al., 1998). This activity has to ensure that the internal transport of the pot plant nursery runs effectively and efficiently and that a correct quantity of the required product arrives in time and at the desired position. In general, the model of planning and control provides the systems, procedures and decisions to develop the capacity to supply customers according to their demands. The general planning and control activity has to take place within certain constraints. These can be divided into four groups: cost, capacity, timing and quality constraints. These constraints also apply to planning and the control of internal transport in pot plant nurseries. Slack et al. (1998) state that the division between planning and control is not clear. However, they define *planning* as a formalization based on expectations of what is intended for the future. *Control* is defined as the process of coping with changes in variables that could result in a plan becoming unworkable. Control monitors what actually happens and makes adjustments which allow the pot plant nursery to achieve the objectives defined in the plan.

Slack et al. (1998) distinguish between *long-term*, *medium-term* and *short-term* planning and control. These types differ with regard to the time horizon, the aggregation level of the required data, the objectives and the balance between the planning activity and the control activity. Long-term planning and control puts emphasis on planning rather than control, has a time horizon of months/years, uses aggregated data, determines the resources in an aggregated form and primarily uses financial objectives. Short-term planning and control is aimed mainly at control, has a time horizon of hours/days, uses disaggregated data, makes interventions to resources to correct deviations from the plans made and considers objectives ad hoc. Slack et al. (1998) distinguish three distinct, though integrated activities within planning and control: loading, sequencing and scheduling. They define *loading* as the amount of work that is allocated to a work centre. *Sequencing* is the activity associated with determining the order of work. The priorities given during the sequencing activity can be given by some predefined set of rules. Finally, the *scheduling* activity is aimed at deciding on a detailed timetable, showing at what time or date jobs should start and when they should end.

Sequencing internal transport is a very complex task, because several different internal transport tasks have to be taken into account simultaneously (Slack et al., 1998). Differences in the capacity of the internal transport system(s) and the machines at the work stations further complicate the task. The number of sequences increase rapidly as the number of crop batches increases. For n crop batches there are factorial n (n!) different ways of sequencing the internal transport of the crop batches. Considering the complexity of the internal transport control problem, it is a difficult task to determine *optimal* sequences on the operational planning level. Fortunately, in most cases it will be sufficient to find an *acceptable* sequence within the given constraints.

The *interactions* between the strategic, tactical and operational level should always be taken into account. According to Reuter (1989) it is especially important to consider the interaction between the production plan (tactical level) and the transport sequence (operational level), and to see how this interaction influences space utilization. The integration of the different planning levels (strategic, tactical and operational) was mentioned as a difficult problem in Annevelink & Van der Voort's survey (1995 & 1996). A general production plan (tactical level) is needed for the design of the internal transport system (strategic level) to evaluate the design. A general production plan can be based on the current situation in a pot plant nursery, with some modifications for the strategic options which have to be evaluated. The production plan can also be based on a completely new situation. Decisions made at the strategic planning level generate constraints for the tactical planning level. Strategic constraints, which are too narrow for the tactical level, will lead to a sub-optimal production plan. Feed-back from the tactical level will result in adjustments to strategic decisions when they substantially impede tactical decisions. A production plan that has been fixed on the tactical planning level has to be translated to an operational planning level to establish day-to-day operations. The production plan specifies the exact size of the crop batches, and it guarantees that crop batches fit in the growing compartment during every period of the production plan. On the operational planning level some additional constraints have to be taken into account which cannot be taken into account on the tactical planning level because they are too detailed for that level. Constraints on the control of internal transport might lead to adjustments of the production plan, when it is impossible to start each crop batch exactly as planned. However, the control activity should first try to solve problems on the operational level before it signals that the production plan should be changed.

Design, planning and control have different phases, which are repeated sometimes once in several years sometimes several times each day, depending on the planning level. The stages of planning are:

- collect data;
- generate and evaluate alternative plans;
- implement and monitor a chosen plan;
- adjust a plan when necessary.

In the early eighties, growers recognized that adequate *data recording* within the firm was necessary to support planning activities and therefore a growing interest developed in automating the recording process (Ammerlaan, 1984). This development was stimulated by a desire on the part of the growers to exchange experiences with colleagues, based on accurate data. In 1983 the first project to automate data recording was started. At that time a relatively small group of motivated growers was interested in planning systems. In the Netherlands, an *information model* was designed to show what information is used in a horticultural company (Van Rijssel & Nienhuis, 1988). The information model was meant to serve as a basis for deciding what parts of a company could be computerized. The information model had two parts: a process model and a data model. Standardization was an important reason for developing the information model. Standardization was needed because, in the eighties, data exchange involving growers, advisory services, suppliers and the national authorities increased. In the glasshouse information model, three levels of planning are distinguished: long-term or strategic planning, medium-term or tactical planning and short-term or operational planning. In this model strategic planning is divided into four processes: the definition of objectives; the definition of production potential; drafting of alternative plans; and a selection drawn from these alternatives. The tactical planning function in the glasshouse information model is divided in two processes: the drafting of the production plan and the drafting of subsequent plans. Based on a production plan several subsequent plans can be drafted, including plans for climate control, nutrition, crop protection, specific operations for the production of plant material, specific growing operations, sales, management of durable means of production, personnel management and liquidity management.

In horticulture two types of computers can be identified: the climate computer and the management computer. The *climate computer* is used to automatically control the climate and the supply of nutrients in the growing area. The first climate computer appeared in 1975 (Van der Kwaak, 1989). According to LEI/CBS (1996) 6 852 climate control computers were being used in 1990 in horticultural companies under glass. The second type of computer is the *management computer*, which is more relevant for the data collection and planning phase. It is used for tasks like recording data, crop planning, activity planning, cost price calculations, invoices, shipping documents and packing lists, profit and loss calculations and the sales of each crop (Ball, 1988). The first data recording computer appeared in 1983 (Van der Kwaak, 1989). Management computers are less widely used than climate control computers. In 1990, 2 384 horticultural companies owned a management computer. All eleven companies in Annevelink & Van der Voort's survey (1995 & 1996) used a climate control computer and a management computer. However, these two types of computers were never directly connected or integrated. Almost all of the companies owned and used software to make a connection with the auction, to record data, to process and analyse these recorded data and to make a production plan.

Most managers in Annevelink & Van der Voort's survey (1995 & 1996) recorded data on selling price, costs, space and labour requirements, energy consumption, gross margins, locations, crop protection, crop growth, climate, order handling and pot plants in stock. They used the code number of the crop batch to record all data. All companies had been recording data for at least three years. Five had recorded data for more than eight years. The recorded data were used to understand crop growth and price trends, for internal comparison, as a basis for making and monitoring the production plan, and for space allocation and cost price calculations.

Several *techniques and tools* can be used to support the process of generating and evaluating alternative plans during the design, planning and control activities of the internal transport process. These techniques and tools can be divided into three main categories:
- rules of thumb;
- simulation;
- optimization.

Rules of thumb are derived from the day-to-day experiences of growers and their advisors. Each rule of thumb focuses on one small specific item of the planning process. Rules of thumb can be compared with priority rules that are frequently applied heuristics for solving job shop scheduling problems in practice, because they can be easily implemented (Dorndorf & Pesch, 1995). Sometimes rules of thumb are only valid for a specific situation, sometimes they have more general applicability and sometimes they may even conflict with each other. However, rules of thumb are always easy to understand, remember and apply. They are sometimes given in the form of a check-list, which can be consulted during the planning process (Fang, 1989; Reuter, 1989; Dambruin, 1989; IKC, 1991). Different rules of thumb apply for different planning levels, although some of them cannot be classified in terms of one particular planning level alone.

Simulation is a tool to evaluate rather than to generate a plan (Hax & Candea, 1984). Kleijnen & Van Groenendaal (1992) speak of simulation whenever a model has a time dimension (dynamic) and it is solved numerically. Simulation provides a means to look at proposed alternatives of the design, planning or control of an internal transport system. The grower must provide these alternatives himself and decide when he is satisfied with a solution. A limitation of simulation is that it is not able to optimize an alternative. According to Kleijnen & Van Groenendaal (1992) the most important advantages of simulation are that no advanced mathematics is required and that realistic models become possible. Simulation can evaluate a realistic internal transport system by taking into account detailed data on several aspects of the system.

Optimization techniques are used to generate an optimal solution to a problem within some specified constraints (Hendriks & Van Beek, 1991). Several optimization criteria can be used, including for example, financial criteria such as gross margin. Examples of frequently used optimization techniques are Linear Programming (LP) and Dynamic Programming (DP).

3.2 Strategic level: design

On the strategic level design decisions are made on subjects that influence internal transport in a pot plant nursery in the long term. When a design decision has been taken it will influence internal transport for several years. Important strategic decisions made about the components of the internal transport system described in Chapter Two include:

- layout of the pot plant nursery (see Section 2.2);
- production system (see Section 2.3);
- internal transport devices (see Section 2.4).

All managers visited in Annevelink & Van der Voort's survey (1995 & 1996) were innovators who continuously thought about long-term innovation in their companies. However, they considered the strategic planning process to be a difficult task. Design was always done in consultation with the builders and suppliers of greenhouses and internal transport systems. Some managers thought that some kind of management support system would improve the strategic design process, but this tool was not available to them at the time they had to make their decisions. During the strategic planning process, managers paid special attention to interaction between mechanization and labour, budget, available space, transportable bench size, crop specifications, project phases, choice of internal transport system and choice of mechanization systems. Objectives of strategic planning included improved labour and space utilization, lower costs, and expanding or modernizing the company. Many companies focused on the design of a pot plant nursery where they could grow one or two specific species. In theory they were still able to switch between products, but in practice this was very difficult to accomplish. Often it was not possible to have accurate data for the strategic planning process in advance and it was difficult to foresee all situations.

3.2.1 Layout of the pot plant nursery

The structure of internal transport should be taken into consideration when a new pot plant nursery is being designed and built (Reuter, 1989). One of the merits of being able to build a new greenhouse layout is the opportunity to create adequate working and growing areas and paths. The layout of a pot plant nursery should eliminate poor design in these areas, and also eliminate long transport distances and other obstructions. The ideal situation would be to have 100% of the ground area covered and still be able to meet the cultural requirements of each crop batch (Hanan et al., 1978). According to Nelson (1991) and Aldrich & Bartok (1990) it is important to follow a robust approach to design, one that develops a layout for the pot plant nursery that allows for future expansions, and ensures that an efficient operation will be possible in the near future. In greenhouse layout design, sub-decisions have to be made about:

- working area;
- growing area;
- paths;
- the spatial arrangement of working area, growing area and paths;
- accessibility of the rows.

The size of the required *working area* depends on the size of the pot plant nursery, the rest of the layout and the work stations used. Most internal transport occurs during input (after potting, spacing and sorting) and output operations (to enable spacing and harvesting) between fixed locations along predetermined routes (Fang et al., 1990a). Some other internal transport is needed to reorganize the sequence of transportable benches in the rows. An important issue is to find the appropriate capacity for input and output operations. Capacity is restricted by the size of the buffer space around a work station. Just-in-size buffers are needed in the working area to maintain capacity, to deal with tuning problems in respect of input and output movements, to minimize potential bottlenecks of materials transport and to minimize the space required for the working area (Fang et al., 1990a). The division of the *growing area* into different compartments influences the size and the number of required paths. This division enables the grower to create different climate sections. This puts a restriction on the allocation of transportable benches with certain crop batches needing specific climate conditions. The *paths* through a greenhouse should be large enough to accommodate available internal transport systems (Nelson, 1991). The choice of the direction of transport in the main path (one or two way traffic) has an impact on the possibilities for transport routing. The *spatial arrangement of the working area, growing area and paths* has an enormous influence on internal transport and should be carefully planned (Reuter, 1989). A nearly square design minimizes the distances across which plants and materials have to be moved (Nelson, 1991). The *accessibility of the rows* can be a serious problem for internal transport. Poor accessibility leads to additional internal transport movements. The possibilities of accessing the transportable benches in a row are determined by the input sequence in the row (Section 2.5.5). When the required transportable benches are not directly accessible because they are in the middle of a row, it will be necessary first to transport obstructing transportable benches to other rows within the growing compartment. In this way the positions of the transportable benches in the row are reorganized. This might well cause new problems later on.

3.2.2 Production system

A production system has to be chosen at the strategic planning level. This study concentrates on one production system only: the transportable bench production system. When a decision has been made on the main production system, further sub-decisions still have to be made such as determining the dimensions of the transportable bench.

3.2.3 Internal transport devices

Finally design decisions have to be taken at the strategic planning level about internal transport devices. This involves sub-decisions about efficient ways of moving the transportable bench, about the question of whether to move transportable benches automatically or to use workers, and about investments in new devices to save labour. Internal transport device capacity depends on potting, spacing, sorting and harvesting machine capacity, workers and transport distances. The capacity should be sufficient to keep up with the fastest operation (in most cases potting) in the pot plant nursery (IKC, 1991). According to Van Weel et al. (1991) the choice of complex internal transport systems for an integrated production system for greenhouse crops should always be considered in relation to the total production system. Possible *performance indicators* for an internal transport system have been given by several authors (Janssen, 1987; Dambruin, 1989; IKC, 1991):

- total operating time to perform a group of work orders;
- throughput time per transportable bench when performing a group of work orders (total operating time divided by number of processed transportable benches);
- utilization of workers, transport devices, buffers and work stations;
- cycle time of a transport device;
- amount of crossings;
- amount of delays;
- waiting time in buffers;
- technical space utilization;
- organizational space utilization.

Fang et al. (1990a) point out that transport distance is one of the important factors of the greenhouse internal transport system. Transport distance is variable and it affects the availability of transport devices and workers. Transport speed is another important factor. In a study by Dambruin (1989) a transport cycle of a continuous input operation starts when a transportable bench is taken from the buffer of a work station - for example the potting machine - and brought to a row in a growing compartment. It ends when the transport device is back at the buffer. The transport cycle time in that study also included delays. However, in practice internal transport movements for input, output and reorganizing operations are mixed, and therefore it is more difficult to define the beginning and the end of the transport cycle in practice than in Dambruin's study (1989).

3.2.4 Design techniques and tools

This section describes some of the design techniques and tools described in the literature concerned with the strategic level. Some general *rules of thumb*, which are relevant for the design of internal transport in greenhouses at the strategic level are given in Table 3-2. These rules of thumb refer to the working area, growing area, paths, routing and transport devices. They should be regarded as general considerations during the design process, and not as actions that can be directly applied during a design process.

An internal transport *simulation* model was built by Elsner & Reuter (1989; Reuter, 1989) as a planning and advisory tool for building new horticultural companies or rebuilding the existing ones. The model was intended to improve internal transport through technical and organizational measures. They focused on a production system with transportable benches. Their internal transport model had to be supplied with relevant data based on the present-day situation in a pot plant nursery. General data dealt with the production plan, labour management and technical equipment. Nursery-specific data dealt with greenhouse area and size, distances, the production and transport systems, the production plan and labour management. A work routine library described the order of events within the transport systems. During a simulation run, data on material movements, transport, and handling times were recorded in the processes. The report of a simulation run with Reuter's model (1989) gave diagrams of the use of the transport devices, the use of transport- and handling-labour, the use of the production area and the number of transportable benches actually transported. These results were able to show bottlenecks in the transport process. Results were discussed with the manager and if improvements were identified they could be applied in the company. Reuter (1989) tested his internal transport simulation model at a large pot plant nursery in Germany (38 500 m^2 glasshouses), that had both transportable benches (3 927 in total) and a mechanized internal transport system. The production plans for the examples were made by hand. A year was used as the simulation period.

The examples given by Reuter did not test the simulation model to its extreme. The total space utilization was only 60 to 70%, so it cannot be said that all possible bottlenecks occurred in his tests. At a space utilization of 90 to 95%, new bottlenecks

Table 3-2. Rules of thumb for *strategic* design of the production system, layout and internal transport devices in pot plant nurseries (Hamrick, 1988; Dambruin, 1989; Reuter, 1989; Aldrich & Bartok, 1990; IKC, 1991).

Category	Rule
Working area	- place buffers around continuous transport systems and work stations; - make buffers First In First Out preferably; - avoid positioning work stations with manual labour in line with each other; - avoid the use of continuous transport systems for storage; - minimize the amount of stored materials;
Growing area	- avoid doors or automate them; - avoid thresholds and unevenness; - do not store non-producing benches or units in growing positions;
Paths	- keep transport distances as short as possible; - reduce the width of paths by using a track for guidance; - a main path with dual-way traffic is preferred in terms of space utilization; - a main path at the centre and receiving/shipping at the centre are preferred in terms of transport distance; - for accessibility, all main paths should be adjacent to both receiving and shipping; - as far as transport time is concerned, the direction of the main path should be parallel to the longer edge of the growing area, if the transport speed in the main path is higher than in the row;
Routing	- avoid crossing transport routes; - make transport routes with low transport capacity as short as possible, if they cannot be avoided; - avoid corners in transport routes; - avoid vertical transport; - keep transport routes clear of obstacles; - create alternative routes for emergency; - find optimal routing for processing lines, such as potting-, spacing- and sorting machines;
Transport devices	- make transport capacity larger than the capacity of an operation; - avoid manual transport; - avoid relocation during growth; - make movement continuous; - maximize transport speed by using a track for guidance.

could have occurred in his runs. One of the assumptions Reuter made when he tested his model was that each row in the growing compartment contained only one crop batch. This meant that the size of a crop batch was determined by the size of the row (in his case 29 transportable benches). This is an unrealistic assumption, because it does not take into account economic considerations, which should be the main determining factor for crop batch size and which might lead to differences in the size of the crop batches. It is also unclear what happens in Reuter's simulation model after a crop batch was spaced.

Lombardot (1989) developed the management support system 'Adi serre' to assist the design and comparison of projects for greenhouse investments. The programme worked with a greenhouse investment file. Five projects can be taken into account at the same time. A project was described by the investments costs. These include site development, greenhouse, irrigation, fertilization, climate, farming equipment, packaging and other investments. The economic aspects of the operation were included as estimates of costs, income and results.

Fang (1989) developed a model to support the strategic design process of a greenhouse system. His model was called Computer Assisted Systems Engineering (CASE) and consisted of six integrated modules: crop, device, layout, simulation, optimization and engineering economic analysis module. *Part I of the device module* of CASE contained the device and relocation data bases of Giacomelli et al. (1987), who studied available transportation techniques and devices. They found that it is difficult to select and analyse combinations of mechanized systems, because transport needs are interrelated with the production processes and impose an unpredictable transportation demand. Flow charts of all potential combinations of transport devices and pathways, were created with the aid of a database analysis programme. Transportation and relocation processes were distinguished, both of the labour intensive and automatic type. The labour requirements of a specific transport system were determined by adding the labour requirements of all elements in a given pathway. The elements in the device database were described by type of task (transport or relocation), time required for a task, labour input, type of device, device capacity, and initial and final locations of the transport device. *Part II of the CASE device module* was used to compare transport devices in a user-defined system with a base system (a layout with a bare floor and an unrestricted cart). According to Fang (1989) the input data to evaluate a transport device should always be derived from a complete company with a year-round crop production schedule. The *CASE layout module* (Fang, 1989) was used to determine optimal bench and layout design under given constraints. The optimum dimensions of a bench (fixed greenhouse size) or the optimum greenhouse growing area (fixed bench dimensions) could be calculated. Different alternative layouts were stored in this module. The *CASE simulation module* was developed to study the dynamic behaviour of a greenhouse internal transport system with a transportable bench growing system

(Janssen, 1987; Dambruin, 1989; Fang, 1989; Fang et al., 1992a). This simulation model focused on the strategic planning level. Two separate simulation modules were developed. One in the SIMAN discrete event simulation modelling framework and the CINEMA animation system and one in a general purpose language Basic. The first system had an animation component, which helped to debug and verify the model visually. Stochastic modelling techniques were used by Fang et al. (1990a) because several factors are probabilistic (e.g. potential machine failure and bench relocation time). A production plan was used as the starting point of the approach. Animated graphics displayed real time statistical information in the CASE simulation module. Three feasibility studies were performed with CASE. These were aimed at mechanizing or automating the process of placing plants on floors, picking them up again and transporting them (Van Weel, 1992). These studies focused on pot plants on concrete floors in a glasshouse (Boot & Van Waarde, 1990) and on growing nursery stock in pots outside on hardened soil (Smeenge, 1991) and non-hardened soil (Janssen, 1991). The simulation model, which was built with SIMAN/CINEMA, did not perform well. The learning curve required to build new adapted models was too long. The possibilities for analysing the simulation data were insufficient because it was impossible to interrupt the simulation process.

Boot & Van Waarde (1990) studied the possibilities of automating internal transport at pot plant nurseries with concrete floors. They compared five fictitious automatic transport systems using the optimization module of the IMAG Production Planning system (Section 3.3.3). Transport with an Automatic Guided Vehicle was found to be the best alternative. The results improve when the internal transport systems can be used in more than one compartment.

3.3 Tactical level: planning

On the tactical level planning decisions have to be made on topics which influence internal transport in a pot plant nursery in the medium term. In his study Reuter (1989) found the production plan had a strong influence on internal transport. Once decisions have been taken on the tactical level, these choices will influence internal transport for a period that can range from several months to a year. The production plan is the most important tactical decision in planning internal transport. Estimates of space and labour requirements and gross margin are important considerations when constructing a production plan. Uncertainties that have to be taken into account include strong fluctuations in selling prices, the sudden occurrences of plant diseases, variations in crop duration and unexpected delays in the supply of raw materials. A special problem is the fact that pot plant nurseries have a year-round production, and crop batches follow one another continuously in the growing compartment.

3.3.1 Production plan

Each crop of pot plants has its own standard production process with specific operations (potting, spacing, sorting and harvesting), which result in a need for internal transport (Section 2.5). A crop with strongly varying space requirements will need more internal transport than a crop which has unchanging space requirements throughout its growing period. Most pot plant nurseries produce a number of different pot plant species. The number of species produced in Annevelink & Van der Voort's survey (1995 & 1996) varied from 1 to 25 (Appendix 1, Table 14). Both green and flowering pot plants and various combinations of the two are grown on the transportable bench production system. The complex production situation leads to an almost intractable number of possible combinations and it is almost impossible to make a choice by hand. Therefore a grower needs a good production plan if he is to utilize the full production capacity of the pot plant nursery.

In the production plan the grower takes decisions about the exact size of the *crop batches* within the constraints set at the strategic planning level. A crop batch is a specified amount of a specified crop started in a given week. The production plan leads to a certain amount of internal transport. It specifies the exact number of pot plants, which have to be transported for each crop batch and it also specifies the actual moment of transport actions. The objective of production planning is to maximize profit or gross margin within the constraints of the space available in the growing compartments and the labour available. A company must ensure that its tactical production plan can be executed at the operational level. This includes checking the amount of internal transport. However, operational constraints should hardly influence the size of the crop batches in the production plan. Maximizing gross margins (on the tactical level) is a more urgent priority than minimizing the amount of internal transport. Some companies give priority to filling complete rows in the growing compartment with only one crop batch to avoid extra internal transport. In such cases the production plan on the tactical level is unjustly subordinated to the control of internal transport on the operational level.

Five pot plant nurseries in Annevelink & Van der Voort's survey (1995 & 1996) used software for production planning. The main constraints taken into account during production plan design were space and labour availability. The rolling planning horizon was usually one year. Most companies used planning periods of one week. The number of transportable benches needed for a crop batch often increased strongly in each phase of crop growth with considerable consequences for the amount of internal transport needed for the crop batch. Not all data on the average number of pot plants started per week and the number of transportable benches needed in each growing phase were available for every company covered in the survey. The number of transportable benches increased very strongly for some crop batches when they were spaced and, therefore, the amount of internal transport also increased (Table 3-3).

Table 3-3. The mean number of pot plants started per week and the number of occupied transportable benches in each growing phase in Annevelink & Van der Voort's survey (1995 & 1996).

Company	Start weeks	Potted plants per week	Number of occupied transportable benches per week after:				
			potting	first spacing	second spacing	third spacing	fourth spacing
(4)	yr[1]	67 800	130	260	-	-	-
	yr	15 000	26	-	-	-	-
(6)	yr	15 000	10	29	-	-	-
	yr	85 000	53	126	-	-	-
	4-14	25 000	25	48	89	-	-
	10-20	7 500	5	11	25	50	114
(7)	yr	50 000	116	241	285	-	-
(8)	yr	10 800	45	90-103	127-180	-	-
(9)	yr	50 000	110	258-294	-	-	-
(10)	yr	20 800	52	104	-	-	-
	15-45	3 000	12	28	50	-	-
	yr	4 000	10	24	40	80	-
(11)	yr	9 000	23	58	84	-	-

1) yr = crops started every week year-round

3.3.2 Production planning techniques and tools

Rules of thumb are not often used for the tactical planning of internal transport in greenhouses. However, many tools have been developed to support the *optimization* of production planning in pot plant nurseries. Most optimization models were based upon Linear Programming (LP) and were developed in the (early) eighties (Basham & Hanan, 1983; Bloch, 1983; Fang, 1989; Lentz, 1985; Lentz & Buchwald, 1989; Ottosson, 1983; Saedt & Annevelink, 1988; Saedt et al., 1991; Sowell et al., 1982; Weston & Schumacher; 1983). Most of the LP-models are cyclic: all decisions are repeated after some time, in most cases after one year. Some research also focused on the transition from the present situation to a future cyclic situation (Bloch, 1983; Saedt et al., 1991). Schumacher & Weston (1983) discuss the advantages of the LP technique. These include rapid production of solutions once the model is formulated in the required format and the fact that 'what if' type of analysis can be conducted. They also see limitations to LP. One is the assumption of linearity. Another is the assumption of certainty. A major limitation may be the capacity of the manager to adequately express his or her problem in a mathematical framework. Finally, data accuracy is often a limiting factor in an LP-model. However, Schumacher & Weston (1983) still think that the use of LP permits the grower to evaluate a wide range of product-mix combinations under a variety of constraint considerations. In their opinion, the tool is relatively flexible once the grower learns its capabilities and limitations. Several authors do not agree with this opinion: they did not consider optimization (with Linear Programming) to be the most suitable technique for finding solutions to production planning problems. They emphasized the restrictions of LP, like the necessity of having linear functions, the need for detailed input data and the relative complexity of the modelling technique. In response to the optimization approach, they developed other planning techniques and tools that rely more heavily on *simulation* (Håkansson, 1983, 1987 & 1991) or *other techniques* like interactive backtracking (Hofstede, 1992) and chance-constrained programming (Ludwig, 1989 & 1993). These techniques focussed less on the mathematical solving technique and more on the role of the grower in the planning process. The grower (and not the computer) had to be in full control of the planning process. A third group of planning systems used a combination of both approaches. Optimization was then used to generate ideas for new production plans and simulations to allow an adjustment of production plans. The rest of this section will give an overview and a short description of some of the production planning systems from these different approaches.

According to Basham & Hanan (1983) the practical difficulties of applying LP to solving greenhouse management problems mostly centre on three areas: deciding upon input parameters and gathering data, formulating the matrix so the mathematical model expresses physical and economic reality and interpreting the results and applying them in operations. They used two week periods in their model as a compromise between periods of one month (with a loss of resolution) and periods of one week (with a very large matrix). Their model used a cyclic approach to avoid the production plan being affected by the way start and end conditions are specified. According to Basham & Hanan (1983) the success of an LP management application largely depends on close co-operation between the decision maker (the grower) and analyst (advisor).

Bloch (1983) distinguished the problem of implementing a theoretical optimal plan in practice. He constructed a transition plan by hand, but was not able to optimize it. Optimal solutions produced in Bloch's (1983) 'free' production plan were too theoretical because they did not take the market, sales, price, quality and climate sufficiently into account. Furthermore, the LP-plan was based on a continuous solution and not a mixed integer solution. Therefore, he sometimes had to adjust crop batches to an integer number of benches by hand.

The *CASE optimization module* was used to maximize profit on a year-round basis, subject to labour, budget and space restrictions (Fang, 1989). Three optimization techniques were used: Linear Programming, Integer Linear Programming and enumerative methods. The optimization module had two functions: to perform case studies to determine the feasibility of a proposed production plan and to solve optimization problems (Fang et al., 1990b). The results of the CASE optimization module are:

- feasibility of a given production plan;
- number of potted plants to be produced for each selected crop;
- initial number of cuttings or seedlings required for each crop;
- weekly and annual space utilization;
- weekly labour requirement and yearly labour cost;
- annual budget requirement;
- annual profit.

Lentz (1985) built the first version of his programme 'Anbauplanung' in 1982. This planning model is based on a traditional approach. First, the user has to define a number of different production methods and then a good or optimal combination of the defined production methods is found (by Linear Programming or simulation), which maximizes the gross margin. According to Lentz (1985) the advantage of LP is, that it enables the grower to generate alternative plans during an iterative process, and that it gives the grower new ideas to break through routine plans. The disadvantage of simulation is the danger that the grower goes back to a certain routine and does not consider new and perhaps better alternatives. According to Lentz & Buchwald (1989) it is impossible to work out an optimal and practically feasible plan in just one LP-model run. Instead there must be an iterative planning process. During the search for a practical production plan it is necessary to adjust production method data and to insert a number of additional restrictions several times.

Buchwald (1987) and Lentz & Buchwald (1989) also describe a second simulation model for production planning, in which the production methods of a crop are not predetermined but variable. The definition of the production methods takes place during the planning process itself by specifying the date and kind of operations. The model is supported by growth functions, which consider seasonal growth rates and effects from the temperature regime and by functions for the input of energy, labour and materials. No formal optimization algorithm assist the search for an optimal plan in this second simulation model. The model assists in production planning and it integrates production control and provides for revisions of the initial plan. The simulation model was intended to reduce the time needed to collect and record the large amount of data and to make the planning system more flexible. Buchwald (1987) emphasizes the dynamic character of his planning tool. According to him the lack of a search algorithm is both the strength and the weakness of simulation because, on the one hand, a good correspondence between the production plan and the actual state of production can be achieved, while on the other the path from this actual state to an optimal plan might be troublesome and requires an experienced planner.

Ludwig (1989 & 1993) considers the factor *risk* in production planning. She distinguishes three different types of risk factors. Availability risks include factors relating to the available stocks for production. Production risks affect the production process itself. And finally marketing risks threaten sales. Ludwig's model (1993) used stochastic Linear Programming, where some or all coefficients are stochastic variables with known probability distributions. Chance-Constrained Programming was used, a special type of stochastic LP with random parameters only in the constraint set. Chance constraints and probability distributions were used to represent the stochastic nature of the cultivation periods in pot plant production. Stochastic coefficients were used for space requirements, because the actual length of the cultivation period during plan realization is not known. During winter the cultivation periods vary strongly. The model results for single crops showed the large influence of the decision maker's risk attitude on the gross margin. The higher the security level, the more buffers have to be considered in the model and this results in lower gross margins.

Ottosson (1983) describes the TEU method, which is a management method for business analysis and planning of horticultural firms that was developed by the Division of Horticultural Economics at the Swedish University of Agricultural Sciences (TEU) in 1978. An *optimal plan* was calculated with LP for the year ahead. Thorough knowledge of firm-specific data was needed and therefore a good recording system was essential. Ottosson (1983) gives some disadvantages of the LP-planning method: it demands a lot of work to get firm-specific data, a grower very often does not use the optimal LP-solution, because it is not possible to make such drastic changes in his company and finally a lot of factors are very uncertain, which makes a very exact LP-plan unnecessary. According to Håkansson (1987) pre-calculations of gross margins are fairly unreliable in horticulture, because of the great fluctuations in market prices that characterize the sector and because of disturbances in the development of crop batches. Therefore, he concludes that it is impossible to use the LP-method as an optimization tool. Another disadvantage of the LP-method that he identifies is its inability to handle practical limitations such as the different temperature requirements of crop batches in the same compartment. Often the size of some of the crop batches in the plan has to be fixed and in this way optimization is overruled. However, he sees an advantage of the LP method as being that it suggests alternative plans and that it gives information about shadow prices. To meet the difficulties caused by the restrictions of the LP method, Håkansson (1983, 1987 & 1991) built a *simulation model* for production planning in greenhouses as an extension to the TEU-planning system. The production plan was based on gross margin calculations and information about the space and labour needed in the greenhouse for each crop. Different cropping alternatives could be simulated to compare their gross margins. The optimization algorithm was replaced by a routine where a diagram, which showed time on the X-axis and the greenhouse area on the Y-axis, was drawn on the screen. In this diagram, the area occupied by each crop in the greenhouse is marked by a certain colour whereas empty space is left blank. In the opinion of Håkansson (1991) the applied trial and error procedure in his simulation model is more consistent with the grower's natural way of thinking than the LP-optimization, where decision making is more or less left to the machine. He emphasizes that it must be the grower who controls the computer as a planning tool. It should not be the computer that tells the grower what to do.

Sowell et al. (1982) developed a Linear Programming model to study optimal production levels for greenhouses producing floricultural crops. The optimal crop mix consisted of the number of production units of each crop that should be produced for sale during a given week. The objective function was the market value of the crops produced minus the operating cost. The cost coefficients for crop decision variables were the selling price of a unit of production minus the cost of all production materials. There were also the hourly cost of unskilled labour. Management was an overhead in their model. Limitations were labour (unskilled, management), the availability of space, market quotas (minimum percentages, upper and lower limits), and other market restrictions (colour). These results were only presented as examples. Sowell et al. (1982) also performed a sensitivity analysis, to study the effects of varying input data on the optimal solution. This showed that in their example the optimal solution was highly sensitive to hourly labour availability.

In Weston & Schumacher's model (1983) each year consists of 26 two-week periods. They use a cyclic model where the periods wrap around from the end of the year to the beginning of the same year. Labour, overhead and fuel are excluded from the costs of a crop. Their model calculates a contribution value to overhead and profit. They emphasize that an LP-model will require considerable care in both formulation and interpretation. The grower may desire to make manual adjustments to the plan. These adjustments will be based on individual growing conditions, market and price adjustments, cost changes, and the fact that growers may reject certain product-mix decisions.

3.3.3 IMAG Production Planning system (IPP)

The *IMAG Production Planning* system *(IPP)* is a tool that was developed for production planning in horticulture (Saedt, 1982; Saedt & Annevelink, 1988; Saedt et al., 1991; Annevelink, 1989, 1992). IPP was developed for pot plant nurseries, but it is also suitable for nurseries with cut flowers and tree nurseries. IPP helps the manager to make an optimal production plan. IPP offered two possible ways of coming to a solution: simulation and optimization. A combination of both was also possible.

Product constraints are given in the IPP *product data* module. In each starting period, a maximum and minimum crop batch size can be specified for each product. Also an overall maximum crop batch size can be given for each year. The maximum crop batch quantities protect a company against the negative effects of fluctuating selling prices. *Company* constraints in IPP are the number of growing compartments, the maximum available space in each compartment in each planning period and the maximum available labour (fixed and extra) in each planning period.

In the *simulation module,* IPP calculates the consequences of a given production plan with respect to space and labour requirements and financial results. A given production plan can have several origins. It can be the manager's proposed production plan, a modification of an existing production plan, or an optimized production plan. An optimized production plan can serve as a good starting point for a number of iterations with simulated, slightly modified plans, until an acceptable production plan is obtained. This approach can take into account various aspects, which are not represented in the optimization model, for example the removal of very small crop batches or converting the crop batches into round numbers. An advantage of simulation is the simplicity of constructing a production plan: for a manager it is completely transparent how the production plan has been created, because he has formulated it himself. However, a disadvantage is that the manager does not know if the simulated production plan is the best alternative, given all constraints and possible alternatives. Furthermore it can be time consuming trying to formulate possible alternatives. Therefore, IPP also offers the possibility of calculating a production plan by optimization.

For the *optimization* option a Linear Programming (LP) model was developed which leads to optimal financial results for a pot plant nursery, combined with high space and labour utilization. The production plan consists of a future plan and a transition plan with a given planning horizon. The *future plan* in IPP is a cyclic plan. This means that the same crop batches will be started each year. The future production plan can be constructed by means of Linear Programming, but it is also possible that the future plan is determined by the manager. It is assumed that space and labour are entirely available for the future plan. However, in a practical planning situation a manager always has to consider the present-day situation in his or her pot plant nursery. In the recent past, crop batches have been started, which will still require space and labour during a certain number of future periods. Furthermore, the future plan will need a certain number of periods to initiate the cyclic rhythm (Figure 3-2). These two facts necessitate another optimization run, which calculates an optimal solution for the remaining available space and labour during the transition between the present-day situation in the nursery and the situation when the future plan is fully operational. In this way a non-cyclic *transition plan* is calculated: unlike the future plan, decisions are not repetitive in the transition plan.

The planning horizon of a transition plan indicates the period in which the future plan becomes completely operational. The choice of a planning horizon depends on several factors. The manager should not choose a planning horizon that is too short, because IPP will not have sufficient freedom to determine an optimal transition. A rule of thumb is to choose a planning horizon that is 1.5 to 2 times the maximum crop duration. In practice a cyclic future plan will never be achieved completely, because it is based upon expected prices which will change over time. When the cyclic rhythm of the future plan has almost been attained, the constraints and the optimization goals will have been altered in such a way that a new (up to date) future plan and a new transition plan will have to be calculated. The future plan is therefore a moving target and a manager's main concern should be to keep the transition towards the future plan optimal.

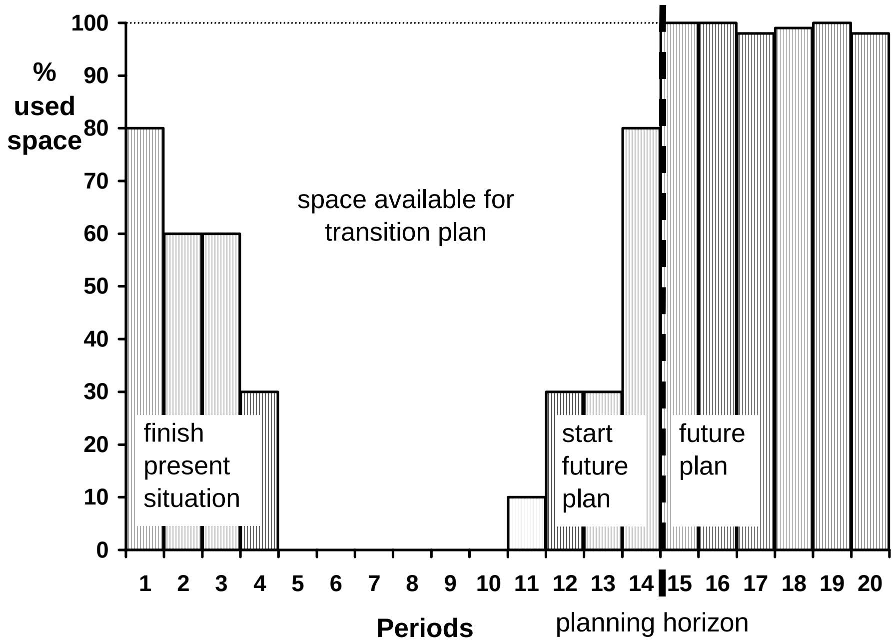


Figure 3-2. Schematic representation of the transition plan as it relates to space available (Annevelink, 1989).

The Linear Programming models for the future plan and for the transition plan have basically the same kind of objective function and constraints. The Linear Programming model maximizes the sum of the gross margins of all crop batches of the production plan minus the costs of extra space and the costs of extra labour. The coefficients of the crop batch decision variables in the objective function consist of expected selling price minus variable costs, such as costs of materials and delivery costs (corrected for losses). Some semi-variable costs, such as energy costs, are not included in the objective coefficients. Hiring extra space is only introduced to prevent infeasible solutions as a result of transition plan calculations. The given present-day situation may cause a violation of a space constraint, due to the space requirements of crop batches that have already been started in the recent past. There is a "Big M"-penalty on hiring extra space. Hence the variable extra space is always zero in the future production plan. Hiring extra labour is possible against a normal hourly wage rate.

Apart from the (non)cyclic property, the LP-models for the future plan and the transition plan differ in two other major ways:

- They have a different set of crop batch periods. The future plan optimization always includes a whole year for each product. The transition plan's set of optimization periods depends on the present period and the planning horizon (Figure 3-2).
- They use different values for the right hand sides. In the transition plan the right hand sides 'available space and labour' and 'permitted production maxima per year' are reduced with quantities corresponding to crop batch decisions, already taken in the present-day situation or in the future plan. These crop batches already use parts of the right-hand side resources during transition plan optimization periods.

The results of the calculations in IPP can be reported in the module production plan. The output report of a production plan contains:

- all crop batch decisions for 39 periods, immediately after the present period;
- a limited financial survey with financial yields, the contribution margins of the products, the costs of hiring extra labour and a plan value;
- a survey of the space utilization for each period in each compartment;
- a survey of the labour utilization for each period;
- a detailed survey of the space and labour utilization of each crop batch for each period.

The result of one run of the Linear Programming model is a production plan which fits the specified constraints, but which may not completely satisfy the manager (for example because the labour pattern is too irregular). After several steps of slightly adjusting the specified constraints, for example available extra labour and re-running the Linear Programming model the manager will reach an acceptable, optimized production plan.

IPP was thoroughly tested in several application experiments, using nursery-specific data. One of the conclusions of these experiments was that preliminary recording of nursery-specific data is essential for the success of IPP. Product-, space-, labour- and financial data should be present and sufficiently reliable for the construction of a production plan. The application experiments also indicated the necessity of external advice in the introduction phase of IPP. Operating IPP (data entry, calculating plans and producing reports) does not give rise to many serious problems, but guiding and assisting a manager when he or she is interpreting a production plan are very important issues in the introduction phase. The manager has to learn how to judge the significance of the outcome of an optimization run and he or she has to realize that a calculated production plan can be modified by altering the constraints (such as the available labour and space or the product mix). Without good advice in the starting phase, and in some cases later, a manager will not be able to use IPP optimally.

3.4 Operational level: control

On the operational level control decisions are made on subjects that influence internal transport in a pot plant nursery in the short term. This short-term character of the operational planning process forces the grower to choose frequently and responsively during the day. Decisions have to be implemented immediately and they will influence internal transport for a period of several hours. During this period the operational plan should be monitored so a decision can be made about when it should be adjusted. Slack et al. (1998) indicate that the lack of time during the control activity makes it impossible to calculate the effects of short-term decisions on all objectives in great detail. Short-term decisions are based more on an understanding of priorities than on extensive calculations. The most important operational planning decisions as far as internal transport is concerned relate to:

- work sequence;
- internal transport sequence;
- parking positions.

The processes of determining the work sequence of the operations, determining the internal transport sequence and choosing specific parking positions for transportable benches are closely interrelated parts of the operational plan, and decisions on these subjects should therefore be brought into concordance .

Decisions on the operational planning level and related to internal transport aim at achieving certain objectives, which will vary from company to company as will the relative weight of each objective. Therefore, a manager should specify company-specific objectives. Possible internal transport objectives are:

- minimal total number of transport movements;
- minimal sum of transport distances;
- high average number of transported pot plants per unit of transport distance;
- maximum compliance with demands made by the production process;
- ease of management and control.

It is important to minimize the total number of transport movements because each transport movement requires time and consequently costs money. The total number of transport movements can be influenced by choosing a good internal transport sequence. The choice of the exact rows and positions for parking transportable benches can also influence the amount of internal transport. Some transport movements are essential (related to potting, sorting, spacing and harvesting) but others (related to reorganization) should be reduced or avoided. Another important objective can be to minimize the sum of the transport distances of all transport movements. The number of pot plants which are transported during one transport movement depends on the density of pot plants on a transportable bench. The aim should be to achieve a high average number of transported pot plants per unit of transport distance. This can be achieved by putting transportable benches with a high density of pot plants in rows at larger transport distances in the back of the growing compartment and with a low density of pot plants near to the working area. Specific constraints given by the production process such as the demand of a certain growing climate, nutrient regime or light intensity must be complied with as much as possible. Demands can vary for each crop batch and for each phase of the crop batch. These constraints can eliminate the availability of some rows for transportable benches of some crop batches or they can imply a zone of preferred rows. The manager should be able to inspect crop batches easily and to take certain cultivation measures effectively like spraying or extra watering. This is best guaranteed when transportable benches of a certain crop batch are clustered near to each other, either in one row or in some adjacent rows. If the transportable benches of a crop batch are scattered all over the growing compartment it is difficult to recognize problems and to take appropriate actions.

3.4.1 Work sequence

A specific operation (potting, sorting, spacing and harvesting) on a group of pot plants, standing on a single transportable bench, can be described as an elementary task which has to be performed (Hax & Candea, 1984; see Section 5.2.1). A crop batch that contains more than one transportable bench can be seen as a job. Several jobs have to be executed according to a certain work sequence. Jobs should be completed on time and workers and machines should be kept busy. The sequence of the jobs on each work station should be scheduled in advance (Janssen, 1987). The work sequence per machine operation determines the order of the crop batches, which have to be handled on the work station on a specific (part of the) day. Most operations lead to internal transport between the growing area and the working area. However, the work sequence does not directly specify the internal transport sequence for moving the transportable benches of each crop batch. This is because the transport sequence also depends on how the work sequences of the individual groups of jobs (potting, spacing, sorting and harvesting) should be mixed during the day and because the sequence of transportable benches can often still be changed in the working area to satisfy the work sequence required. The reorganization of the position of transportable benches is a special kind of operation which causes internal transport within the growing compartment. Managers in Annevelink & Van der Voort's survey

(1995 & 1996) considered it a difficult problem to schedule the sequence of different operations. Operations scheduling was even more difficult in weeks with an irregular number of days, for example when there were public holidays, because this disturbed the normal work routine.

3.4.2 Internal transport sequence

The internal transport sequence determines in what order transportable benches have to be transported during a specific period. A general transport sequence decision could be, for example, that all transportable benches for harvesting have to be transported from the growing compartment to the input buffer of harvesting before the transportable benches of the output buffer of the potting machine can be transported to the growing compartment. However, it will be more likely that the different work sequences and the required future positions of the transportable benches lead to an internal transport sequence which mixes transportable benches from different work sequences. After an internal transport sequence has been determined, a computer will be able to control the automatic transport of transportable benches to indicated positions when automatic guided vehicles are used. For this purpose each transportable bench will have to be identified by a unique code (Van Weel, 1988; Van der Kwaak, 1989).

3.4.3 Parking positions

The current and future positions of transportable benches exert a considerable influence on the amount of internal transport required. Decisions have to be made about the origin of required transportable benches (what positions do they come from) and their destination (to what positions do they have to be moved). The positions of transportable benches in the near future have to be determined on the base of the current position of all transportable benches in the company and the transport movements required in relation to selected and sequenced transportable benches. A decision has to be made about to which row in a growing compartment or to which buffer of a work station in the processing compartment the selected transportable benches have to be transported. The exact parking positions of transportable benches in the growing compartment will not be known in advance and they will vary during the day. Different crop batches have different climate, light, nutrients, water consumption and pest control requirements which influence their position in the growing compartment. This constrains the number of suitable positions for a transportable bench with a certain crop batch.

3.4.4 Control techniques and tools

First, relevant data about internal transport have to be recorded frequently on the operational planning level, to support the control process. These data include the processing time of all required operations (potting, sorting, spacing and harvesting), required transport movements, exact positions of all transportable benches in the rows of the growing area and in buffer rows, space utilization of each row and the status of implementation of current sequences. Data which influence internal transport indirectly as constraints should also be recorded, for example, the climatological circumstances in each growing compartment and the nutrient regime of each row. Recorded data contain valuable information for the control process and can be used to generate new or modified sequences. However, it is not possible to record all required data. Some data have to be estimated. This can cause disturbances because of uncertainties about such factors as customers demand, for example.

In the early nineties, automatic data recording systems with sensors became available in pot plant nurseries and were installed on work stations and internal transport devices (Van Weel, 1991). These systems partly replaced manual data collection. Examples of automatically recorded data are the number of pot plants handled and transport times. It was possible to measure the capacity, number of workers, disturbances and type of pot plants on a potting machine. Automatic data recording enables the complex flow of products in the company to be controlled. Other data like disturbances in the planned growing speed of the pot plants, which influence the expected spacing or harvesting moment still needed to be collected manually. Data collection can be supported automatically by coding systems, keyboard input or recording programmes. The enormous amount of available data should be analysed instantly and converted into information, if it is to be used to optimize the decision process.

Some of the Automatic Guided Vehicles in Annevelink & Van der Voort's survey (1995 & 1996) were equipped with an automatic data recording device, which was able to collect data on transport movements and the placement of transportable benches in the growing compartments rows. In theory these data could be used for control and monitoring internal transport from hour to hour and from day to day. However, in most of the companies surveyed, these data were still insufficiently recorded and not stored in a way that made it possible to use them in any kind of automated control system. Therefore, it was still necessary to determine which data are needed and how it could efficiently be recorded.

Slack et al. (1998) distinguished between simple, visual non-optimizing techniques and mathematical optimization as far as scheduling was concerned. They consider the last approach only suitable for relatively simple applications, because the complexity of the scheduling task is beyond the scope of most optimizing techniques. They suggest a Gantt chart as a simple device for scheduling that provides a simple visual representation of what is scheduled and what really happens. A Gantt chart is not an optimizing tool, but it allows the grower to compare different space allocation plans. An automated version of a Gantt chart for the space allocation problem was developed by Annevelink (1992) for pot plant nurseries with the concrete floor production system. The space allocation plan deals with the allocation of crop batches in a production plan to specific positions on the concrete floor of a growing compartment.

Most pot plant nurseries in Annevelink & Van der Voort's survey (1995 & 1996) did not make an internal transport sequence and they did not determine the parking positions on a daily basis. The growers reserved different destinations for different phases of crop batches: a potting zone, a spacing zone and a harvesting zone. The zone determined the preferred position of the transportable benches in the growing compartments. The boundaries between the zones could vary during the year. A few companies did record the positions of transportable benches in the rows during internal transport, but they did not make a formal schedule of these positions or determined the transport route towards them. However, they did use several *rules of thumb* to control day-to-day internal transport (Table 3-4). These rules of thumb refer to the potting process, spacing process, harvesting process, growing process and other general aspects. Different rules of thumb applied to FIFO and LIFO systems (Section 2.5.5). Each single rule of thumb has a logical reason and a certain effect. However, it is difficult to predict the consequences of a combination of these rules of thumb for internal transport. This is where a simulation could support the grower (Chapter 4).

Leutscher (1995) was one of the very first to pay serious attention to operational management in pot plant production. He simulated the implementation of a given tactical production plan under uncertainty. In the *simulation* two factors, crop growth and price formation, were varied randomly, which leads to deviations from the production plan. Operational decision-making concentrated on the adaptation of the cultivation-schedules to balance production plan and reality. His simulation experiments showed that operational management had a significant impact on the performance of a pot plant nursery.

Based on a broad literature survey, Van Elderen & Kroeze (1994) gave their view on the expected development of operational planning methods. They expected that the most likely approach in the near future would be simulation within the context of a database with historical, current and expected data, and decision support based on optimal solutions from Linear Programming (LP) or Dynamic Programming (DP), heuristic algorithms, expert systems or a combination of these three methods. However, they also state that traditional programming techniques (LP and DP) are scarcely used at the moment for operational planning problems on individual farms (they looked at arable production). These traditional techniques often lead to models of unreasonable size if they have to deal with all short-term uncertainties on the operational planning level.

Table 3-4. Rules of thumb for *operational* planning of internal transport (IKC, 1991; Fang et al., 1992b; Annevelink & Van der Voort, 1995 & 1996).

Category	Rule	Effect
potting process	- influence sequence of TBs[1] in row by the work sequence of potting crop batches;	- TBs with slowly growing crop batches in the back of the row;
	- move TBs to growing compartment immediately when output buffer of potting machine is full;	- avoid waiting time of potting machine;
	- locate potting zone far from processing compartment;	- longest transport distance with highest number of pots per TB;
	- determine exact destination within potting zone based on crop batch type;	- cluster crop batch types in specific rows for management purposes;
spacing process	- move TBs to input buffer of spacing work station early in morning;	- create empty positions in growing compartment for allocating TBs of output buffers;
	- locate spacing zone half way from processing compartment ;	- medium transport distance with medium number of pots per TB;
	- FIFO system: allocate TBs with flowering pot plants of same colour together in same row in last phase before harvesting;	- cluster colours in specific rows for management purposes;
harvesting process	- move TBs to input buffer of harvesting work station early in morning;	- create empty positions in growing compartment for allocating TBs of output buffers;
	- locate harvesting zone close to processing compartment;	- pot plants close at hand and shortest transport distance with lowest number of pots per TB;
	- keep input buffer of harvesting process filled;	- avoid waiting time at harvesting work station and enough supply to choose from;
	- LIFO system: relocate TBs with pot plants ready for harvesting to positions at the front of the rows;	- TBs can be collected faster when they are needed without disturbing remaining TBs;
growing process	- keep large pot sizes in separate zone of the growing compartment;	- cluster pot sizes in specific rows for management purposes;
	- allocate TBs with pot plants of certain growing phase only in rows, with qualified nutrient, water and light conditions;	- avoid growing disturbances or delays;
	- take growing speed of different crop batches into account when determining sequence in row;	- avoid reorganization of TBs in rows during growth;
	- move TBs temporarily to rows with artificial light if needed;	- avoid growing disturbances or delays;

1) TB(s) stands for ‘transportable bench(es)’

Table 3-4. (continued).

Category	Rule	Effect
general	- fill complete row with same crop batch;	- cluster crop batch for management purposes;
	- use empty positions which are created on daily basis for reorganizing sequence of TBs[1];	- anticipate on required transport in the near future;
	- reserve special zone for each phase of crop batch (potting, spacing, harvesting);	- avoid mutual transport disturbances;
	- FIFO system: use special row with high throughput speed for TBs with residues (pot plants which are not ready to be harvested);	- avoid unnecessary movement of TBs in other rows;
	- LIFO system: do not allocate TBs in row where other TBs behind them have to be transported shortly;	- avoid unnecessary movement of TBs in other rows;
	- determine transport priority ranking of different types of operations (e.g. harvesting more important than spacing);	- avoid mutual transport disturbances;
	- keep input buffers of work stations sufficiently filled;	- avoid idleness of work stations;
	- empty output buffers when completely filled;	- avoid idleness of work stations;
	- transport the biggest volumes and the highest frequencies over the shortest distances.	- decrease the average transport distance per pot plant.

1) TB(s) stands for 'transportable bench(es)'

Buxey (1989) states that a practical alternative approach to sequencing is dispatching, whereby jobs are chosen on the basis of some kind of priority rule. Many publications suggest that this approach is of widespread value in the field of job sequencing, and that sophisticated algorithms are only of limited practical interest. He suggests four fundamental reasons for this. First, sequencing is often too difficult for the application of mathematical models. Second, schedules are subordinated to priorities at a higher level. Third, there is always some degree of uncertainty about many aspects of the job shop, for example, machine break-downs and rush orders. And finally, high productivity is not achieved in practice via mathematical calculations but by engineering know-how. Buxey (1989) describes several references that take the approach of combining priority rules with simulation to assess the impact of scheduling decisions in a complex situation. The results of the simulation runs can be animated to provide greater insight.

3.5 A pot plant nursery in practice: test case

General description

Data concerning internal transport in a practical situation were obtained from pot plant nursery ‘Kwekerij De Goede Hoop’ (KDGH) in Honselersdijk. The pot plant nursery provides a test case and data from it are used in subsequent chapters. It was thoroughly analysed by Luijks (1993) and Van der Voort (1995) and some of its general characteristics are given in Table 3-5.

Table 3-5. Characteristics of the professional pot plant nursery with transportable benches ‘Kwekerij de Goede Hoop’ (Luijks, 1993; Annevelink & Van der Voort, 1995 & 1996).

Characteristic		Value
Area (m^2)	working area	2 500
	growing compartments	22 680
	transportable bench (6.20x1.80)	11.16
Capacity of buffer row	output potting	24
(number of transportable benches)	input spacing	35
	output spacing quality 1	31
	output spacing quality 2 and 3	28
	input harvesting	34
	storage harvesting	97
Number of compartments in growing area		6
Number of rows in growing area		26
Number of transportable benches per row		61
Total number of transportable benches		1 800
Accessibility of rows in growing area		LIFO
Accessibility of rows in buffers		FIFO
Number of crop batches per year		1 000
Number of pot plants per year (in millions)		6.5
Mean growing time (in weeks)		9-12
Mean number of transportable benches moved/week		2 000
Estimated mean % of transport	for input and output movements	23-39
	for reorganization	61-77

A schematic layout of the pot plant nursery is given in Figure 3-3. The working area is separated from the growing area. All operations, except for the transport operation, are performed in the working area. Transportable benches are moved between the working area and the growing area by an AGV, which can transport three benches one above the other at the same time. The first five rows in the growing compartment are buffer rows with different capacities for the output of potting (Row 01), output of spacing quality 1 (Row 03), output of spacing quality 2 and 3 (Row 02), input of spacing (Row 04) and input of harvesting (Row 05). The next 26 rows in the growing area can hold 61 transportable benches (Row 06 - Row 31). The total number of transportable benches at the pot plant nursery is 1 800 (including empty benches in stock). The growing area is divided into 6 compartments with (slightly) different climate zones. The rows in the growing area have a LIFO access system, with the AGV driving on the left side of the glasshouse. The largest transport distance of the AGV is from Row 01 to Row 31, which is 192 m. Each new crop batch receives a crop batch number. The mean total number of crop batches per year is 1 000.

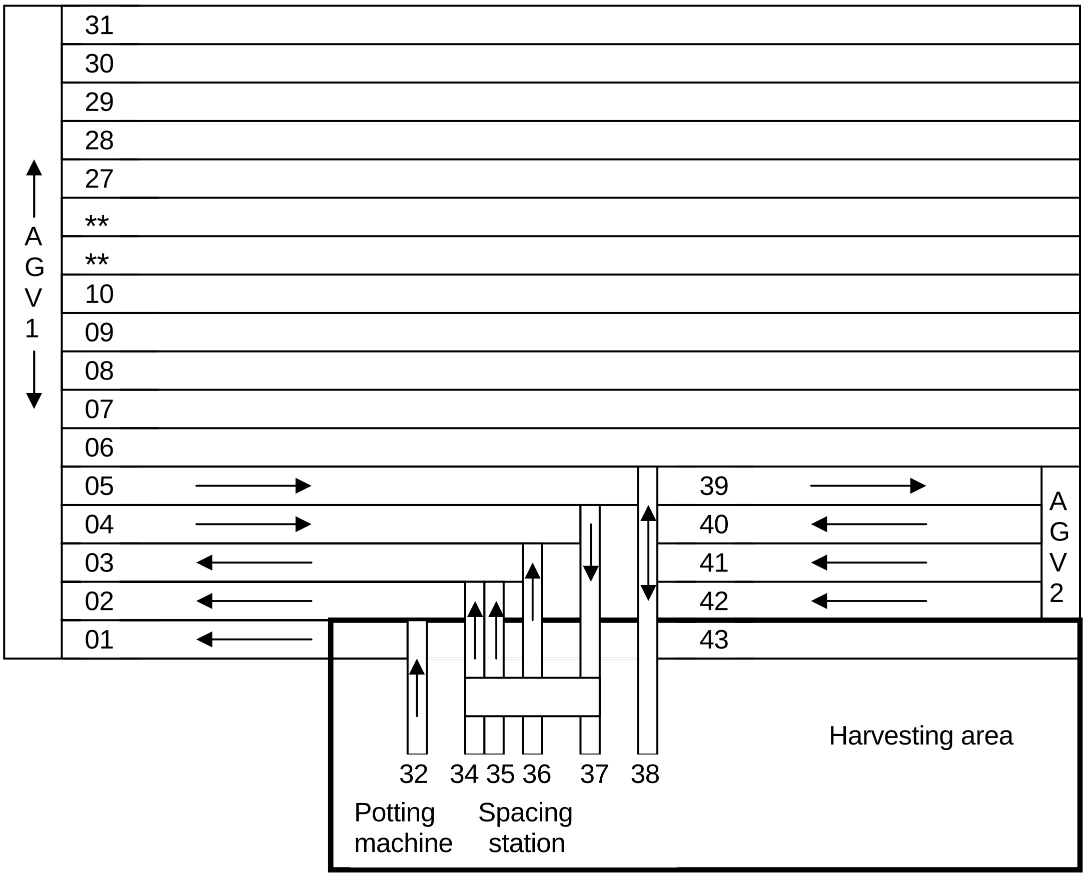


Figure 3-3. Schematic representation of the layout in pot plant nursery 'Kwekerij de Goede Hoop' (Van der Voort, 1995).

Strategic level: design

The pot plant nursery KDGH was designed in 1989 and the actual production of pot plants started in 1990. During the design process three fixed constraints were taken into account: the size of the building plot, the financial means available and the preference for a growing area that was separate from the working area. The pot plant nursery was designed for 6 to 7, year-round species of ferns grown in 8.5 cm pots. Some 100 000 -150 000 pots were produced each week. The calculated amount of internal transport for this global production plan was 9 to 10 hours per day, so it was decided that one AGV would be sufficient. The size of the buffers were adjusted to the same global production plan. The internal transport system was designed first in co-operation with the supplier, and then the glasshouse was built around this system. The choice of a LIFO access system for the rows in the growing area was based on the high space utilization and the lower costs of this system. The lower accessibility of the benches and the higher complexity of the internal transport control process were accepted as disadvantages of this choice that could be overcome. After one year of production, the disadvantage of using a global production plan during the design process became apparent. The production plan had to be thoroughly revised because selling prices were much lower than expected. This resulted in an increase in the amount of internal transport required, for example, because the new production plan contained other pot plant types with different transport requirements and frequencies due to other lengths of the growing phases.

Tactical level: production plan

The grower at KDGH uses the IMAG Production Planning system (Section 3.3.3) each autumn to determine a suitable production plan for the coming year. Depending on developments in the market and large new orders, the production plan will be adjusted two or three times in that year. During the planning process the management team first evaluates the production plan made for the previous year. Then a new production plan is calculated using an iterative process, where the constraints on the production plan are slowly tightened in several steps. The production plan enables the grower to optimize his gross margin, and to achieve a high space utilization. However, an optimal production plan with a highly filled growing area can sometimes cause problems on the operational level with the control of internal transport. A lack of available manoeuvring area can lead to a large amount of extra relocation movements with obstructing transportable benches.

Operational level: control internal transport

The internal transport control activity takes about 3.5 hours per day at KDGH (Van der Voort, 1995). The management team, which consists of two persons, is responsible for scheduling the internal transport. The AGV is controlled with special software enabling the grower to specify different types of transport plans. The main priorities during the day related to the internal transport of benches are:

- keep crop batches together;
- collect the correct benches for the harvesting operation;
- always supply the spacing operation with a sufficient number of benches (to keep the work station busy);
- avoid potting machine waiting time by emptying the output buffer in time.

During the day it is not possible to perform many relocation movements, because these would obstruct transportable benches that have to be collected for the harvesting operation, and lower the flexibility of delivering pot plants to customers. During most of the year, 10 positions at the front of each row are reserved as a supply buffer within the growing area, to increase the flexibility of collecting output benches. Only in summer are crop batches with larger pot sizes grown, and then a crop batch may fill a complete row. The result is that output benches can be collected with less obstructions.

The KDGH grower appreciated the idea of optimizing the internal transport sequence as an option in decreasing the number of transport movements. However, he had no system available to perform such an optimization, so he used the priorities that were mentioned earlier and some other general rules of thumb. Sometimes the work sequence for harvesting is more important than the internal transport sequence, but often the actual sequence of delivering the output benches to the working area does not really matter (certainly within a group of output benches) because the sequence can still be changed by reorganizing the transportable benches in the harvesting buffer (Row 39-43). Sometimes output benches are even transported to the harvesting buffer some days in advance to ensure some buffer supply for rush orders.

Data recording system

The internal transport system at KDGH is largely automated and records are kept automatically of all transport movements and placements of transportable benches in the growing compartment (Luijks, 1993; Van der Voort, 1995). The recorded data were used to verify the simulation model (Chapter 4) and to construct real-scale cases for experiments with the optimization of internal transport sequences (Chapter 7). The recorded data files include:

- the space utilization per row at a standard moment of the day (24:00 hours);
- the actual positions of all of the transportable benches in all of the rows at the same standard moment of the day (24:00 hours);
- all transport movements performed by the AGV during the day;
- all operations performed by the work stations (potting, spacing and harvesting) during the day.

The recorded data were processed for the period February 1994 - May 1998. The mean values per year of all recorded data for this four year period were analysed to obtain a general picture of the factors influencing the amount of internal transport in this specific pot plant nursery.

The mean number of transportable benches per week in the growing area of KDGH is given in Figure 3-4. The growing area consists of 26 rows which can each hold 61 transportable benches, so the maximum number of transportable benches that fits in the growing area is 1 586. The underlying data show that the mean total available free space varied between 54 (3.4%) and 161 (10.2%) empty positions in different rows of the growing area. If all empty positions were to be clustered in rows, then this would mean that approximately 1-2.5 rows would be completely empty. Figure 3-4 shows that the mean number of transportable benches per week changed throughout the year. Periods with a lower mean number of transportable benches per week are mainly the result of major selling moments, like Easter (around week 12), Mother's Day (around week 17), and Christmas (around week 50). Before these favourable selling moments peaks occur in the mean number of transportable benches per week because of extra production requirements.

Figure 3-5 gives the mean number of transport movements per week in the period 1994-1998. The total number of transport movements is divided into three categories at KDGH: input in combination with relocation, output in combination with relocation, and separate relocation (independent of input or output movements). These categories are based on the so-called internal transport plan types, that are used at KDGH to control the automated guided vehicle. Input movements empty the output buffer rows of the potting and spacing operation and output movements supply benches to the input buffers of the spacing and harvesting operation. Input and output movements are combined with the required relocation movements in the same internal transport plan type at KDGH. So unfortunately, it is not possible to get accurate data on the relocation movements as a separate group. However, the group of input movements in combination with relocation was taken as an estimate of the mean number of input movements, thus neglecting the relocation movements. This approximation could be made because input movements often needed little or no relocation movements. This estimate was then doubled under the assumption that each transportable bench that enters the growing area also has to leave the growing area at one time. The estimated mean number of relocation movements was then obtained by subtracting the estimated input and output movements from the measured mean total number of transport movements. The estimated mean number of input and output movements per week varied between 345 and 871 per week with a mean of 625 and the estimated mean number of relocation movements per week varied between 592 and 1 983 with a mean of 1 374. This means that the relocation movements constitute the major part of the internal transport movements: between 61% and 77% per week with a mean of 69%. The mean total number of transport movements per week varied between 937 and 2 807 with a mean of 1 999. The peaks in the mean total number of transport movements per week more or less correspond with the major selling moments already mentioned. However, it should be noted that these peaks are mainly caused by output movements (for harvesting and spacing) in combination with relocation movements. The mean values of the input movements (with relocation) and the separate relocation movements for reorganization are much more stable throughout the year.

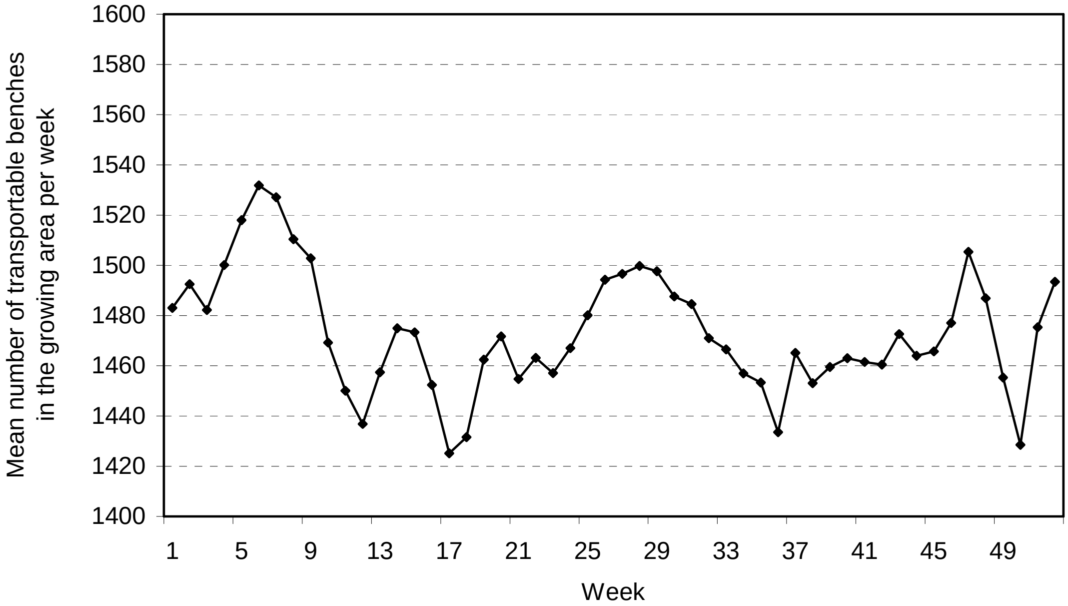


Figure 3-4. The mean number of transportable benches per week in the growing area of Kwekerij de Goede Hoop in the period 1994-1998. The maximum number is 1586 transportable benches.

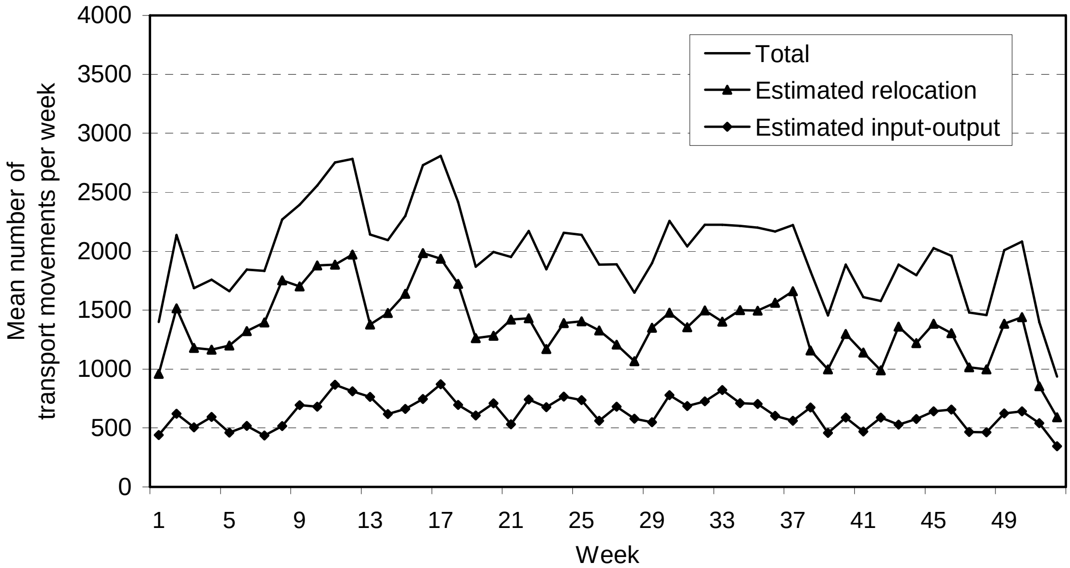


Figure 3-5. The estimated mean number of input and output movements, the estimated mean number of relocation movements and the measured mean total number of movements per week in Kwekerij de Goede Hoop in the period 1994-1998.

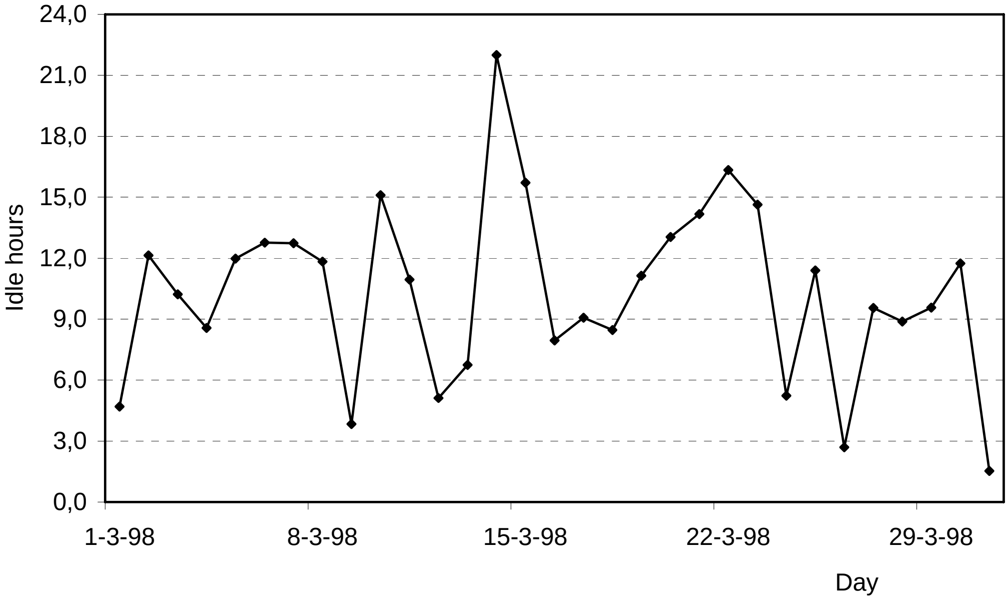


Figure 3-6. The idle time (hours) of the automatic guided vehicle in March 1998.

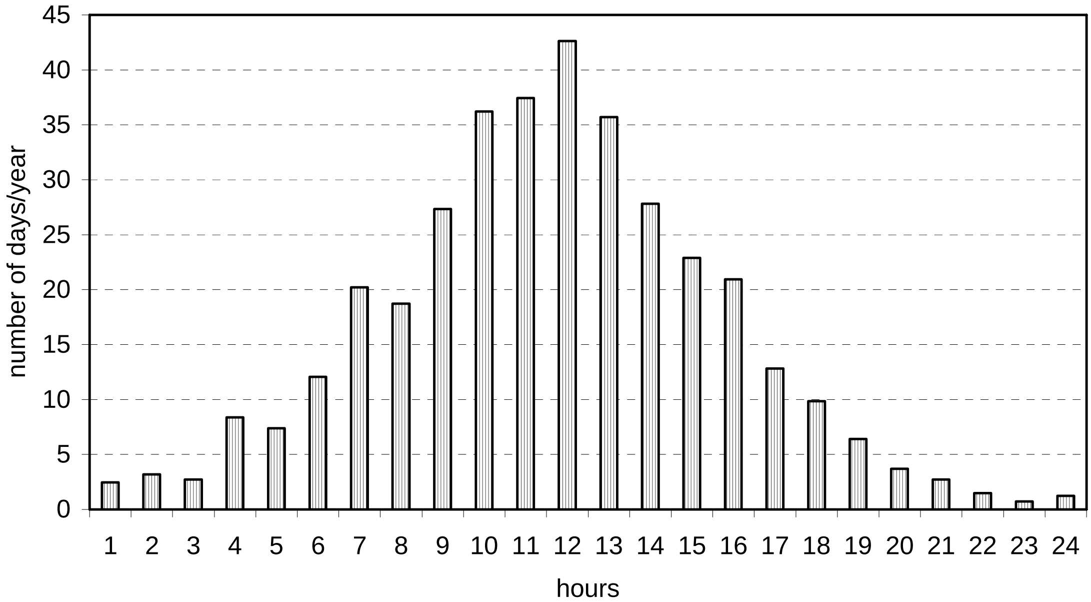


Figure 3-7. The mean frequency in the period 1994-1998 of the number of days per year with a certain amount of idle time of the automatic guided vehicle.

Figure 3-6 shows the time the AGV was idle each day in March 1998. Two peaks can be noticed on a Sunday (15/3/98 and 22/3/98), when the pot plant nursery was closed. However, Figure 3-6 shows that some internal transport also takes place at the weekends. On some days the AGV was almost in full use, for example on 31 March.

On such a day any disturbance in the internal transport process would lead to serious problems in other parts of the production process at potting, spacing, sorting or harvesting level for example.

Figure 3-7 shows the mean frequency of the number of days in the period 1994-1998 with a certain amount of idle time. For approximately half the year, the AGV travels more than 12 hours per day. These data show that the AGV very often works at night, at times when no manager or workers are present at the pot plant nursery to correct anything that may go wrong with it. The fact that the AGV works alone puts high demands on its technical reliability and on its capacity to warn about errors in time so that they can be corrected either automatically or by the grower after he has received an alarm call at home. Furthermore, the internal transport sequence that is responsible for all successive transport movements of the AGV should be absolutely correct otherwise transport problems may occur, such as pushing too many transportable benches into a row which will cause serious damage. Other problems may also occur the next day, for example, if the AGV has been interrupted during the night and no further action has been taken. Input buffers of work stations may not have been filled with transportable benches, for example, and the machines are forced to remain idle because they have no supplies. In the case of the harvesting work station this will seriously delay the delivery of pot plants to customers.

In most cases, KDGH returns relocated transportable benches to their original rows after the required output benches have been removed from a row. The main objective of their approach is to maintain the original sequence of the transportable benches in the rows because their assumption is that such a sequence is a relatively good one. This approach of course leads to a higher number of transport movements than if relocated transportable benches remained in the new parking row. In the research on the simulation model (Chapter 4) and the internal transport sequence optimization techniques (Chapter 7), relocated transportable benches remained in their parking row. This approach was taken to study whether the action of returning the relocated transportable benches can be avoided, thus saving on the number of transport movements. Therefore, the research results cannot be compared directly with the recorded data at KDGH.

The fact that the AGV at KDGH is able to move three transportable benches at the same time causes special sequencing problems because transportable benches are unloaded in the reverse order in a row. These problems were avoided in the research with the internal transport sequence optimization techniques (Chapter 7). This was done by choosing an AGV that can only transport one bench at a time.

4. Simulation with rules of thumb for internal transport

E. Annevelink & A. Vink

4.1 Introduction

Some general remarks about simulation were already made at the end of Section 3.1. Simulation has often been used as a technique in solving planning problems. According to Kleijnen & Van Groenendaal (1992) simulation is one of the most frequently applied techniques in operations research. Fang et al. (1990) used the simulation of the internal transport of transportable benches as a quantitative approach to support a strategic planning problem: the design and choice of a layout appropriate for a pot plant nursery. At the tactical planning level simulation has been used to evaluate the feasibility of a production plan for a pot plant nursery (Reuter, 1989; Elsner & Reuter, 1989). Research concerning the planning of internal transport in pot plant nurseries has mainly been concerned with strategic and tactical planning levels. So far little research has been done in the area of operational planning for internal transport in pot plant nurseries.

Simulation is an appropriate tool for evaluating and analysing rules of thumb used at the operational planning level (Section 3.4.4). Meuleman & Van Weel (1997) state that simulation and visualization of processes is a powerful tool for increased understanding of processes. The results of a simulation model run can show bottlenecks in the performance of an internal transport process. This can indicate where rules of thumb should be changed to improve internal transport. The results can also show whether tactical or strategic decisions should be revised if it is not possible to achieve improvements on the operational level. Therefore, a model was designed and built to simulate internal transport in a pot plant nursery that was equipped with transportable benches and an Automatic Guided Vehicle. The simulation model TRANSIM (TRANsport SIMulation) was initially intended for research purposes, but at a later phase a grower might also be able to use it. In practice TRANSIM could then be applied on a daily basis. Perhaps it could be used more frequently, for example, every time a grower takes a decision about the movement of new transportable benches. The problem was modelled using the simulation language Prosim (Sierenberg & de Gans, 1994). Various components of the internal transport system have been taken into account in the simulation model, such as type of production system, transport device, work stations (potting, spacing and harvesting) and the layout of the compartment.

4.2 Overview of the TRANSIM discrete simulation model

4.2.1 Processing pot plants in the growing compartment

The layout of the pot plant nursery falls into two parts: the growing area with one growing compartment (greenhouse) where pot plants are grown and the working area where all human activities take place. Machines for potting and spacing are located in the working area and the preparation of the pot plants for sale is carried out there as well. When pot plants have to be spaced or sold, the transportable bench is moved from the growing compartment to the working area.

The following operations (processes) were distinguished for use in the TRANSIM simulation model:

- potting plants;
- spacing plants;
- selling plants;
- movement of transportable benches, divided into:
  - input movements (from the output buffer row of the potting or spacing machine to rows in growing compartment);
  - output movements (from rows in the growing compartment to the input buffer row of the spacing machine or for selling);
  - relocation movements (from certain rows in the growing compartment to other rows in the growing compartment).

The implementation of these processes is described in detail in Sections 4.5 (the module ‘Mainmod’) and 4.6 (the modules ‘Contrmod’ and ‘Cartmod’).

4.2.2 Input and output data

Input: potting, spacing and selling dates
In a final, ideal situation, it should be possible to link the simulation to an automated company recording system in order to use required input data automatically. However, for the time being this interface is accomplished through ASCII-files (Figure 4-1), with a format specified by the simulation model. All input files in the experiments were generated from the recorded data of the pot plant nursery KDGH (Section 3.5; Van der Hoorn, 1995). The main input file *Hndldate.udf* (UDF = User Data File) contains data about necessary operations such as potting, spacing and selling, and on six variables that characterize the pot plant nursery and that remain constant during the complete simulation run:

- the maximum number of rows in the growing compartment;
- the length of the rows (all rows are of equal length);
- the maximum number of transportable benches on the AGV;
- the critical length of the output buffer of the potting machine;
- the critical length of the first output buffer of the spacing machine;
- the critical length of the second output buffer of the spacing machine.

In practice, plants are potted, then spaced one or more times, and finally harvested and sold (Section 2.5). Necessary data, therefore, includes the potting date, spacing date(s) and selling date. A complicating factor is that a crop often is not spaced and sold as a whole, but in smaller batches, depending on the size and the quality of the pot plants. Therefore, the standard unit of plants in the program is not a crop, but a crop batch, which is a group of transportable benches of the same crop, which has the same quality, and the same starting and selling date.

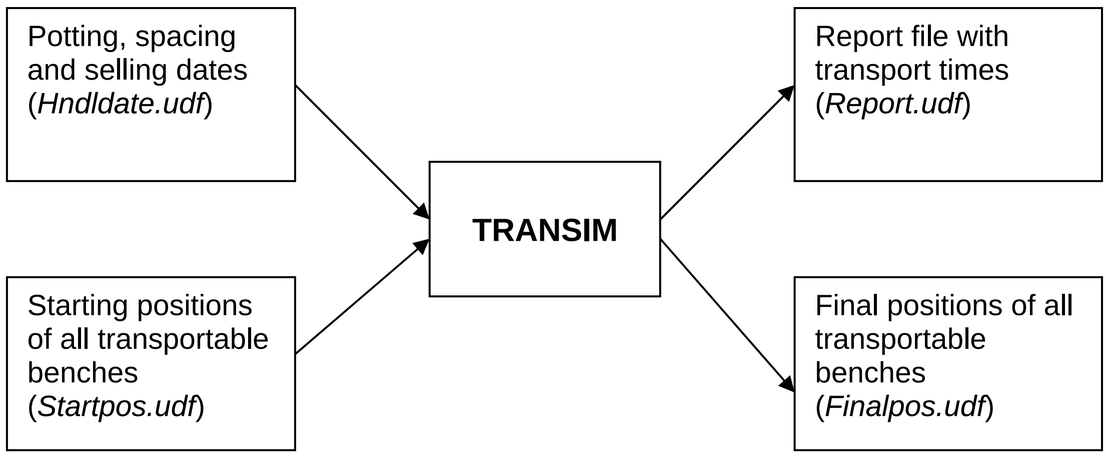


Figure 4-1. Schematic representation of the input and output data files of the TRANSIM simulation model.

In TRANSIM, the total growing time is divided into smaller growing phases, each denoting a period of time during which the pot plants are in the growing compartments. A growing phase starts and ends with a transport movement of the transportable benches of the crop batch to or from the working area. Each crop batch is represented by one line in the data file *Hndldate.udf*, containing the start and the end date, the crop identification, the number of transportable benches in that crop batch, the growing phase (1 = potted, 2 = spaced), the quality grade (1, 2, or 3) and the stage (Table 4-1). The crop batch itself does not have an identification number in the input file, rather the program assigns one. The data file *Hndldate.udf* looks like Table 4-1. A larger example of *Hndldate.udf* is shown in Appendix 4.1. The first line in Table 4-1 contains the following information: one crop batch of crop 3953 is potted on 31-03-97 and six transportable benches belong to this crop batch. When a crop batch has just been potted, it is in Phase 1 and its Quality Grade is 0. Different qualities of the pot plants are sorted later during the spacing process. This crop batch is spaced on 02-05-97, and at that moment the crop batch was split into two smaller crop batches, consisting of eight and four transportable benches. This means that the crop batch in this example is spaced with a Factor 2. Both crop batches are now in Phase 2, which is the phase after spacing. Eight transportable benches carry pot plants of Quality Grade 1, which is the better quality and these are sold a week earlier (on 06-06-97) than the pot plants on the other four transportable benches with Quality Grade 2.

Table 4-1. A description of the input data file *Hndldate.udf.*

Start date	End date	Crop identification	Number of transportable benches	Phase	Quality grade	Stage
31 03 97	02 05 97	3953	6	1	0	20
:	:	:	:	:	:	
:	:	:	:	:	:	
02 05 97	06 06 97	3953	8	2	1	40
02 05 97	13 06 97	3953	4	2	2	43
:	:	:	:	:	:	:
-1						

The crop batch stage depends on the combination of the last operation and the next operation. This information is needed in TRANSIM to control the movement of transportable benches on the animation screen to the correct input buffer row during a simulation run. The different stage numbers that were used in the simulation experiments are based on the situation at the test case pot plant nursery KDGH (Section 3.5). The stage numbers were added automatically to *Hndldate.udf,* when it was generated from the recorded data set of KDGH (Van der Hoorn, 1995). Possible stage numbers are given in Table 4-2.

The order of the crop batches in *Hndldate.udf* determines the exact moment when the crop batch has to be put in the growing compartment. This event occurs either after potting or after spacing. The same crop identification number will appear on different places in the input file because it will be processed (potted and spaced once or more) several times during the production process.

The spacing process is not yet fully implemented in this version of TRANSIM. No direct connection has yet been made between transport from the working area to the spacing machine (managed by the component *Control*) and the spacing procedure (in the global component *Main*). The transportable benches are moved from the growing compartment to the working area on the correct day, but spacing is performed without checking whether the right transportable benches of the crop batch have already been put in the input buffer of the spacing work station. The order in which the transportable benches arrive is not checked either.

Table 4-2. Different stages at pot plant nursery KDGH (Van der Hoorn, 1995).

Stage	Last operation	Next operation
10	not known	spacing
15	not known	selling
20	potting	first spacing
25	potting	selling
30	spacing class 1	extra spacing
33	spacing class 2	extra spacing
36	spacing class 3	extra spacing
40	spacing class 1	selling
43	spacing class 2	selling
46	spacing class 3	selling
50	potting	not known
53	spacing class 1	not known
56	spacing class 2	not known
59	spacing class 3	not known

Input: positions of all transportable benches

When TRANSIM has no data on filling the growing compartment (if only *Hndldate.udf* is available), it will begin with an empty growing compartment. However, an option has been added to start a simulation run with a filled growing compartment. For that purpose a second input file is needed: *Startpos.udf* (Figure 4-1), that contains the positions of all transportable benches in the growing compartment. This way it is possible to run simulations with different starting conditions in the growing compartment. The data lines in the input file *Startpos.udf* look like Table 4-3 (see also Appendix 4.2).

Table 4-3. The description of the input data file *Startpos.udf*

Row	Position in row	Container identification	Next transport date	Crop batch	Phase	Quality grade	Stage
1	1	1560	03 05 97	4192	1	0	20
1	2	349	03 05 97	4192	1	0	20
1	3	648	03 05 97	4191	1	0	25
1	4	234	03 05 97	4191	1	0	25
:	:	:	:	:	:	:	:

Table 4-3 only describes the positions of the first four transportable benches in the first row of the growing compartment. Each transportable bench is represented by one data line in the file, with the rows and positions in the row in ascending order. The container identification is a unique number, needed to keep track of each transportable bench. The next transport date is the date when the transportable bench has to be transported from the growing compartment to the working area for spacing or selling. This date can be used in the simulation model to find the best parking method for the transportable benches in the rows in the growing compartment (Section 4.4). The first two transportable benches in the example belong to the same crop batch, just like the second two. They have all been potted, because they are in Phase 1 with Grade 0. The stage indicates that the next operation will be first spacing for the first two transportable benches (Stage 20) and selling for the other two (Stage 25).

TRANSIM can also supply the final positions of all transportable benches at the end of each day to the output file *Finalpos.udf*. The formats of this output file and the input file *Startpos.udf* are identical, which makes it possible to rename the output file and use it as input for the next simulation run.

Output: report file with transport times

During a simulation run, the situation at the end of each day can be recorded in an output file *Report.udf* (Table 4-4). A larger example of this output file is given in Appendix 4.3. The following data will be recorded to compare simulation runs:

- execution date;
- percentage of the total number of positions in the growing compartment, that are filled with transportable benches (fill %);
- transport time during the whole day given in hours (day time);
- cumulative total transport time from the starting day until the current day given in hours (total time);
- percentage of the cumulative total transport time, that is spent on input movements (input %);
- percentage of the cumulative total transport time, that is spent on relocation movements (relocation %);
- percentage of the cumulative total transport time, that is spent on output movements (output %).

Table 4-4. The description of the output file *Report.udf*

Execution date	Fill %	Day time (hours)	Total time (hours)	Input %	Relocation %	Output %
02 03 94	95.46	17.34	17.34	31.46	40.22	28.32
03 03 94	93.19	14.77	32.11	24.93	47.54	27.53
04 03 94	93.44	14.29	46.41	22.61	53.43	23.96
05 03 94	94.39	7.02	53.43	22.51	55.01	22.48
:	:	:	:	:	:	:

Other data can be recorded automatically during a run of TRANSIM in so-called ‘store streams’ within Prosim. However, these store streams were not studied or saved in the experiments. Other data that could be collected include:

- utilization of the works stations, the AGV, the buffer rows and the rows in the growing compartment;
- movements of transportable benches, total transport distance, transported quantities;
- feasibility of schedules, bottle-necks, adjustments (e.g. amount transported too late).

4.2.3 Layout

Besides data about the crop batches, the simulation model also needs data about the layout of the pot plant nursery: the position of input and output buffers, the transport distances and the length of the rows. The layout of the pot plant nursery is fixed in TRANSIM and cannot be changed by the user. The positions of the rows and the AGV are shown on the layout of the pot plant nursery in Figure 4-2. This specific layout is based on the situation at the test case pot plant nursery KDGH (Section 3.5; Figure 3-3). The first five rows adjacent to the working area at the bottom of the figure are: the output buffer for potting (Row 1), the output buffers for spacing (Row 2 and 3), the input buffer for spacing (Row 4) and the input buffer for selling (Row 5). The other rows (Row 6 to Row 45) are part of the growing compartment. No difference is made between these rows: it is assumed that the growing circumstances (climate, nutrition, light, etc.) are equal in each row of the growing compartment. This is a simplification, because in practice a growing area is divided in several compartments, and transportable benches can only be placed in certain compartments with specific growing circumstances. The maximum number of rows (45) and the maximum number of transportable benches in a row (70) are fixed in TRANSIM. Their actual size has to be specified by the user in the heading of the input data file *Hndldate.udf* (Appendix 4.1). The transportable benches are being transported by an AGV in the main path on the left side of the rows in the growing compartment. Transport distances and transport times between rows can only be changed by the user within the modules of the simulation model and not interactively during a run. The number of buffers and their use as potting, spacing or selling buffer are also fixed in this version of the program. The number of transportable benches that can be loaded on the AGV can be specified by the user. The maximum load of the AGV is three transportable benches. The critical length of the output buffers of the potting and the spacing machine can also be specified. A potting or spacing machine will have to wait when its output buffer is completely filled. To avoid this the AGV will start emptying the output buffer if the number of transportable benches in the buffer exceeds the specified critical length. This is done before any other (buffer) row is handled.

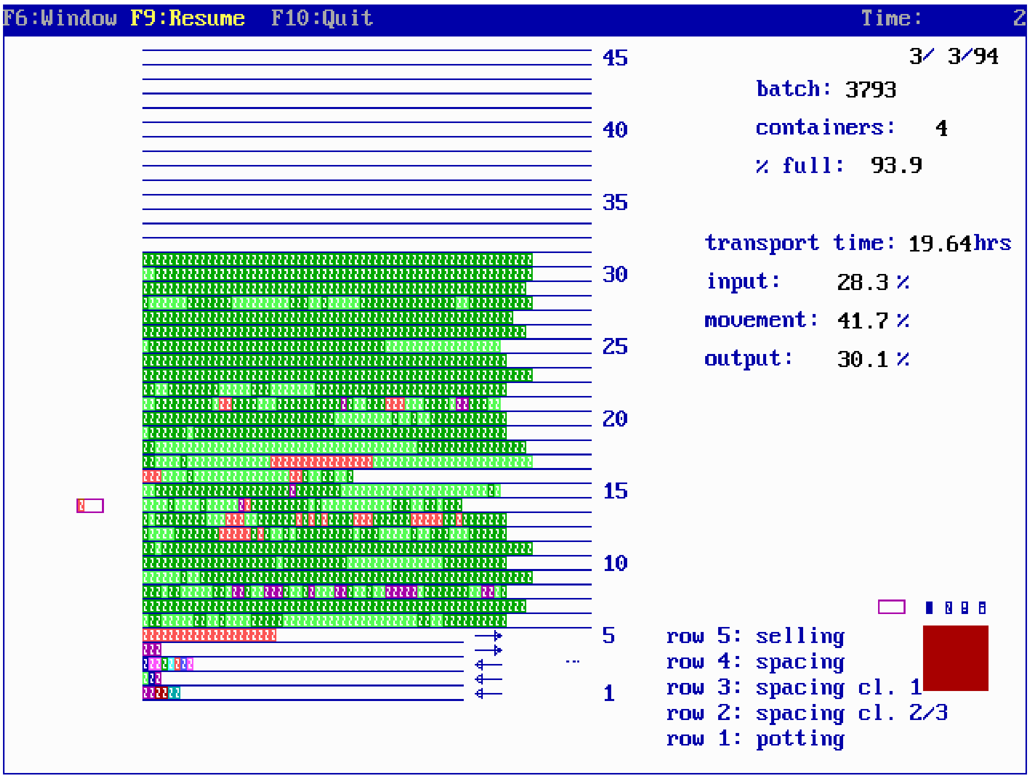
In this version of TRANSIM, it is assumed that the rows in the growing compartment can only be accessed from the left side of the growing compartment, the side where the AGV travels. The rows in the growing compartments have a ‘Last In, First Out’ (LIFO) system of access (Section 2.5.5) for input and output movements. The disadvantage of this system is the necessity of many relocation movements, when required transportable benches are not directly accessible. Transportable benches in the front positions of a row have to be relocated to other rows, to gain access to transportable benches that are located behind them. The buffer rows operate according to the ‘First In, First Out’ (FIFO) system of access (Section 2.5.5). Transportable benches are put in at one side and removed from the other side. The AGV can only travel in a straight line, and it can stop at any row to load or unload transportable benches.

4.2.4 Animation

Contents of the animation screen

In order to obtain visual information about transport movements, animation was added to the TRANSIM simulation model (Figure 4-2). Depending on the size of the simulated system, it is possible to zoom in on the part of the growing compartment, where the movements of transportable benches occur. The animation screen shows the fixed layout of a growing compartment of 45 rows with transportable benches. The first five rows are used as input and output buffers.

The animation screen is flexible (within the limits mentioned in Section 4.2.3) with respect to the number of rows and the number of transportable benches per row. The animation screen does not only show a schematic layout of the pot plant nursery, but it also summarizes the statistics concerning the transport movements. In the upper right corner of the screen the current day number (‘Time’) and the date are presented. Then the number of the crop batch that is being transported at the current moment and the number of transportable benches (‘containers’) in that crop batch are given. The filling percentage of the growing compartment (‘% full’) is also shown. This percentage only concerns the rows in the growing compartment, and not the input and output buffers. Also shown is the cumulative total transport time in hours. This is the total time that the AGV was moving, loading and unloading. The total transport time is divided into input, relocation (‘movement’) and output time given in percentages of the total transport time.



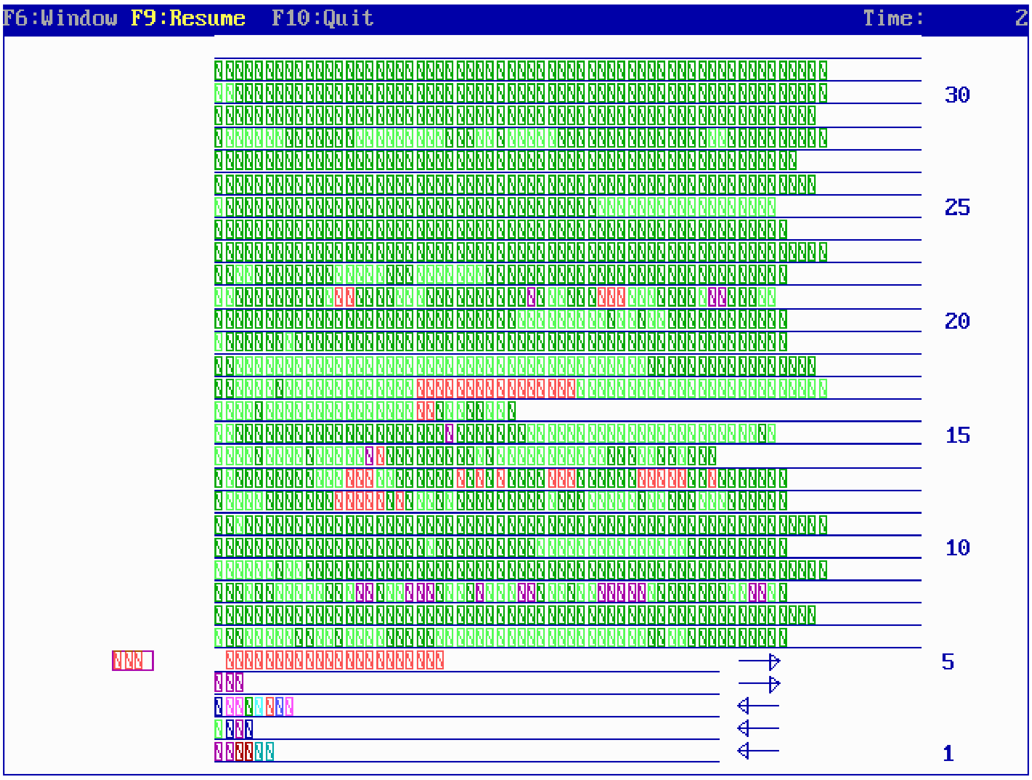


Figure 4-2. Layout of the growing compartment on the animation screen of TRANSIM full screen and enlarged.

Implementation of the animation

The active animation takes place against a fixed background. This background is stored in a separate picture file and has the following elements:

- layout of the pot plant nursery containing a sketch of the growing rows, buffer rows and the potting machine;
- the texts on the screen.

The other elements of the animation (the transportable benches and the statistics on the screen) are created dynamically during the simulation run. These elements are put on the screen using move-statements in the program. To put the transportable benches in the correct positions on the screen, two attributes were added to the definition of the transportable bench: ‘xval’ and ‘yval’. These attributes represent the position were the transportable benches have to be placed on the screen. The ‘xval’ and ‘yval’ attributes are kept up-to-date whenever the transportable bench is moved, so when the transportable bench has to be put on the screen, a simple statement ‘move to xval yval’ is sufficient. A movement can be transportation on the AGV, but also pushing the transportable bench forward or backward in a row or loading and unloading the AGV.

Colours are used in the animation to distinguish between crop batches with specific characteristics. Each crop batch appears with a different colour in the output buffers of the potting machine and the spacing machine. When a transportable bench has been recently added to a row in the growing compartment, it is coloured dark blue. This colour changes to dark green at 0:00 h of the next day. Dark green is the normal colour of a crop batch in the growing compartment indicating that it does not have to be transported for a while. One week before a crop batch is due to be transported to the working area (for spacing or selling) the colour changes to light green. On a specific day of the simulation run, all benches of crop batches that have be transported to the input buffer of the selling work station are coloured red at 0:00 h that day, and all benches of crop batches that have to be transported to the input buffer of spacing are coloured purple at the same time. When it was impossible to transport a crop batch on a specific day, the colour changed to yellow indicating that it was overdue.

4.3 Components and classes in module ‘Define’

Prosim (Sierenberg & de Gans, 1994) was used as computer language in the development of the TRANSIM simulation model. Prosim was used by the Agricultural Research Department (DLO) as the standard simulation tool at the time of this research. Prosim was chosen from twelve simulation tools by a research committee after a broad survey (Projectgroep Evaluatie Simulatie Software, 1989). In Prosim a system is modelled as a set of components. Among these components there are relationships. Kleijnen & Van Groenendaal (1992) use the same terminology. In Prosim, all machines, buildings and people are represented by components and classes. Components represent unique objects, like the growing compartment or the spacing machine. A class contains a group of identical machines or structures, such as the rows in the growing compartment or the transportable benches with pot plants. Components or classes are characterized by their attributes in the Prosim language. Attributes can be of the integer, real, character, macro and reference type. Each class and component contains code that describes one or more processes that are related to it. Kleijnen & Van Groenendaal (1992) classify Prosim as an example of a process-oriented simulation language. They view a process as a series of related events that happen to an entity as it flows through the system. Each process takes time. The component *Potmachine* for example contains code for the process of potting the plants. Each Prosim program must have a component *Main*, that contains global attributes and starts all other processes. It is not possible to assign all processes in a pot plant nursery to machines and buildings. Some processes show interaction or are interdependent. In this case it is best to create a new component, that does not represent a real world structure, but is an abstract object that only controls a certain process. The component *Control* is an example of this kind of component. Some components are empty components (for example *Seller*). These components do not contain any attributes, but are used to start certain parts of the program at pre-set times. Sections 4.5 and 4.6 describe the processes used in TRANSIM and the way they are implemented.

The Prosim source code is divided into several files, called modules. Each Prosim program should contain at least two modules: 'Define' and 'Mainmod', and the user of Prosim is free to add any other necessary module. TRANSIM consists of four modules:

- 'Define' contains all definitions of components, classes and their attributes (Section 4.3);
- 'Mainmod' contains the initiation of attributes, communication with data files, and the processes of the potting and the spacing machine (Section 4.5);
- 'Contrmod' contains procedures to decide in which order the transportable benches have to be transported (Section 4.6);
- 'Cartmod' contains the loading/unloading of transportable benches on the AGV and the movements of the AGV (Section 4.6).

The connection between the components and classes in the model TRANSIM is shown in the Entity Relationship Diagram (Figure 4-3). The texts near the connecting lines must be read clockwise, indicating the nature of the connection.

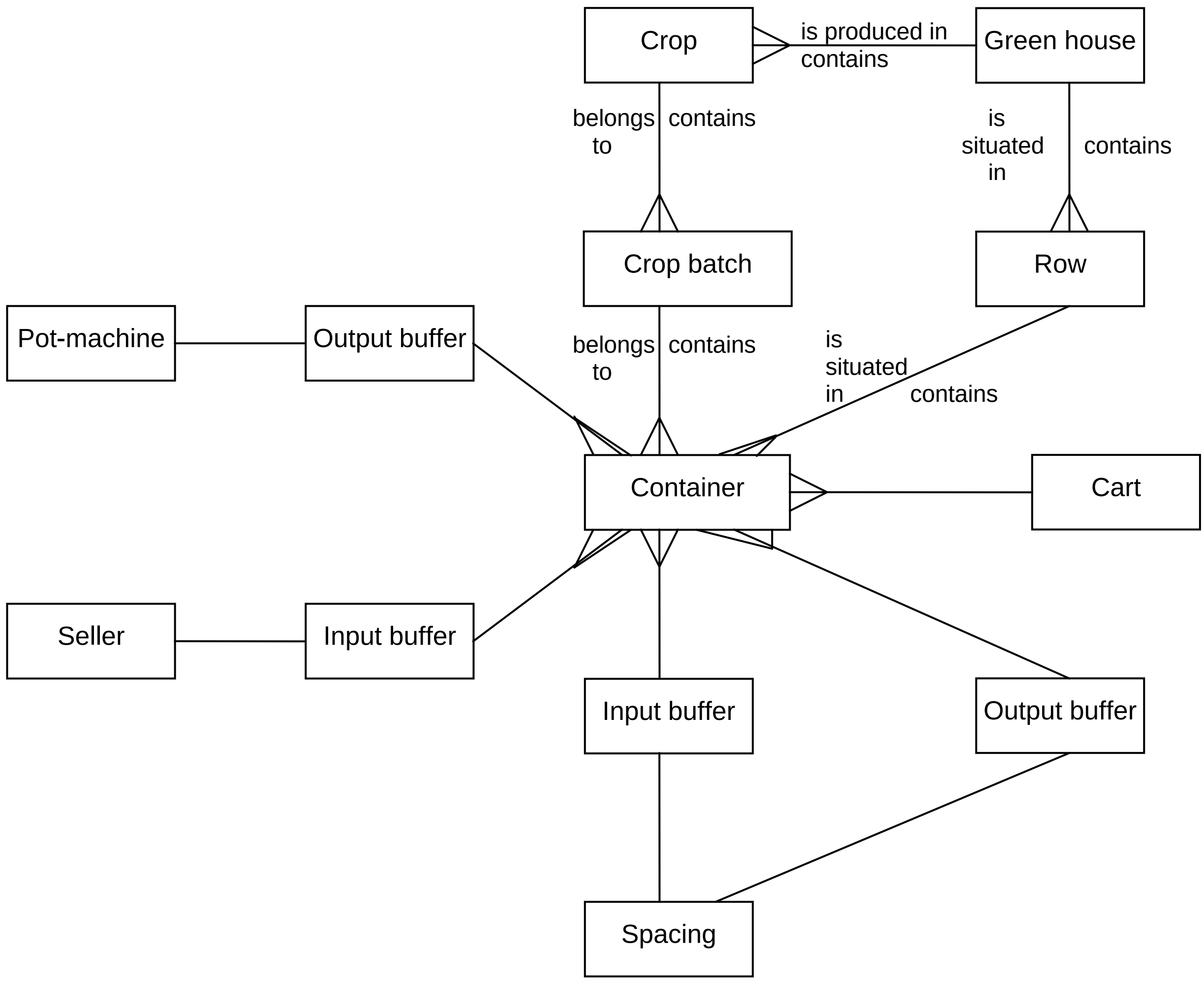


Figure 4-3. Entity Relation Diagram of components and classes in TRANSIM.

Table 4-5 briefly describes the components and classes that are used in the simulation model TRANSIM. A detailed list of all attributes contained in these components and classes is given in Appendix 4.4.

Table 4-5. Components and classes in the simulation model TRANSIM.

Component/ Class	Description
Cart	This component describes the Automatic Guided Vehicle (AGV) that moves the transportable benches.
Container	This class describes the transportable benches on which pot plants are grown and transported. In TRANSIM an instance of this class is created for each transportable bench in the growing compartment. The attributes contain different values for each transportable bench.
Control	This component assigns which transportable benches must be transported to which rows and when this must occur. This component is not directly related to a machine, but created to simulate transport decisions.
Crop	This class describes a crop. A crop contains one or more crop batches and each crop batch contains one or more transportable benches. This way it is possible to pot, space or sell one crop at different dates, which is common practice.
Crop batch	This class identifies a group of transportable benches, belonging to one crop, that may be handled as one unit with the same potting, spacing and selling dates and the same grade and phase.
Greenhouse	This component contains all attributes concerning the general layout of the growing compartment.
Main	This global component is automatically initiated when the simulation program starts. From this component all other components are activated. The attributes declared in Main are used for global operations, that are not directly connected to other components.
Potmachine	This component describes the potting process. It attaches an identification to the transportable benches when pot plants are potted.
Removal	This component is used to start the process of the output movements of transportable benches from the growing compartment to the working area. This component does not contain any attributes.
Row	This component contains attributes concerning the filling and identification of rows in the growing compartment.
Seller	This component describes the selling process. The seller component does not contain any attributes.
Set-date	This component shows the date in the upper right-hand corner of the screen.
Spacing	This component describes the spacing process.
Write-filling	This component creates an output file (*Finalpos.udf*) that contains the positions of all transportable benches in the growing compartment. The component does not own attributes.

4.4 Macro's to describe the rules of thumb for internal transport

Prosim offers the possibility of using macro's as a kind of sub-routine. Two general macro's, 'Daydate' and 'Dateday' are used in TRANSIM to translate the day number of the simulation into a real date and vice versa. Prosim does not know how to handle dates. Internally it uses day numbers. A conversion is necessary because the data that are read from the files, use a date to characterize a day. The macro 'Dateday' is used to calculate selling and spacing days. In TRANSIM, macro's are used to describe the rule of thumb that is used to choose the destination row in the growing compartment for parking the transportable benches of a crop batch. The rules of thumb were derived from the ones that growers use for the day-to-day control of internal transport (Section 3.4.4, Table 3-4). The macro's with rules of thumb are called 'Rule 1' until 'Rule 5' in TRANSIM. The numbers correspond with the options on the opening screen of the simulation program (Figure 4-4). The macro with the chosen rule of thumb is called from the module 'Cartmod', after the AGV has loaded the transportable benches that have to be moved. The macro's with these rules of thumb have some features in common:

- The macro has to determine the destination row for the first transportable bench on the AGV that has to be unloaded.
- If all selection criteria to determine a destination row are equal, the first row in the set 'rows' is selected as the destination row. It is not possible to say in advance in which order the rows are located in the set 'rows' at any moment during the run of the simulation program. Initially, the rows will be in ascending order, so the row with the lowest identification code will be the first row to receive any transportable benches. However, when transportable benches are relocated their row of origin is temporarily removed from the set 'rows'. This is done to guarantee that the transportable benches will be transported to another row and not back to the original row. After moving the transportable benches, the removed row is inserted again in the set 'rows' at the last position. After some of these changes, the order of the set 'rows' is shuffled and consequently the first row in this set can be any row. Therefore the program selects a destination row more or less at random when all selection criteria are equal.
- The rules of thumb are used when transportable benches are transported from the output buffers of the potting and spacing machine to the growing compartment and when transportable benches are relocated.

4.4.1 Macro 'Rule 1': row with lowest number of transportable benches

In this first rule of thumb, the destination row of the first transportable bench on the AGV is the row that holds the lowest number of transportable benches and thus has the largest amount of free positions. If several rows have an equal number of transportable benches, the first of these rows in the set 'rows' is chosen. The positive effect of this rule of thumb is that the rows are filled in an even manner. Every new load of transportable benches on the AGV can be assigned to another destination row, regardless of the row where the first transportable benches of the crop batch were placed. The effect is that crop batches will not be kept together, and that they will be spread over several rows. This can be a disadvantage at the moment when the crop batch has to be transported again to the working area to be spaced or sold. Collecting transportable benches of one crop batch from many different rows will probably require a lot of relocation movements.

4.4.2 Macro 'Rule 2': keep transportable benches of crop batch together

This second rule of thumb is aimed at keeping the transportable benches of the same crop batch together in one row. The program tries to find a row that already contains one or more transportable benches of the crop batch on the first transportable bench on the AGV and then places the load of transportable benches on the AGV in that row. If such a row containing transportable benches of the current crop batch is not found, the transportable benches are placed in the most empty row to achieve an even distribution of the empty space. If the destination row is full after a load of the AGV, another destination row is sought with transportable benches of the current crop batch. If no such row is found, the transportable benches of the crop batch are placed in the most empty row.

4.4.3 Macro 'Rule 3': row with latest transport moments of transportable benches in the row

The third rule of thumb offers a more sophisticated method of determining the destination row. It is actually a combination of more rules of thumb that have been given a certain order of priority. The first part is identical to the second rule of thumb: the preferred destination row is a row, which already contains other transportable benches of the current crop batch. If there is no such row the program tries to find the row with the latest transport moment of the first transportable bench of the row. The transport moment of the first transportable bench in the row should be later than the transport moment of the added transportable bench(es) or the difference between the transport moments should be as small as possible. The idea behind this rule of thumb is that an added transportable bench at the front end of a row should not obstruct standing transportable benches in the row. Therefore, the added transportable bench should be removed again (just) before the moment when the one behind it needs to be transported from the row. If it is not possible to find such a destination row, the transportable benches are placed in an empty row. When the transport moment in all available rows with empty positions is earlier than the transport moment of the transportable benches on the AGV, the load is placed in the row that will remain undisturbed for the longest possible time.

4.4.4 Macro 'Rule 4': row with lowest number of transportable benches and without output movements on current day

The fourth rule of thumb is very similar to the first rule of thumb. The only difference is that all rows in the growing compartment are checked for transportable benches that have to be moved to the working area on the current day. Such rows are excluded from the set of rows that are available as a destination row. Again, the most empty row is chosen from the set of remaining available rows. The intended effect of this rule of thumb is that unnecessary transport on a current day can be avoided.

4.4.5 Macro 'Rule 5': keep transportable benches of crop batch together in a row without output movements on current day

This fifth rule of thumb is the same as rule number two. The only difference is again the elimination of rows with transportable benches that have to be moved on the current day to the working area.

4.5 Description of processes in the module 'Mainmod'

The module 'Mainmod' in TRANSIM contains general processes including initializing the simulation run by reading the required data from the input files (*Hndldate.udf* and *Startpos.udf;* Section 4.2.2) and the activating of components and processes in other modules. Potting machine, spacing machine and selling system processes are also stored in the 'Mainmod' module. It will also produce an output file with the statistics of the simulation run.

4.5.1 Initializing the program

When a simulation run is started, some initial questions have to be answered on the opening screen (Figure 4-4). Some parts of the program will be executed or skipped depending on the answers given to these questions. The default values for the layout of the pot plant nursery are read from *Hndldate.udf*. A new set is created for each row to be filled with transportable benches during the simulation run. The starting date of the first crop is read from *Hndldate.udf.*

```
PROSIM RUNTIME SYSTEM                                    MODEL IS TRANSIM
    Simulation of internal transport pot plant nurseries


Input Rule  3

    1  Row with lowest number of transportable benches
    2  Keep transportable benches of crop batch together
    3  Row with latest transport moment of other transportable benches
    4  Row with lowest number of transportable benches/avoid transport rows
    5  Keep transportable benches of crop batch together/avoid transport rows



    Read transportable bench positions from file? (Y/N) Y
    Write transportable bench positions to file? (Y/N)   N
    Number of days in simulation run                    31

                                                              Rpl
   Enter value                                            <F1:Help>
```

Figure 4-4. Opening screen of the simulation program TRANSIM.

If chosen in the opening screen, filling the growing compartment is obtained from the file *Startpos.udf* by reading the position and the identification of the transportable bench and creating a new transportable bench. The crop number is read and the program checks if this crop number already exists. If it does not a new crop is created. When the phase and grade are read, the program checks if a crop batch with the same phase, grade and selling day already exists. If it does not a new crop batch is created, otherwise the transportable bench is joined to the existing crop batch. The transportable bench is also joined to the set of the correct row, and it is assigned a colour and a shape.

After reading the filling (optional), the first five rows are addressed as input and output buffers for spacing, potting and selling. The file with operations (*Hndldate.udf*) is then read further. The creation dates and selling dates of future crop batches are read and converted to a day number. The phase, grade and stage are read and assigned to every transportable bench in the crop batch. All operations on all crop batches are read from the file at simulation day 0. So the set 'crops' contains all data about all crops, both those that are already positioned in rows in the growing compartment on the first day and the crop batches that will be potted after this date.

The following components are activated from ‘Main’ at simulation time 0: *Removal*, *Control*, *Seller*, *Set_date* and *Write_filling*. Other components are not activated from ‘Main’, but from actions in other components (for example, *Cart* is activated by *Control*).

Prosim does not work with dates internally, but with day numbers. A day must, therefore, be converted (with macro ‘Daydate’, Section 4.4) to a date and then moved to the screen. This is done by the procedure set_date. At the start of each day, the new date is displayed. Because the procedure set_date is run at the beginning of each new day, it is a suitable place to check whether crops remain from previous days that must be processed. This check is performed by testing whether the set ‘crops’ contains any crops. If the set is empty, it means there are no more crops to process and the program will be terminated.

4.5.2 Selecting transportable benches that must be transported on a particular day

Transportable benches that must be transported on a particular day are selected at 0:00 h every day. All crop batches currently in the growing compartment, and the crop batches that are yet to be started, are collected in one large set. The difference between the crop batches is the state variable ‘potted’. The value of this state variable is 0 for crop batches that have yet to be started and 1 for crop batches that have already been potted and stand in the growing compartment or in the buffers. Crop batches are selected for potting when the value of the variable potday equals the current simulation day, and when the value of the variable potted is 0. The variable that distinguishes between potting and spacing is the quality grade. All crop batches that have to be potted today (identified by quality grade = 0) are placed in the ‘potset’. The crop batches that have to be spaced today (identified by quality grade > 0) are placed in the ‘outset’. The potting machine and the spacing machine are then activated.

4.5.3 Potting process

The potting machine (potmachine) is started every day at 0:00 h. The ‘potset’ contains all crop batches that have to be potted on that day. The order of the crop batches in the ‘potset’ is the order in which the crop batches were read from the file *Hndldate.udf*. If ‘potset’ is empty the potmachine will stop. If ‘potset’ contains any crop batches, the first one will be processed on the potting machine. It is assumed that it takes 20 minutes to fill a transportable bench with potted plants. This time is based on the data of pot plant nursery ‘Kwekerij de Goede Hoop’ (Van der Voort, 1995; Luijks, 1993). This filled transportable bench is then joined to the output buffer of the potting machine (Row 1). Each transportable bench is assigned a unique identification number. The program keeps a set with the numbers of all used transportable benches, and always assigns the first unused number to newly filled transportable benches. The maximum number of transportable benches in TRANSIM is 2 400, because of memory limitations in the animation. Each transportable bench in the processed crop batch is assigned the same colour. When the whole crop batch has been processed, the procedure starts again.

4.5.4 Spacing process

The spacing machine is started every day at 0:00 h. The procedure for spacing pot plants is almost the same as the procedure for potting plants, except that the transportable benches are taken from the 'outset' instead of the 'potset'. The contents of 'outset' are not necessarily the same as the contents of the input buffer of spacing, because there is no direct connection yet between the two in this version of TRANSIM. The input buffer of spacing is only used for animation purposes. The duration of spacing for each transportable bench is assumed to be 11 minutes (based again on data from pot plant nursery 'Kwekerij de Goede Hoop') and the transportable benches are put in two different output buffers, depending on their grade. Grade 1 is put in Row 3, and Grade 2 and 3 are put in Row 2.

4.5.5 Selling process

Every day at 0:00 h all transportable benches in both selling and spacing input buffers are removed from the animation screen and from the set of used transportable benches. The crop batches that have to be transported from the growing compartment to the working area that day, are put in one set ('to_shed'). This speeds up the selection of which transportable benches have to be transported next. When the AGV does not succeed in removing all scheduled transportable benches on a particular day, these transportable benches are the first ones to be transported the next day.

4.5.6 Producing an output file with the filling of the growing compartment

Every day at 0:00 h an output file (*Finalpos.udf*) containing the positions of all transportable benches in the growing compartment is updated. The file is emptied before the new filling is written to it, so only the filling of the last simulation day is saved. At the top of the file the creation date is written, and apart from that the rest of the format is equal to the format of *Startpos.udf*. The creation of this output file is optional. It depends on the choice of the user in the opening screen.

4.6 Description of the processes in the modules 'Contrmod' and 'Cartmod'

4.6.1 General description

The module 'Contrmod' is used to choose the crop batches that have to be transported on a certain day and to determine the transport sequence. The control-component in the module 'Contrmod' module determines at what row the next load of transportable benches has to be picked up, and the module 'Cartmod' determines to what row it has to be transported, based on the chosen rule of thumb. As a standard, crop batches are always handled as a whole. When a crop batch consists of more than one cartload of transportable benches, all transportable benches of the same crop batch have to be transported to their new positions before the next crop batch can be handled. This can result in half full loads of the AGV.

The 'Cartmod' module is called from the 'Contrmod' module each time a cartload of transportable benches has to be transported. A transport movement consists of:

- moving the empty AGV to the row in the growing compartment or the buffer row that contains the transportable bench(es) that have to be transported;
- loading the transportable benches onto the AGV;
- moving the AGV, filled with transportable bench(es), to the destination row in the growing compartment or the buffer row;
- unloading the transportable benches at the destination row or the buffer row.

Data from the pot plant nursery 'Kwekerij De Goede Hoop' were used to calculate work times for loading, unloading and transporting (Van der Voort, 1995; Luijks, 1993). In the 'Cartmod' module three types of movement are possible:

- output movements;
- input movements;
- relocation movements.

4.6.2 Output movements

Usually several crop batches have to be moved from the growing compartment on the same day, so first of all a decision has to be made about which crop batch is transported. The output order of the transportable benches of the crop batches is:

- crop batches left from previous day(s);
- crop batches to be sold/spaced today:
  - crop batch with one or more transportable benches at the front of any row;
  - crop batch with one or more transportable benches closest to the front of any row.

If there are any crop batches or transportable benches left in the growing compartment that should have been transported on an earlier day, then these will be transported first. Then crop batches will be transported that have to be spaced or sold now. The transportation order within this group of crop batches is determined by the positions of the transportable benches in the rows. If a crop batch exists that has one or more transportable benches at the front of a row, where they can be loaded by the AGV without having to relocate other transportable benches, this crop batch will be transported first. If such a crop batch cannot be found, the crop batch that has transportable benches closest to the front of a row will be transported. The remainder of the transportable benches from the chosen crop batch will be subsequently transported to the input buffers, taking the transportable benches that are located closest to the front of a row first. However, some of the transportable benches of this crop batch may still be located at the end of a row. This may cause the relocation of all the transportable benches in front of it, even if these transportable benches belong to another crop batch that also has to be transported to the working area.

When all transportable benches in a crop batch have been moved, the next crop batch is chosen in accordance with the same rule. The destination of the crop batches (input buffer of spacing or selling) does not influence the chosen transport sequence. When the transport movements with the transportable benches of the chosen crop batch have been started, the correct input buffer is selected in the animation according to the stage of the crop batch. This selection takes place in module 'Cartmod'. In the module 'Contrmod' a check has been made as to whether the transportable benches that obstructed the crop batch in the output row have all been removed. If this is so, the first transportable bench at the front of the output row will belong to a crop batch that has to be spaced or sold. The AGV travels to the output row and loads transportable benches, until the first transportable bench in the row belongs to another crop batch or until the maximum AGV load has been reached. When the AGV is loaded the AGV is directed to the appropriate input buffer and here the transportable benches will be unloaded. The AGV remains at the buffer until the next load has to be transported. When the AGV is empty, control is returned by the module 'Cartmod' to the module 'Contrmod'.

4.6.3 Input movements

As a rule input movements from the buffers to the growing compartment are done in TRANSIM when all required crop batches have been transported from the growing compartment to the input buffers for selling and spacing. There is only one exception to this transport sequence. The output buffers of the potting and spacing machine have a limited length, which means that the potting or spacing machine will have to wait when their output buffer is completely filled. To avoid this, the AGV will start emptying an output buffer if the number of transportable benches in the buffer exceeds its critical length. The user has entered these critical lengths in the input file *Hndldate.udf*. The AGV will remove complete crop batches from an output buffer and will continue to do so until the output buffer contains less transportable benches than the critical length. By experimenting with this critical length it is possible to find the optimal length of a buffer row. In this case optimal is a balance between the investment costs of the buffer and the variable waiting costs of the potting and spacing machine. If the number of transportable benches in the output buffer is less than the number of transportable benches that can be loaded on the AGV, and if no other benches have to be transported in the growing compartment, the AGV will wait 0.5 h and it will then check the buffer row again for a maximum load.

Input movements from the buffer rows of the spacing machine are always performed after input movements from the buffer row of the potting machine. To start with an input movement, the AGV must be transported from the row where the last transportable benches were unloaded, to the output buffer row of the potting or spacing machine to pick up the transportable benches. The distance (number of rows) and the time between the last stop and the output buffer will be calculated. Transport time of the AGV means that the program ‘works’ for so many seconds. The formula for the calculation of the transport time is:

transport time (s) = 12 + 4.1 * distance with distance > 1
transport time (s) = 45 with distance = 1

These times are again based on data from the test case pot plant nursery KDGH. There is a discontinuity in the transport time, because relocating transportable benches to an adjacent row takes quite a lot of time. The reason for this is that the AGV stays in low gear for this short distance, while over longer distances a higher gear is used.

The identification number of the crop batch that has to be transported is already known from the calculations in ‘Contrmod’. The following procedure applies for each transportable bench that is loaded:

- test whether the AGV is full (cartload equals maximum load). If this is the case the AGV is ready for transport, otherwise proceed to the next step;
- test whether the first transportable bench at the front end of the output buffer row belongs to the right crop batch. This is always the case for the first transportable bench that is loaded, but the next transportable benches in the output buffer row may belong to another crop batch. If the next transportable bench does not belong to the same crop batch, it will not be transported. This means that the AGV will not always travel with a full load. If the transportable bench does belong to the same crop batch, it is loaded from the output buffer

row on the AGV. The time necessary for loading a transportable bench is assumed to be 51 seconds based on data of pot plant nursery KDGH.

As soon as the correct transportable benches are loaded, the AGV can travel to the destination row. The macro 'strategy' is called to assign a destination row. Five possible rules of thumb (strategies) are available for deciding on the destination row and one of them must be chosen in the opening screen (Section 4.5). The program will stop if all rows are completely filled. If this is the case no row can accommodate the cartload of transportable benches. In all other cases the AGV travels to the destination row and the distance and transport time are calculated. When the AGV has stopped at the destination row, it starts unloading the transportable benches. If the destination row is completely filled before all transportable benches are unloaded, another destination row will have to be found for the rest of the load of the AGV. The program repeats the rule of thumb for finding a new destination row. The AGV travels to that row and unloads the rest of the transportable benches. When all transportable benches are unloaded from the AGV, control is returned to the module 'Contrmod'. The AGV stays at the row where it was unloaded until it is called to collect a new load somewhere else.

4.6.4 Relocation movements

The procedure for relocating transportable benches from one row to another within the growing compartment is largely the same as the procedure for input movements, described in Section 4.6.3. The empty AGV is directed to the output row, which is already chosen in 'Contrmod'. The transport time and distance of the AGV are calculated. When it arrives at the output row, the AGV loads as many transportable benches as possible from the same crop batch. Finding a destination row and unloading the transportable benches is done in exactly the same way as in the situation when transportable benches are taken from the output buffers. When the AGV is empty, control is returned to the module 'Contrmod'.

4.7 Simulation with different rules of thumb

The TRANSIM simulation model was used to perform experiments with the recorded data of 'Kwekerij de Goede Hoop' (Section 3.5). The main aim of the simulation experiments was to see how a rule of thumb performed when it was continuously applied over a period of many days. The transport times calculated by the simulation model were taken as a performance measure. Another aim was to compare different rules of thumb, to be able to distinguish between good ones and less effective ones.

Each simulation experiment had the following characteristics:
- name including the date (for example e940301);
- input file with all the potting, spacing and selling dates in the next few months (*Hndldate.udf*, Appendix 4.1):
- input file with the positions of all transportable benches on the day previous to the date on which the simulation run is started (*Startpos.udf*, Appendix 4.2);
- duration of a simulation run which was set to 31 days.

Five different rules of thumb for the choice of a destination row were tested for each experiment:
1. Choose the row with the lowest number of transportable benches (Section 4.4.1);
2. Keep the transportable benches of a crop batch together as much as possible (Section 4.4.2);
3. Choose the row with the latest transport moment of the other transportable benches in the row (Section 4.4.3);
4. Choose the row with the lowest number of transportable benches and without output movements on the current day (Section 4.4.4);
5. Keep the transportable benches of a crop batch together in a row without output movements on the current day (Section 4.4.5).

Experiments were performed with 12 different start dates throughout the year (from 01-03-94 to 03-04-95). Usually a date was chosen somewhere near the beginning of the month. The datafile *Hndldate.udf* contained all the data needed on the required operations (potting, spacing and selling), so the situation was deterministic as far as the arrival of the crop batches during the simulation run was concerned. As mentioned earlier all required crop batches are selected at the beginning of a simulation day. The transport sequence of the crop batches that are scheduled for output movements is determined dynamically by their position in the rows of the growing compartment (Section 4.6.2). The sequence of the crop batches in the output buffer rows of the work stations is determined by the work sequences for potting (Section 4.5.3) and spacing (Section 4.5.4) that were read from the file *Hndldate.udf*. However, it is not known in advance how the transport sequences of the input and output movements will be mixed. This depends on the actual simulation run. Different starting dates of the experiments involved different fillings of the growing compartment. This led to different internal transport situations, that were used to verify the five rules of thumb in varying circumstances.

A simulation run of 31 days meant that all destination rows in the growing compartment were chosen according to the same rule of thumb applied for the whole period. This is a simplification of the practical situation in pot plant nurseries, where the choice of different rules of thumb will probably depend on the state of the growing compartment. However, the objective of the simulation experiments was to test individual rules of thumb. The rule of thumb applied to transportable benches that were moved from the output buffer rows, and also to obstructing transportable benches that were relocated within the growing compartment. During the experiments it became apparent that it was not always possible to complete the whole course (31 days) of the simulation run. The main reason for this was that in some experiments, a specific rule of thumb was no longer able to find a suitable destination row. If this was the case the simulation run was interrupted automatically. When the rule of thumb was applied for several days, the filling situation in the growing compartment and the sequence of the transportable benches in the rows became so complex, that a required transportable bench could not be moved further because there was insufficient manoeuvring space available. In a practical pot plant nursery this situation will never occur, because the grower will switch between different rules of thumb if necessary, whereas the simulation experiments used only one single rule of thumb. The mean last day finished completely before a run was interrupted prematurely, and the mean filling percentage on that day are given in Table 4-6. The data for individual experiments are given in Appendix 4.5 (Table 1 and 2).

Table 4-6. The mean interruption day and the mean filling percentage at that day of experiments with different starting dates and five different rules of thumb.

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
Mean interruption day	22.5	20.9	22.9	22.6	22.2
Mean filling percentage	94.7	94.6	97.5	93.8	94.4

No difference could be found between the five rules of thumb with regard to the mean interruption day of the experiment. The only exception was Rule 2 that was interrupted approximately two days earlier than the other rules. All rules, except Rule 3, were interrupted at approximately the same mean filling percentage of the growing compartment (93.8-94.7). Rule of thumb 3 was the only rule that was interrupted at a higher filling percentage (97.5). This makes Rule of thumb 3 more flexible in a growing compartment with less manoeuvring space.

The transport times of all of the different experiments are given in Appendix 4.5 (Tables 3 to 8). Three different times are given: the total transport time, the relocation time and the sum of the input and output time. The transport times were also indexed to allow a better comparison to be made. Rule of thumb 3 was always given an index value 1.00. In Table 4-7 the mean transport times of all experiments are given and in Table 4-8 the mean indexed values of these experiments are also recorded. The transport times during a simulation run have been given after 7, 14, 21 and 28 days so that experiments that were interrupted after a varying number of days could be compared. Time periods of 7 days are suitable for comparing different rules of thumb, because such intervals always include a weekend. Other time periods will lead to a different number of working days in successive intervals. Table 4-7 was used to produce Table 4-9 with the mean total transport time per day and Table 4-10 with the mean relocation time as a percentage of the total transport time.

Table 4-7. The mean times of all experiments with the five rules of thumb (Appendix 4.5, Tables 3, 5 and 7).

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
Total transport time					
After 7 days	74.03	73.41	58.42	63.56	64.09
After 14 days	179.81	177.83	144.44	157.24	154.74
After 21 days	255.06	241.66	187.75	207.89	206.02
After 28 days	350.94	341.19	260.67	282.69	276.00
Relocation time					
After 7 days	44.49	44.23	29.04	33.96	34.54
After 14 days	119.93	118.22	83.39	96.50	94.33
After 21 days	173.83	160.40	105.89	126.10	124.21
After 28 days	250.66	241.86	159.92	183.31	176.26
Input-output time					
After 7 days	29.55	29.18	29.38	29.60	29.55
After 14 days	59.88	59.61	61.06	60.74	60.41
After 21 days	81.23	81.25	81.86	81.79	81.81
After 28 days	100.29	99.33	100.75	99.38	99.75

Table 4-8. The mean indexed times of all experiments with the five rules of thumb (Appendix 4.5, Tables 4, 6 and 8).

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
Total transport time					
After 7 days	1.29	1.26	1.00	1.10	1.10
After 14 days	1.27	1.25	1.00	1.10	1.08
After 21 days	1.37	1.30	1.00	1.11	1.10
After 28 days	1.37	1.33	1.00	1.10	1.07
Relocation time					
After 7 days	1.69	1.65	1.00	1.26	1.26
After 14 days	1.54	1.50	1.00	1.21	1.17
After 21 days	1.70	1.57	1.00	1.22	1.21
After 28 days	1.66	1.59	1.00	1.19	1.14
Input-output time					
After 7 days	1.01	1.00	1.00	1.01	1.01
After 14 days	0.98	0.98	1.00	0.99	0.99
After 21 days	0.99	0.99	1.00	1.00	1.00
After 28 days	0.99	0.98	1.00	0.99	0.99

Table 4-9. The mean total transport time per day of all experiments with the five rules of thumb.

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
Total transport time					
After 7 days	10.58	10.49	8.35	9.08	9.16
After 14 days	12.84	12.70	10.32	11.23	11.05
After 21 days	12.15	11.51	8.94	9.90	9.81
After 28 days	12.53	12.19	9.31	10.10	9.86

Table 4-10. The mean relocation time as a percentage of the total transport time of all experiments with the five rules of thumb.

	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
Total transport time					
After 7 days	60.1	60.3	49.7	53.4	53.9
After 14 days	66.7	66.5	57.7	61.4	61.0
After 21 days	68.2	66.4	56.4	60.7	60.3
After 28 days	71.4	70.9	61.3	64.8	63.9

4.8 Conclusions

Clear differences can be found between the effects of the different rules of thumb on the relocation time of the transportable benches (Table 4-7). The differences in relocation time fully account for the difference in the total transport time within a particular experiment. The total input-output time of each experiment was hardly influenced by the rules of thumb (Appendix 4.5, Table 7 and 8). This means that no single rule of thumb leads to a large delay of input or output movements and that the differences between the rules of thumb are caused by the number of relocation movements needed before the input and output movements can be performed.

Rule of thumb 3 - Row with latest transport moments of transportable benches in row - had the best performance in almost all of the experiments. Rule of thumb 3 is a combination of several rules of thumb and takes into account the transport dates of other transportable benches in a destination row. After each time period of 7 days, Rule of thumb 3 gave the lowest mean relocation time. Rule of thumb 1 - Row with lowest number of transportable benches - and Rule of thumb 2 - Keep transportable benches of crop batch together - that both do not take into account the transport moment of any of the transportable benches in the destination row, gave the worst performance. The mean relocation time for these two rules was about 50-70% higher than the relocation time of Rule of thumb 3 (Table 4-8). Rule of thumb 4 - Row with lowest number of transportable benches and without output movements on current day - and Rule of thumb 5 - Keep transportable benches of crop batch together in a row without output movements on current day - that both only take into account the transportable benches that have to be transported on the current day, showed a much better performance than Rules of thumb 1 and 2. However, they still had a 14-26% higher mean relocation time than Rule of thumb 3. From these results it can be concluded that it is very important to take into account the future transport movements of other transportable benches currently present in the rows, while choosing a suitable destination or parking row.

The mean total transport time per day varied between 8.3 and 10.3 hours for Rule of thumb 3 (Table 4-9). This was about 2 hours lower than Rule of thumb 1 that led to a mean total transport time between 10.6 and 12.8 hours. It is not possible to really compare the total transport times in the simulation experiments with the actual total transport times of pot plant nursery KDGH (Section 3.5; Figure 3-7) because the grower at KDGH uses different rules of thumb alternately. Still it should be noted that the total transport times obtained with TRANSIM are slightly lower, but in the same order of magnitude as the total transport times of pot plant nursery KDGH. The main aim of controlling internal transport should be to lower relocation time as much as possible. The mean relocation time comprises a high percentage of the total time of all the movements (Table 4-10). After the first 7 days of the simulation run, the mean relocation time was about 50-60% of the total transport time. After 28 days this percentage increased to 60-70%. These percentages are more or less into line with the actual relocation percentages at pot plant nursery KDGH (Section 3.5; Figure 3-5).

When the filling percentage became higher than 97%, none of the rules of thumb were able to prevent the interruption of the simulation run (Table 4-6). At that moment it is impossible to move any further transportable benches according to one chosen rule of thumb, because there is insufficient manoeuvring space. Therefore, better, more advanced rules of thumb or combinations of rules have to be found and implemented in a next version of TRANSIM (Section 8.2).

The results of the simulation experiments show that a grower should take care to select the right rules of thumb for parking transportable benches. The applied rules of thumb strongly influence the amount of internal transport needed. The TRANSIM simulation model proved to be an appropriate technique for analysing and evaluating different, individual rules of thumb. Rules of thumb for other types of pot plant nurseries with different production plans but with the same type of layout (LIFO) can also be evaluated with TRANSIM. However, TRANSIM should be extended to handle other types of access of the rows such as FIFO.

5. Specification internal transport control problem

5.1 Description of the internal transport components

The real scale internal transport control problem is too complex to be solved all at once. Therefore, it is necessary to reduce the complexity of the control problem in order to study the possibilities of different sequencing methods. This chapter will describe a simplified version of the internal transport control problem.

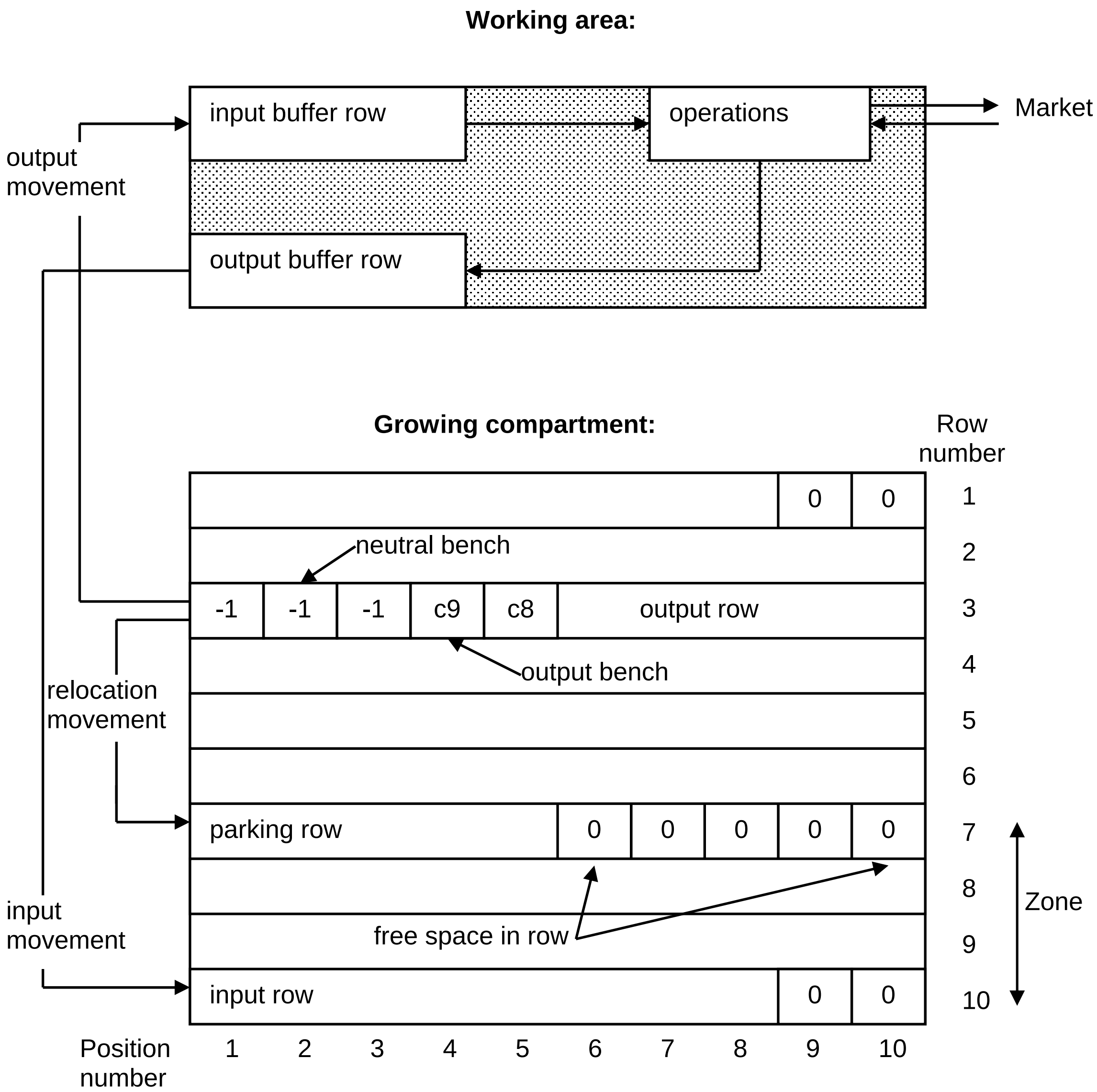


Figure 5-1. Schematic description of the components of the internal transport control problem.

A *growing compartment* in a pot plant nursery (Figure 5-1) contains a certain number of *rows*. Transportable benches can be located on *positions* in a row in relation to the entrance of the row. Position 1 is at the entrance of a row, near the path (situated at the left in Figure 5-1). A row has a certain number of positions for transportable benches, the *row length*. The rows in the growing compartment of the simplified internal transport control problem have a LIFO-access-system (Section 2.5.5), which means that transportable benches have to be inserted and collected at the same end of the row. The main transport *path* is situated on the left side of the rows and the working area is adjacent to Row 1. In the growing compartment three row types can be distinguished. An *output row* contains one or more transportable benches, that have to be moved to the working area or that have to be moved from one row to another row in the growing compartment (during a relocation process). A *parking row* can be used to locate obstructing transportable benches (permanently or temporarily). An *input row* is the destination of transportable benches that are moved from the working area to the growing compartment. A *zone* is a subset of rows in the growing compartment. A *buffer row* is a row type in the working area where transportable benches can be temporarily placed before or after an operation on the pot plants on the transportable bench. An *input buffer row* holds transportable benches before and an *output buffer row* after an operation in the working area.

The transportable benches in the initial situation of the simplified internal transport control problem can be divided into *output benches, input benches* and *neutral benches*. An *output bench* is a transportable bench of a certain crop batch that has to be moved from a row in the growing compartment to an input buffer row in the working area. An *input bench* has to be moved in the opposite direction. A crop batch is identified by a *crop batch code* (c with a positive integer number, for example c1, c2, c3, etc.), which supplies information such as the pot plant type, starting date, and so on. A single crop batch may consist of one or more output benches in the simplified control problem. Output benches can be located on any position in any of the rows of the growing compartment. They can be at the front end of the row, but sometimes they stand behind neutral transportable benches or in the wrong order behind other output benches. *Neutral benches* are not involved in the current control problem, which means that it is not necessary to transport them at the moment. Moving neutral transportable benches should be avoided as much as possible, but it is allowed to relocate them temporarily or permanently when this is necessary in order to perform other movements. A neutral transportable bench (identified by ‘-1’) has not yet received a transport sequence number because it belongs to a future control problem. It can sometimes receive a 'global' number to distinguish between groups of neutral benches, for example, the number of one of the next control problems or the number of the day on which the bench is expected to be moved. Neutral benches fill most positions in the rows of the growing compartment.

A *transport movement* describes the movement of one transportable bench from one row number (output) to another (parking or input). Three types of transport movements can be distinguished: output, relocation and input. An *output movement* means that an output bench is taken from the front side of an output row and moved from the growing compartment to an input buffer row in the working area. A *relocation movement* means that an obstructing transportable bench (neutral or output with a higher transport sequence number) is moved from the front end of the output row to the front end of a parking row. No relocation movements are needed to transport an output bench with the required sequence number to the working area, when it already stands at the front end of the output row. Temporarily parked output benches with a higher transport sequence number can be moved straight to the working area later on, when they are not obstructed by relocated neutral transportable benches during the rest of the transport and parking process. An *input movement* means that an input bench is taken from an output buffer row in the working area and moved to the front end of an input row in the growing compartment. Only one transportable bench can be moved at a time in the simplified control problem. This is a simplification because, in a practical situation, in several pot plant companies, more than one transportable bench can be moved at a time (Appendix 1, Table 9). In that practical situation transportable benches can be collected and delivered at more than one row during one transport movement.

Some rows have empty positions available (identified by ‘0’), where transportable benches can be parked (temporarily or permanently). The *total available free space at t* is the accumulated number of empty positions in all rows of the growing compartment at time t. The total available free space may be randomly distributed over the rows. It does not change through a relocation movement. The only way to change the total available free space is to increase it by an output movement or to decrease it by an input movement. For some groups of simplified internal transport control problems, the total amount of available free space will slowly increase when only output benches are removed from the growing compartment and no input benches are added from buffers in the working area. When a simplified control problem also includes adding input benches, the total available free space may remain more or less stable. The *available free space in a row at time t* is the number of free positions in a specific row of the growing compartment at time t. This number can change through a relocation, input or output movement. The *manoeuvring area* is the set of rows with available free space, which can be used for relocation movements. The manoeuvring area can be expanded with new rows during the internal transport control problem. To achieve this, empty positions have to be created in completely filled rows by relocating neutral benches from these rows to rows of the original manoeuvring area.

The status of all positions in the growing compartment at a certain time t can be described by a *position-matrix* P(t) with elements $p_{ij}(t)$. Each row $p_{i.}(t)$ in the matrix describes a row in the growing compartment and each column $p_{.j}(t)$ describes one specific position in all rows. The *initial situation at t=0* P(0) gives the starting positions of all transportable benches in the rows of the growing compartment. A crop batch code in the position-matrix P(t) indicates an output bench. Neutral benches are identified by '-1' and empty positions by '0'. The position-matrix may contain extra columns that give additional information about the row, like the free space available in the row and the lowest and highest sequence number of output benches in the row.

5.2 Internal transport control problem

5.2.1 Comparison with standard scheduling problems

It is useful to compare the internal transport control problem with standard scheduling problems that are described in the literature (Slack et al., 1998). Hax & Candea (1984) give a classification of job shop scheduling models, that can be applied very well to the internal transport control problem of pot plant nurseries. The internal transport control problem closely resembles a single-machine shop problem. Hax & Candea (1984) define an *operation* as an elementary task that has to be performed. A transport operation is equivalent to transporting one transportable bench of a crop batch to or from the growing compartment, including all additional relocation movements that are needed before the required output or input bench can be transported. A crop batch that contains more than one transportable bench can be seen as a *job*, which is defined by Hax & Candea (1984) as a set of operations that are interrelated by precedence restrictions derived from technological constraints. Each transport operation has a processing time, which is equivalent to the transport time or the number of required transport movements (including relocation movements). The internal transport control problem is a closed job shop where the customers are served from inventory. The production method in a pot plant nursery is to use demand forecasts to grow crop batches in advance and to carry inventories of the product in the growing compartment. A closed shop is different to an open job shop. In the latter products are made to order and the emphasis is put on the readiness of the work force and equipment to absorb fluctuating demands.

Hax & Candea (1984) define *scheduling* as assigning each operation of each job a start time and a completion time onto the time scale of a machine and *sequencing* as establishing the order in which the jobs waiting in the queue in front of a machine have to be processed. The internal transport control problem is an example of a sequencing problem. No start time or completion time is specified for transport operations with output or input benches, but only the sequence of the crop batches that contain these output or input benches.

The nature of the job arrival process of the internal transport control problem can be classified as partly *static* and partly *dynamic*. Some jobs are known in advance at the beginning of the day (static) and some jobs arrive during the day (dynamic). Furthermore, the control problem is a *probabilistic* job shop model, because the job arrival times and the operation processing times are random variables. The job arrival times are not always known because some requests for output movements of benches, especially for harvesting, may arrive during the day. The processing time of a transport operation is unknown in advance because it completely depends on the chosen transport sequence and on the positions of other transportable benches in the growing compartment that may obstruct the transportable benches needed. In most pot plant nurseries, the internal transport control problem is a *single-machine shop*, because all transport movements are performed by only one machine (the AGV). In this case scheduling is equal to finding a sequence of the jobs - containing one or more transport operations - on that single machine.

Hax & Candea (1984) distinguish a number of optimization criteria. They note that costs are not the universal optimization criterion with job shop scheduling models. Schedules are judged by *performance measures* that relate to the individual job or sometimes the entire shop. Examples of performance measures relating to the individual job are completion time, flow time, lateness and makespan. Examples of performance measures relating to the shop are work-in-process inventory and costs. Hax & Candea (1984) emphasize that, in general, the optimization of some job- or shop-related measure will not minimize the total costs. In the internal transport control problem, a shop related performance measure can be chosen that is the required total number of transport movements to complete a certain transport sequence of benches. A further characteristic of the internal transport control problem is a sequence-dependent number of transport movements per job. Some *precedence restrictions* may exist, for example, when the transport benches stand in a certain order in a row. In this case a transportable bench at the entrance of a row should be transported (and thus sequenced) before a transportable bench that stands further back in the row to avoid additional relocation movements. Another precedence restriction may be that certain crop batches have to be transported as a cluster because they have to arrive together at a work station.

The internal transport control problem belongs to the class of permutation schedules. A *permutation schedule* specifies the order in which the benches of the crop batches in the control problem have to be transported. It is a permutation of the crop batch numbers. According to Hax & Candea (1984), sequencing in the single-machine dynamic shop is often done by dispatching procedures, using priority disciplines, which are rules for selecting one of the jobs waiting in the machine queue. These rules are often related to the job processing time. Examples of these rules are Shortest Processing Time (SPT), Shortest Remaining Processing Time (SRPT), Earliest Due Date (EDD) and Shortest Expected Processing Time (SEPT). However, it is difficult to apply these rules to the internal transport control problem, because the processing times (the number of required transport movements) are transport sequence-dependent.

5.2.2 Decomposition into sub-problems

The internal transport control problem of a pot plant nursery consists of determining all transport movements on the operational planning level (Section 3.4). The main control problem is divided into three sub-problems (Figure 5-2) that deal with determining:

- the work sequence (Section 3.4.1);
- the internal transport sequence (Section 3.4.2);
- the parking positions (Section 3.4.3).

These sub-problems are described in more detail in Section 3.4. The interaction between these three sub-problems plays an important role in the whole internal transport control problem. The set of output and input benches is determined in the work sequence problem. Output benches have to be transported to the working area for spacing, sorting and harvesting operations. Input benches are the result of potting and spacing operations and have to be transported to the growing area. The transport sequence partly depends on the chosen work sequence of the output (and input) benches. In some situations a work sequence completely determines the transport sequence. In other cases only a weak relation exists between the work sequence and the transport sequence, because it is possible to interchange output benches when they have arrived in the working area, for example, by moving them around in the input buffer rows. The second situation offers an opportunity for focusing on a good transport sequence that avoids extra relocation movements in the growing compartment, without disturbing the work sequence. The second situation was assumed to be the basic situation in this study. This means that the transport sequence sub-problem was assumed to be more important than the work sequence sub-problem.

The best way to solve the internal transport control problem would be to solve all three sub-problems at the same time with one integrated method. However, it would be very difficult to develop such an integrated solving method. The solving method for the work sequence sub-problem is not taken into account in the simplified internal transport control problem, so the set of output benches is assumed to be a given fact. The two remaining sub-problems are solved with two different methods, a transport sequence generating technique and a parking method. The main control task is to find the best transport sequence for the current given set of output - and input - benches, in combination with determining the best parking positions for obstructing transportable benches. Therefore, the interaction between the parking positions and the internal transport sequence has to be taken into account. To be able to move an output bench from an output row, it may be necessary to relocate obstructing transportable benches from the output row to one or more parking rows in the growing compartment. The required amount of relocation movement is strongly related to both the chosen transport sequence and the parking method. Therefore, there should be an integration between the determining of a transport sequence and the choice of the parking method which determines new positions for all relocated obstructing transportable benches. When the control process is finished, each output bench will have received a *transport sequence number* (s with a positive integer number e.g. s1, s2, s3, etc.) and parking rows will have been chosen for all obstructing benches (according to a certain parking method).

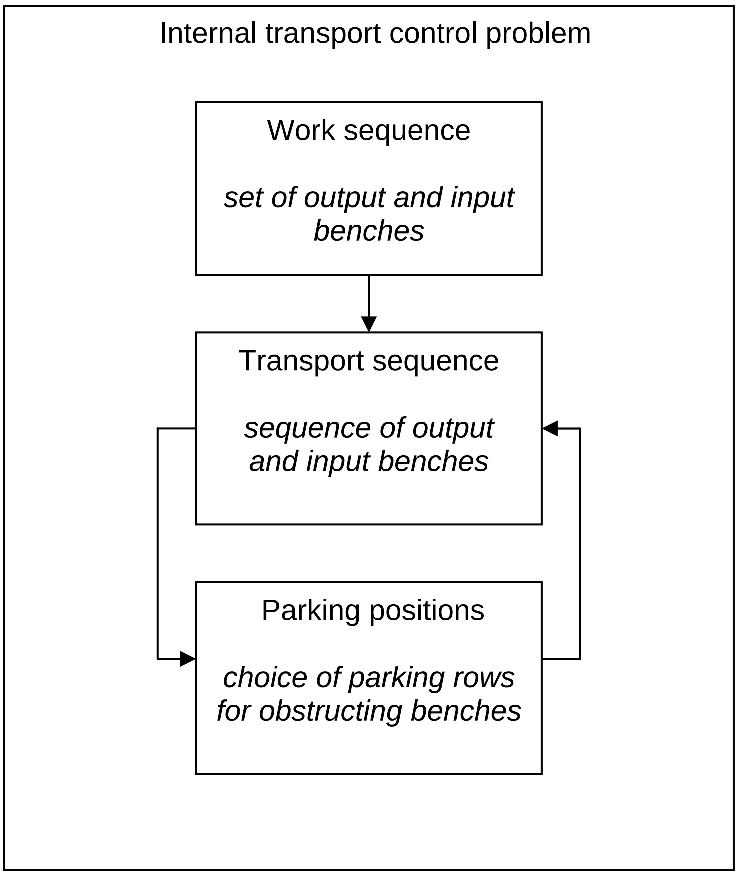


Figure 5-2. The three sub-problems of the internal transport control problem

The initial distribution of transportable benches in the rows of the growing compartment P(0) partly determines the complexity of the internal transport control problem. The internal transport sequence specifies the order of the output benches, that have to leave the growing compartment. It does not determine the exact transport moment of the output benches. A group of transportable benches for one crop batch, for example, can have the same sequence number when the order of transport within the group does not really matter. Sequence numbers do not change when the transport movements have been started.

5.2.3 Interaction between the transport sequence and the parking positions

Several techniques can be used to generate a transport sequence of output and input benches in combination with several parking methods to relocate obstructing transportable benches and to park input benches (Figure 5-3).

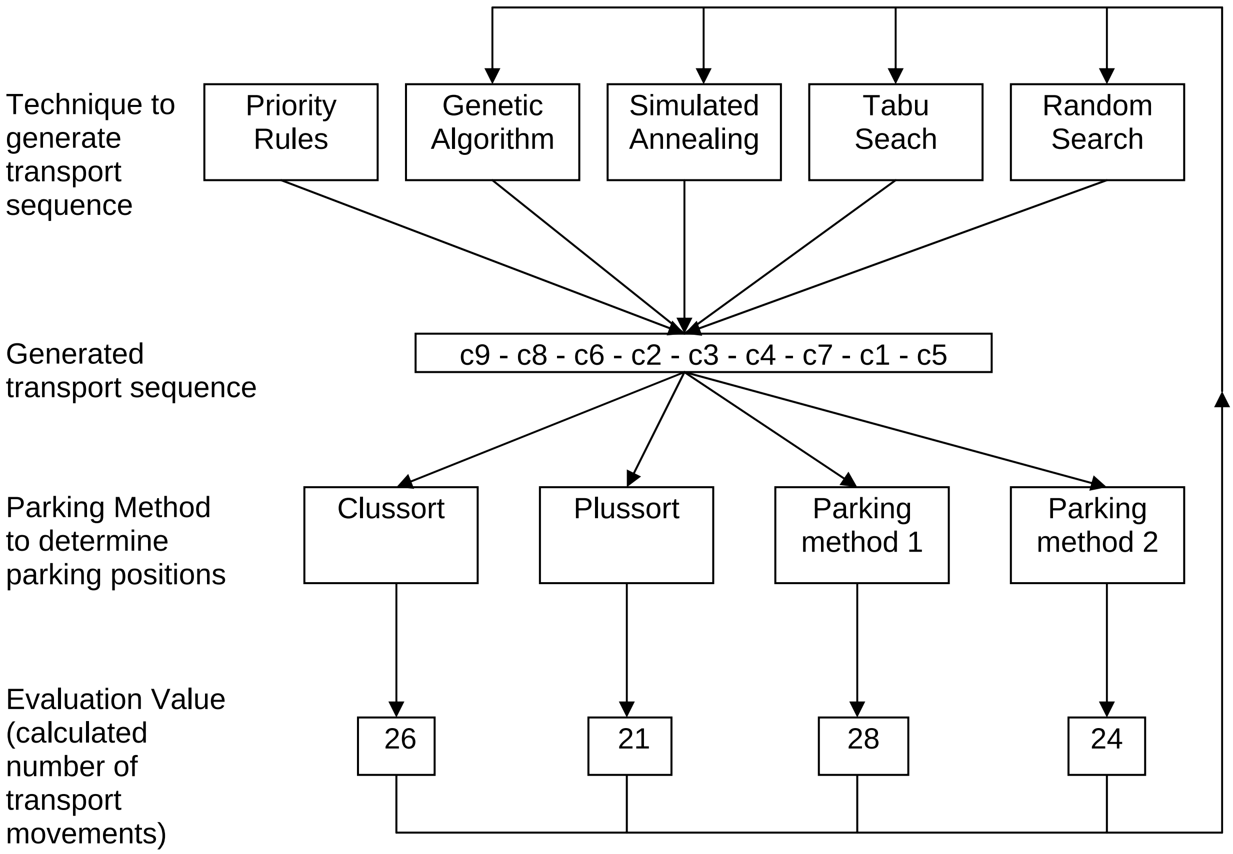


Figure 5-3. The interaction between techniques to generate a transport sequence and parking methods to determine parking positions in the rows of relocated obstructing transportable benches and of input benches. The result of performing a transport sequence in combination with a parking method is a certain number of required transport movements, which is the evaluation value of the combination.

Examples of these techniques are Priority Rules, Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS) and Random Search (RS). A simple transport sequence generating technique will be described in Section 5.3. More sophisticated techniques and how they were implemented to generate transport sequences for the simplified internal transport control problem will be further described in Chapter 6. A sequence generating technique does not determine the positions of relocated obstructing transportable benches and input benches. These positions are chosen by a parking method that guides the relocation movements of obstructing transportable benches by determining the parking rows. Possible parking methods will be described in Section 5.4. Given the transport sequence, the exact number of transport movements depends on the chosen parking positions for relocated transportable benches and input benches. The total number of transport movements (output, input and relocation) constitutes an evaluation value for the combination of the transport sequence and the parking positions (determined by the parking method). The evaluation value will be described in Section 5.5. In the simplified internal transport control problem, the parking method remains fixed during the process of generating alternative transport sequences.

A balance should be found between the effect of a transport sequence generating technique and the parking method (Table 5-1). The main question is whether the most effort should be put into implementing a sophisticated generic sequence generating technique with a very simple parking method or into developing an intelligent problem-specific parking method while spending less time on the sequence generating technique.

Table 5-1. Aspects of the balance between the transport sequence generating technique and the parking method.

Combination of methods	generic value of approach	quality of solution	development time	calculation time
sophisticated transport sequence generating technique & simple parking method	high	sufficiently high	low	high
simple transport sequence generating technique & intelligent parking method	low	high	high	low

An intelligent parking method will often give better solutions with less relocation movements using relatively short calculation times, but it will also take more time to develop and it may be less generic and thus of only limited value for other problems because its parking rules are too problem-specific. An intelligent parking method will limit the search capability and the strength of a sophisticated transport sequence generating technique. When the parking method gives a reasonable but not sufficiently low number of transport movements for both good and bad transport sequences, all these sequences will have more or less the same evaluation value. This would be no problem if the values were equal to or near the optimum evaluation value. However, when these values are all equal to or near a local optimum and when the values do not give any information about the real quality of the sequences - because this is disguised by the intelligent parking method - the transport sequence generating technique will not be able to discriminate sufficiently between these sequences. The most extreme case would be a parking method that is able to find a very low number of transport movements for any possible transport sequence. Sophisticated transport-sequence generating techniques will be easier to develop and more generic than intelligent parking methods, because their approach is less problem-specific. On the other hand a sophisticated sequence generating technique combined with a very simple parking method will take a relatively long calculation time to find a transport sequence with a sufficiently low number of transport movements and it might not find an optimal solution at all because the simple parking method prevents this. So it is essential to find a good balance between the efficiency and the effectiveness of the combination of the sequence generating technique and the parking method. In this research an approach was chosen that puts more emphasis on generic sophisticated transport sequence generating techniques and less on developing intelligent problem-specific parking methods.

5.2.4 Uncertainty about future control problems

In the simplified internal transport control problem, it is assumed that different successive control problems are strictly separated in time. First, all transport movements for one control problem will have to be finished before the next control problem can be solved. In theory, however, it is also possible to look at successive control problems in a more integrated way (for example three control problems with sequence numbers s1-s10, s11-s20, s21-s30 could be considered as one larger control problem with sequence numbers s1-s30). Such a procedure leads to a more complex control problem and, therefore, it was not taken into account in the simplified internal transport control problem. However, an advantage of this integrated approach would be that it creates the possibility of being able to avoid unnecessary transport movements with transportable benches, which are neutral benches in the first control problem, but which will be output benches in the next control problem. Obstructing output benches for the next control problem can be prevented in such an integrated approach. However, an uncertain sequencing factor is the stability of the sequence numbers. The higher the sequence number, the more uncertain it will be whether the transportable bench will really have to be transported in the order indicated. The reason for this is that circumstances may have changed by the time that the transportable bench has to be moved and that its sequence number may also have changed by then.

5.3 Priority rules as a simple transport sequence generating technique

A simple transport sequence generating technique would be to use some kind of priority rule (Section 5.2.1). A priority rule could be based on several characteristics of the output (and input) benches of the involved crop batches. One characteristic could be the expected number of relocation movements of each crop batch. A priority rule could be, for example, that output benches with a lower expected number of relocation movements are to be sequenced early because their transport will involve less obstruction of other output benches and will create more free space for manoeuvring the output benches with a higher expected number of relocation movements. However, it is difficult to establish the exact number of relocation movements needed for each crop batch in advance because these numbers depend on the chosen transport sequence. Another useful characteristic of the transportable benches could be their position in the rows. A priority rule could be that output benches nearer to the front end of the row are sequenced first. This kind of priority rule was used in the experiments with the TRANSIM simulation model (Section 4.6.2) where output benches at the front end of any row were selected first for output movements. Another possibility for a priority rule related to the sequence of the output benches in the row is to empty each output row completely starting at the lowest row number and removing the output benches according to their sequence in the output row. This general priority rule is ‘first output bench encountered sequenced first’.

Different priority rules will give results of different quality depending on the initial situation of the internal transport control problem. A disadvantage of using priority rules is the lack of some kind of feed-back mechanism that tells the priority rule where to adjust unsatisfactory transport sequences. Feed-back is provided when local search methods are used as a sophisticated transport sequence generating technique. This will be discussed in Chapter 6.

5.4 Parking Methods

The chosen *parking method* determines the choice of the parking rows for the obstructing benches during the relocation movements and it also determines the input rows for the input benches of a given transport sequence. The algorithms *Clussort* of Huijbers (1996) and *Plussort* of Janssen (1996) are examples of intelligent parking methods using a flexible combination of several parking rules for a given transport sequence. These combinations of parking rules can be deduced from the rules of thumb that are applied by growers in daily practice (Section 3.4.4). These algorithms try to minimize the number of transport movements for a given fixed transport sequence. These two intelligent parking methods will be briefly described to illustrate the kind of approach where more development time is invested in the parking method. However, no further effort was put into refining and implementing these two intelligent parking methods, because, in this research, emphasis was put on combining sophisticated transport sequence generating techniques with relatively simple parking methods. Therefore, in the experiments presented in Chapter 6 and 7 two simpler parking methods were used, Parking Method 1 ‘first available parking row’ and Parking Method 2 ‘first available parking row without output benches’, that only utilize a single simple parking rule. None of the parking methods try to change this given transport sequence, even when it is impossible to relocate obstructing transportable benches properly.

5.4.1 Parking clusters of transportable benches (Clussort)

The Clussort algorithm was implemented as a separate support tool by Huijbers (1996). The main characteristic of the Clussort algorithm is that it continuously tries to create clusters (groups) of transportable benches during output and relocation movements. The assumption in Clussort is that one crop batch can have more than one output bench with the same transport sequence number. Output benches with transport sequence number s (s = s1 at the start of the algorithm) can be located in more than one row, so Clussort starts to determine all potential output rows. Then one of these output rows has to be chosen, according to certain decision rules. The output benches with transport sequence number s compose an output cluster that will have to be moved out of the growing compartment.

Obstructing output benches in the output row are grouped in relocation clusters with a decreasing order of the transport sequence numbers seen from the entrance of the row (e.g. s6-s4-s3). These relocation clusters are then moved to a suitable parking row, where they are parked in reverse order (s3-s4-s6), so that the output benches can later be taken for their output movement without any further obstruction. A suitable parking row for a relocation cluster is a row that only contains neutral benches and/or output benches with a transport sequence number that is higher than or equal to the transport sequence number of the first output bench of the relocation cluster. Preferably the relocation cluster is moved to a parking row where the available free space in the row exactly matches the size of the relocation cluster.

Obstructing neutral benches are only temporarily moved to parking rows in Clussort. They should not be blocked by relocation clusters because they are only parked there for a short while during the output movements of output benches. Parking rows for obstructing neutral benches should be as near to the output row as possible. Neutral transportable benches are always moved back to their original output row in Clussort before a new output row is selected and dealt with. The reason for returning neutral benches is the assumption that they were already standing in good sequence in the output row. In Clussort, all output benches are removed from the output row including the ones behind the output bench with the required sequence number. If they were to remain in the output row, they would be obstructed by neutral benches after these were returned. So when an output row has been dealt with by the algorithm, only neutral benches remain in the row. The only exception to this rule is the situation where the required output bench is on the front position of the output row. When all output benches with transport sequence number s have been moved out of the growing compartment, output benches with transport sequence number (s+1) are taken out and so on. This process continues until all output benches have been removed.

Clussort produces infeasible solutions when the total available free space is insufficient at a given moment during output and relocation movements. If this is the case, the Clussort algorithm is interrupted and the planner must decide for him or herself. Three actions could be taken: another output row could be chosen (which needs less parking space for obstructing transportable benches), other clusters could be chosen from the output row or the total available free space could be enlarged by temporarily moving transportable benches to another growing compartment. However, this last option means changing the conditions of the original control problem.

5.4.2 Parking neutral transportable benches (Plussort)

Plussort was developed on the basis of recommendations made for the improvement of Clussort (Janssen, 1996). It was only partly implemented in a separate module. The main differences between Plussort and Clussort are described below. In Plussort:

- neutral benches will not be moved back to the output row; they remain in the parking row;
- other output benches that lie behind the requested output bench are not moved out of the output row;
- obstructing output benches are no longer considered as relocation clusters but as individual benches; a suitable parking row is chosen independently for each obstructing output bench;
- neutral benches are no longer considered as one large group (identified by ‘-1’); they are divided into different groups with group numbers (100, 200, 300, etc.), depending on the type of crop, stage of development or the expected day of transport;
- the choice of a suitable parking row for neutral benches with a group number depends on the group number of the first neutral bench in the front position of the parking row.

Neutral benches have been given a group number for reorganization purposes. A group number corresponds, for example, to the expected day of transport. Certainty about this day increases for neutral benches that have to be transported in the near future, and this is reflected in lower group numbers. Neutral benches, that constitute the next control problem (with group number 100), have also been given an estimated transport sequence number within the group, for example, 105, 110, etc. Intermediate transport sequence numbers may be skipped or several neutral benches may receive the same transport sequence number.

A parking row has to be chosen for all obstructing benches (both output and neutral). In general, a parking row must have empty positions and it cannot be the output row. When a parking row is determined for an obstructing output bench the following priority ranking is given to the possible choices:

- choose a parking row which already contains an output bench; the transport sequence number of the relocated output bench is *lower* than the lowest transport sequence number of all present output benches in the row;
- choose a parking row which only contains neutral benches;
- choose a parking row which already contains an output bench; the transport sequence number of the relocated output bench is *higher* than the lowest transport sequence number of all present output benches in the row.

When a parking row is determined for an obstructing neutral bench with a group number the following priority ranking is given to the possible choices:

- choose a parking without an output bench and with a neutral bench at the first position with an identical group number;
- choose a parking row with an output bench somewhere in the row and with a neutral bench at the first position with an identical group number;
- choose a parking row without an output bench and with a neutral bench at the first position with a group number that is higher than the group number of the relocated neutral bench;
- choose a parking row with an output bench somewhere in the row and with a neutral bench at the first position with a group number that is higher than the group number of the relocated neutral bench;
- choose a parking row with all positions empty;
- choose a parking row with an output bench at the first position;
- choose a parking row with a neutral bench at the first position with a group number that is lower than the group number of the relocated neutral bench.

It is not really possible to compare the results of Clussort and Plussort, because the algorithms treat neutral transportable benches differently either returning them or parking them permanently. It is obvious that Plussort will need less relocation movements in the short run than Clussort, because obstructing neutral benches are not moved back to their output row in Plussort. In theory, this could strongly reduce the amount of relocation movements in Plussort to almost half the number of relocation movements in Clussort. However, the reduction of the number of transport movements is not exactly 50%. In Plussort parked neutral transportable benches can again obstruct output benches in their parking row, thus causing additional transport movements with them later. Plussort finds a feasible solution more often than Clussort, and it rejects the advantage of the idea of forming relocation clusters with obstructing output benches. An advantage of Plussort is the reorganization of neutral benches through relocation movements during the output movements of output benches. However, if a row does not contain any output benches in the initial situation, it will not be reorganized, because no obstructing neutral benches will have to be relocated from that row. Also neutral benches behind the last output bench in a row will not be relocated and reorganized. Some improvements still have to be made to Plussort. It should also look at neutral benches in second and other positions in the row. Neither does it take input movements into account yet. Clussort and Plussort both require much more time to develop than the two simple parking methods described in the following sections.

5.4.3 First available parking row (Parking Method 1)

Parking Method 1 is a very simple method that just finds the first available parking row without looking at any criterion other than the availability of free space in the row. It first determines the row with the next output bench according to the given transport sequence. Then all obstructing transportable benches - either neutral or output benches that have to be removed later - are relocated one by one to the first available row with empty positions starting at the row with the lowest row number in the growing compartment and excluding the output row. Relocated transportable benches remain in the parking row, so they are not moved back to the output row. This parking method does not take into account any other output benches in the parking row that might be blocked by the relocated transportable benches.

According to the transport sequence in the example the output bench with crop code c9 (see initial situation in Figure 5-4) has been removed from Row 4 and the three neutral benches that were blocking it have been parked in Row 1 (Figure 5-5) because this was the first row with empty positions. Parking Method 1 neglects the fact that Row 1 already contains two output benches of two crop batches (c1 and c6), that have now also been obstructed by the parked neutral benches. The consequence of this very simple parking method is that it will take three extra transport movements to remove the output bench of crop batch c6 in the next phase of the output process.

	working area									
	-1	-1	**c1**	-1	-1	**c6**	-1	0	0	0
	-1	-1	-1	-1	-1	-1	-1	0	0	0
	-1	-1	-1	-1	-1	**c4**	-1	-1	-1	-1
p	-1	-1	-1	**c9**	**c8**	-1	-1	-1	0	0
a	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
t	-1	-1	-1	-1	-1	-1	-1	-1	0	0
h	-1	**c7**	-1	-1	-1	-1	-1	-1	-1	-1
	-1	-1	**c2**	**c3**	-1	-1	-1	-1	-1	-1
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	c5	-1	-1	-1	-1	-1	-1	-1	-1	-1

Figure 5-4. Example of the initial situation in a growing compartment with crop batches that contain *only one output bench*. Transportable benches that have to be removed are indicated by a crop batch for example ‘c1’. No transport sequence has been determined yet. Neutral containers are indicated with ‘-1’ and free positions with ‘0’.

	working area									
	-1	**-1**	**-1**	-1	-1	**c1**	-1	-1	**c6**	-1
	-1	-1	-1	-1	-1	-1	-1	0	0	0
	-1	-1	-1	-1	-1	**c4**	-1	-1	-1	-1
p	**c8**	-1	-1	-1	0	0	**0**	**0**	**0**	**0**
a	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
t	-1	-1	-1	-1	-1	-1	-1	-1	0	0
h	-1	**c7**	-1	-1	-1	-1	-1	-1	-1	-1
	-1	-1	**c2**	**c3**	-1	-1	-1	-1	-1	-1
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	c5	-1	-1	-1	-1	-1	-1	-1	-1	-1

Figure 5-5. Example of the situation in the growing compartment after relocation of obstructing transportable benches to row 1 using Parking Method 1.

5.4.4 First available parking row without output benches (Parking Method 2)

Parking Method 2 is a bit more sophisticated than Parking Method 1. It can be compared with Rule of thumb 3 in the TRANSIM simulation model (Section 4.4.3). Parking Method 2 first finds the row with the next output bench according to the given transport sequence. Then all obstructing benches are moved one by one. The difference between Parking Method 2 and Parking Method 1 is that they are moved to the first parking row available that does not contain any output benches. Thus, a suitable parking row only contains neutral benches that do not have to be transported during the current control problem. The selection of parking rows also starts at the row with the lowest row number in the growing compartment and excludes the output row. Only if no parking rows without output benches are available, is the first available row with empty positions chosen.

In the example (Figure 5-6) the output bench with crop code c9 (see initial situation in Figure 5-4) has been removed and the neutral benches that were blocking it have not been parked in Row 1 because that row still held two output benches of crop batches (c1 and c6). Therefore, they were parked in Row 2, which was the first suitable parking row because it had no output benches.

	working area									
	-1	-1	**c1**	-1	-1	**c6**	-1	0	0	0
	-1	**-1**	**-1**	-1	-1	-1	-1	-1	-1	-1
	-1	-1	-1	-1	-1	**c4**	-1	-1	-1	-1
p	**c8**	-1	-1	-1	0	0	**0**	**0**	**0**	**0**
a	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
t	-1	-1	-1	-1	-1	-1	-1	-1	0	0
h	-1	**c7**	-1	-1	-1	-1	-1	-1	-1	-1
	-1	-1	**c2**	**c3**	-1	-1	-1	-1	-1	-1
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	c5	-1	-1	-1	-1	-1	-1	-1	-1	-1

Figure 5-6. Example of the situation in the growing compartment after relocation of obstructing transportable benches to row 2 using Parking Method 2.

5.5 Evaluation Value

5.5.1 Total number of transport movements

A generated transport sequence in combination with a parking method can be evaluated by calculating the expected total number of transport movements. In the simplified internal transport control problem, an evaluation value is calculated for output movements of all output benches, for input movements of all input benches and for relocation movements of all obstructing neutral and output benches. It is very difficult to determine in advance the total number of required relocation movements for a specific output bench, because it is not known how many relocated transportable benches will obstruct the current output bench during the relocation processes related to previously collected output benches. The number of obstructing benches depends on the initial situation and on whether the output row with the required output bench has been used as a parking row for relocated benches earlier in the transport process. Additional obstructing benches from other rows may have been added. Besides the number of transport movements, it is also possible to look at other performance measures such as total transport distance, throughput time and utilization of equipment (Section 3.4). However, these measures were not taken into account in the simplified internal transport control problem being dealt with in this study.

5.5.2 Theoretical Minimum (TM)

The theoretical minimum (TM) value provides a lower boundary to the required total number of transport movements necessary to accomplish a transport sequence in a specific control situation. A transport sequence has TM transport movements when it:

- collects the output benches from the row in an optimal sequence;
- relocates all obstructing transportable benches to parking rows, where they do not obstruct remaining output benches of the current control problem and;
- moves all input benches to input rows where they do not obstruct remaining output benches of the current control problem.

In the optimal situation no other relocation movements are needed than those that cannot be avoided. The theoretical minimum number of transport movements can be calculated using the next formula:

$$TM = OB + NB + IB + OO$$

This formula states that the theoretical minimum number of transport movements (TM) of a specific internal transport control problem is the sum of the number of output benches (OB) in the current problem, the number of neutral benches (NB) that obstruct the required output benches in the initial situation, the number of input benches (IB) in the current problem and, finally, the number of obstructing output benches (OO) that must be relocated during the current problem.

In a situation where *each crop batch only contains one output bench*, TM can be determined quite easily. The best transport sequence for the output benches in each individual row is always determined by their relative positions to the front end of the row. The general priority rule is 'first output bench encountered sequenced first'. The output bench nearest to the front end of the row should always be taken first, then the output bench second nearest to the front end of the row and so on. The output bench that is most to the back of the row should always be taken last. All neutral benches that are in between the front end of the row and the furthest output bench in the row have to be relocated and parked somewhere at least once. This means that the minimum number of transport movements needed for a row can be calculated by counting the number of transportable benches from the front end of the row to the furthest output bench in the row. TM can be calculated by adding the minimum number of movements of all rows. In this situation, no output benches have to be relocated so the value of OO in the formula of TM is always zero.

In the example given in Figure 5-4, the theoretical minimum number of transport movements to achieve the output movement of the two output benches of two different crop batches (c1 and c6) from the first row is six transport movements. Two output benches have to be moved to the working area and four obstructing neutral benches have to be relocated to another row. However, the theoretical minimum can only be achieved if it is possible to relocate the obstructing neutral benches to parking rows where they do not obstruct other output benches that still have to be moved. The total TM of the case in this example is 24 transport movements.

It is more difficult to calculate TM in the situation where a *crop batch contains more than one output bench* with the same transport sequence number. The output benches of the same crop batch may be distributed over several rows in this type of control problem. This may lead to an initial situation P(0), for example, (Figure 5-7) where an output bench of crop batch c1 is at the front end of one row with an output bench of another crop batch c2 behind it and where two output benches of crop batch c2 are at the front end of another row with an output bench of crop batch c1 behind them. In such a case, in any transport sequence it is impossible to avoid obstructing output benches of either crop batch c1 or c2 having to be parked temporarily, because they are obstructing the other output benches. In this example, at first glance, a transport sequence with c2 before c1 looks better (see Row 1 and 4), when no other crop batches are considered. However, it is difficult to establish

	working area									
	-1	-1	**c1**	-1	-1	**c2**	-1	0	0	0
	-1	-1	-1	-1	-1	-1	-1	0	0	0
	-1	-1	-1	-1	**c1**	**c1**	**c4**	-1	-1	-1
p	-1	-1	-1	**c2**	**c2**	**c1**	-1	-1	0	0
a	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
t	-1	-1	-1	-1	-1	-1	-1	-1	0	0
h	-1	**c5**	-1	-1	-1	-1	-1	-1	-1	-1
	c4	**c4**	**c4**	**c2**	**c5**	-1	-1	-1	-1	-1
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	c3	-1	-1	-1	-1	-1	-1	-1	-1	-1

Figure 5-7. Example of the initial situation in a growing compartment with crop batches that contain *more than one output bench*. Transportable benches, which have to be removed are indicated by a crop batch for example ‘c1’. No transport sequence has been determined yet. Neutral containers are indicated with ‘-1’ and free positions with ‘0’.

how many obstructing output benches in total will have to be temporarily parked because this depends on the other crop batches that are also distributed over several rows. For example, crop batch c4 should be transported before crop batch c2 (see Row 8), but c4 should also be transported after crop batch c1 (see Row 3), resulting in a conflicting sequence of c4-c2-c1. In the example provided in Figure 5-7, the theoretical minimum number of transport movements to achieve the output movement of the two output benches (c1 and c2) from the first row is also six transport movements. However, it is more difficult to establish how many obstructing output benches must be relocated at least once. The total TM of the problem in this example is ‘27 + OO’ transport movements and the number of obstructing output benches that must be relocated (OO) is difficult to establish at first sight. The minimum number of obstructing output benches (OO) that must be relocated in the example is 2. This value is obtained for the transport sequence c1-c4-c2-c5-c3, where c1 is now sequenced before c2. So the total TM in this example is 29 transport movements.

The question whether TM can be obtained in a certain control problem depends mainly on the chosen parking method in combination with the distribution of the available free space in the rows.

6. Local search methods for the optimization of internal transport sequences

6.1 Introduction

Optimization of the internal transport control problem is concerned with determining the best (or at least a good) transport sequence for a specified number of output benches in combination with a specific parking method (Section 5.2). More complex internal transport control problems also include the optimization of the transport sequence of input benches mixed with the transport sequence of the output benches. The aim of the optimization is to minimize the total number of transport movements (output, relocation and input) within the current transport control problem. A global minimum number of transport movements has to be found among a large number of local minima. Optimization of the internal transport sequence is a very complex combinatorial optimization problem. Glover & Greenberg (1989) state that a combinatorial explosion is usually encountered in those situations where choices are sequentially compounded, and this leads to an enormous number of possibilities for choosing alternative routes in a solution network. This is certainly the case for the internal transport sequencing problem, where each crop batch can be chosen on any position in the sequence and where this choice will consequently influence the number of transport movements of all other crop batches that still have to be sequenced after the chosen crop batch. The internal transport sequencing problem most likely belongs to the class of NP-hard or NP-complete problems. Although no formal proof of this statement is given in this study, many other scheduling and sequencing problems have been proven to be NP-complete (Lawler et al., 1993). An exact solution to an NP-complete problem requires an amount of time which grows exponentially with the problem size (Aarts & Korst, 1989; Looi, 1992; Reeves, 1993). This means that an optimal solution cannot be obtained for an NP-complete problem within a reasonable amount of computing time. The class of exact *optimization algorithms* is less suitable for solving large NP-complete problems (Aarts & Lenstra, 1997; Lawler et al., 1993). It is difficult to find an optimal transport sequence within an acceptable time with traditional optimization techniques like Linear Programming (LP) or Dynamic Programming (DP). Complete enumeration methods certainly take too much time because they examine every possible internal transport sequence (Dorn, 1995). Therefore *approximation algorithms* (heuristic algorithms) have been developed that find near-optimal solutions in a reasonable amount of computing time (Looi, 1992; Dell' Amico & Trubian, 1993; Lawler et al., 1993). Approximation algorithms guide the exploration of the search space to a good solution that is not necessarily optimal (Dorn, 1995).

A *local search method* is an example of a heuristic approach that starts with an initial solution generated by some kind of algorithm and then tries to improve this solution by 'local' modifications. When an improved solution is found, it replaces the current solution and the search process continues in the neighbourhood of the new solution. Other names in the literature for this concept are 'stochastic technique', 'repair-based approach', 'iterative improvement', 'perturbation technique' or 'neighbourhood search' (Dorn, 1995; Aarts & Van Laarhoven, 1989; Aarts & Lenstra, 1997). According to Dorn (1995) the main components of a local search method are a representation scheme for a sequence, an evaluation function that assigns a value to such a sequence, a set of operators that can be applied to modify a sequence and finally an algorithm to control the search for improvements.

A disadvantage of a simple local search technique such as, for example, a descent algorithm, is that it can easily be trapped in a local optimum that may be far from the global minimum (Aarts & Van Laarhoven, 1989; Eglese, 1990). Therefore, an important feature of a more advanced local search method should be the capability to escape from such local minima. Unfortunately, the probability of searching in cycles increases with the capacity to escape so a control mechanism is needed to avoid cycles.

Local search methods may be combined with other heuristic approaches to construct an initial schedule. The main advantages of local search methods are that they can be applied in a general manner to various scheduling problems. They only require a specification of the solutions, a cost - or evaluation - function and a neighbourhood structure. Of course, depending on the application they may be successful in different ways and several parameters have to be adjusted to the specific problem (Dorn, 1995). Two disadvantages of local search methods are given by Aarts & Van Laarhoven (1989): the local minimum found depends on the initial configuration of the method and the worst-case time complexity (an upper bound on the computation time) of the search is generally unknown.

Unfortunately there is no classification, common framework or theory available that supports the choice of a good local search method for a given application (Glover & Greenberg, 1989; Dorn, 1995). Such a classification cannot be given since it is not clear what are the discriminating characteristics of an application. Furthermore, according to Dorn (1995), the performed comparisons in the current literature are not sufficient to support such a discrimination. According to Glover & Greenberg (1989) advances in the development and refinement of general methods depend in part on identifying the type of adoption to a specific problem domain that will prove most effective. A given general strategy can always be applied in a variety of ways to the solution of a particular type of problem, and its performance will as a matter of fact depend on which of these ways is chosen. Glover & Greenberg (1989) believe that such an identification can only come about by means of empirical study. This opinion is shared by Dorn (1995) who remarks that *experimental comparisons* seem to be the only possible way of evaluating different optimization methods and their settings. Aarts & Korst (1989) give two quantities, which can be used for the empirical analysis of the performance of an approximation algorithm: the quality of the final solution obtained by the algorithm, and the running time required by the algorithm. The quality of the final solution is defined as the relative difference between the final cost f^* and the optimal cost f_{opt} if that is known. The running time is quantified by the number of elementary transitions or by the required CPU time. Aarts & Van Laarhoven (1989) suggest that, besides effectiveness (quality) and efficiency (running time), other criteria such as simplicity, ease of implementation, applicability and flexibility should also be considered. In their opinion flexibility refers to the ability of an algorithm to handle problem variations and different problems. According to Fox & McMahon (1991), an objective assessment of special operators within local search methods must consider code size and complexity, actual execution time, number of generations, best solution, initial rate of convergence, as well as factors such as long-term rate of convergence.

In this chapter four different local search methods will be described, which will be applied in Chapter 7 to determine the internal transport sequence in a pot plant nursery. These local search methods are:

- Genetic Algorithm (GA);
- Simulated Annealing (SA);
- Tabu Search (TS);
- Random Search (RS).

The first three methods were chosen, because they gave promising results in the field of scheduling and these were mentioned in the literature. Several authors applied GA (Bagchi et al., 1991; Chen et al., 1995; Davis, 1985; Dorn, 1995; Filipic, 1993; Fox & McMahon, 1991), SA (Aarts et al., 1994a; Jeffcoat & Bulfin, 1993; Johnson & McGeoch, 1997) and TS (Barnes & Laguna, 1993; Glover & Greenberg, 1989; Heinrici, 1993; Punnen & Aneja, 1993 & 1995; Reeves, 1995; Rochat & Semet, 1994; Widmer, 1991) to a range of scheduling and sequencing problems such as the timetabling problem, the flow shop scheduling problem, the job shop scheduling problem and the travelling salesman problem. The last local search method RS is an example of a very simple descent method, that only accepts improvements and that is not capable of escaping local minima. This method was used to make a comparison with the other three local search methods that are capable of escaping local minima.

6.2 Genetic Algorithm (GA)

The Genetic Algorithm (GA) was developed by John Holland in the 1970s to mimic some features of natural evolution (Goldberg, 1989; Davis, 1991). It is a search method that is based on natural selection (the principle of 'the survival of the fittest') and on natural genetics (randomized information exchange). GA does not necessarily find the optimal solution, but it can find a satisfactory solution. Several authors have applied GA to scheduling problems (Davis, 1985; Dorn, 1995; Filipic, 1993; Fox & McMahon, 1991). The GA works with a population of solutions coded on strings. It applies three genetic operators to generate new improved strings: *reproduction*, *crossover* and *mutation*. These operators, that involve, string copying, partial string exchanging and random number generation are very simple. After the operators have been used, a new population of strings is formed: a new generation. The operators act as a randomized changing mechanism with which the solution space can be explored. The process is repeated for a number of generations until a certain stop criterion is reached. The size of the population stays the same during this iterative improvement process. A general description of the successive steps of the GA is given by Goldberg (1989) and by Davis (1991) and involves the following stages:

1. Generate an initial population with a fixed number of strings
2. Evaluate each string in the initial population
3. Apply a reproduction operator on strings in the population based on their fitness value
4. Create new strings by applying the genetic operators crossover and mutation
5. Determine the fitness value of each string in the current population
6. Replace some of the strings in the current population by newly formed strings based on their fitness values; this results in a new generation with the same fixed number of strings
7. Stop the algorithm when a chosen stop criterion applies; if not continue with Step 3.

In the GA solutions are represented on *strings* similar to chromosomes in the natural genetics, which contain certain variables or parameters (genes) of the problem. Dorn (1995) classifies the representation of schedules on strings in direct and indirect approaches. In a *direct representation,* the string contains all the required knowledge to evaluate a schedule. The disadvantage of the direct representation is that the GA must be very specialized and cannot be reused for another problem. Therefore he favours the *indirect representation* where a schedule builder constructs a legal schedule from a string. In that approach, the only part that has to be modified for a new application is the schedule builder which makes the indirect representation more general. Dorndorf & Pesch (1992) remark that the traditional *binary* string representation is often unsuitable for combinatorial optimization problems because it is very difficult to represent a solution in such a way that sub-strings have a meaningful interpretation. Therefore, solutions to a scheduling or sequencing problem are often represented as a string with a *permutation* of integer numbers, for example:

9	8	4	5	6	7	1	3	2	0	(string 1)

Chen et al. (1995) describe a string for their flow shop problem as a sequence of the jobs in the problem. This method could be applied to any scheduling problem. Davis (1985) uses an intermediary, encoded representation of schedules that can be changed and improved by crossover operations, while employing a decoder that always yields legal solutions to the problem. He uses a list of preferences for each work station, linked to times. He derived a complete schedule for his job shop problem from this list. The decoding routine for this representation was a simulation of the job shop's operations. The representation guaranteed that only legal schedules were produced. Filipic (1993) represented schedules as strings of length N, where N is the total number of jobs to be executed daily. In his approach the value at the i-th position denotes the set-up time of the i-th job which can only be selected from the interval between the earliest and the latest set-up time for that job. Fox & McMahon (1991) use a Boolean (0-1) matrix representation of a sequence for the travelling salesman problem, which includes the information about the sequence, including individual city-to-city connections and predecessors and successors relations. Matrix element x_{rc} contains a one, if city r occurs before city c in the given sequence. The column sums of the matrix give the number of predecessors of a city and the row sums reveal the number of successors.

The GA considers a fixed number of feasible solutions simultaneously known as a *population*. This approach is in contrast to other local search algorithms such as Simulated Annealing and Tabu Search, which are based on manipulating only one feasible solution, moving from point to point in the search space (Dorndorf & Pesch, 1992; Muller et al., 1993). However, these other local search algorithms may have parallel versions, where solutions are considered independently, without exchanging information. Solutions in the *initial population* of GA can be: a random sample of the search space, generated by a specially designed heuristic sequencing algorithm or the solutions of a previous run with the GA. In the algorithm of Filipic (1993), schedules are initialized by randomly assigning set-up times to jobs and simultaneously checking the constraints. Muller et al. (1993) also create the initial population randomly. A suitable heuristic algorithm will have to generate several alternative initial solutions to create diversity in the population. The initial solutions compose the first generation of solutions.

The GA will then form a new generation of the population of solutions through the execution of the genetic operators (crossover, mutation and reproduction).

The *fitness (or evaluation) function* links the GA to the problem which it has to solve. It is used to evaluate each solution of the population. A fitness value is usually scaled before the selection phase is entered to avoid premature convergence of the population, caused by highly fit individuals (Grefenstette, 1990; Filipic, 1993). Without some kind of fitness normalisation strings with a very high fitness value would rapidly replace all other strings in the population. That way, the variation in the population would be reduced completely and thus the capability of finding a string with an even higher fitness value. On the other hand, methods for fitness normalisation are also needed to avoid an extremely weak selective pressure (Hunter, 1995a), for example, when all fitness values are in a very small range, far from zero (for example 9 090-10 000). Fitness normalisation methods take the original fitness values and change them into values with more contrast. The *selection* of individual strings for genetic operations is performed according to their fitness value. Selected strings are often called *Parents* and the new strings that are generated through the genetic operators are called *Children*. A possible selection method is the so-called roulette wheel selection (Davis, 1991). The selection procedure is random, but strings with a higher fitness value have a higher probability of being selected. This way better strings will be used more often as parents for the genetic operations crossover and mutation.

Reproduction is used to copy solutions with a better value at the cost of solutions with a worse value. Individual strings are copied according to their objective fitness function values. The total size of the population has to stay the same, so *replacement* is used, which means that some strings will disappear in favour of some others. Strings with a higher fitness value have a higher probability of contributing one or more offspring in the next generation. This means that better strings have a higher probability of emerging in the next generation(s).

The *crossover* operator is used to exchange parts between two parent solutions of the population. This process may proceed in two steps. First, members of the newly produced strings are paired at random. Secondly, each pair of strings undergoes crossover. When *one-point crossover* is applied, two new children strings are created by exchanging all characters between the two parent strings starting with the character after a random crossing point until the character on the end of the string is reached. The first part of the sequence of Parent 1 is combined with the second part of the sequence of Parent 2 (and the other way around for the second part of Parent 1). This induces an information exchange between the two parent strings. The best parts of the two parent strings may be combined this way, which will sometimes result in a better child string with a higher fitness value. Grefenstette (1990) concluded that the straightforward application of genetic operators such as crossover and mutation can result in illegal representations when solutions are represented as permutations of discrete objects, which is the representation for most combinatorial optimization problems. The effect of one-point crossover is that a sub-sequence of discrete objects in the beginning of Parent 1 is coupled with a sub-sequence of discrete objects at the end of Parent 2. Non-feasible solutions will occur, when a string part with a certain discrete object is combined with another string part that contains the same discrete object. This will lead to an non-feasible child on which the discrete object occurs twice, and a second child on which it does not occur at all. For example:

	9	8	4	5	6	7	1	3	2	0	(Parent 1)
	8	7	1	2	3	0	9	5	4	6	(Parent 2)
						↑ one crossover point					
→											
	9	8	4	5	6	0	9	5	4	6	(non-feasible Child 1)
	8	7	1	2	3	7	1	3	2	0	(non-feasible Child 2)

Therefore, the representation of solutions as strings with a permutation of integers requires more sophisticated *special crossover operators* than the one-point crossover operator of the traditional binary string representation in order to get feasible offspring (Dorndorf & Pesch, 1992). Crossover methods for permutations need to preserve valid permutations (Hunter, 1995a). The applied genetic operators must be designed so that no orders or jobs exists twice on a new string to avoid illegal schedules (Dorn, 1995). Several special crossover methods have been developed for permutation problems, that use more than one crossover point: order crossover (OX), cycle crossover (CX) and partially mapped crossover (PMX) (Goldberg, 1989; Fox & McMahon, 1991; Hunter, 1995a; Dorn, 1995; Chen et al., 1995).

Order crossover (OX) creates children which preserve the order and position of symbols in a sub-sequence of one parent while preserving the relative order of the remaining symbols from the other parent (Fox & McMahon, 1991). OX uses two crossover points to define the borders of the sub-sequence in each parent. In the first step OX removes the symbols from Parent 1, that will be added from the sub-sequence in Parent 2 later on and replaces them with the symbol X. The same is done for Parent 2. Then OX shifts the symbols of the own sub-sequence of a parent to the left, pushing forward the retained symbols to fill in the gaps that were left by the removal of the exchanged symbols, while preserving the original order of the symbols. When the symbols are ‘pushed off’ the string at the left side, they cyclically enter the string at the right side again. Finally the actual crossover is performed by exchanging the sub-sequences between the two crossover points of the two parents. Goldberg (1989) gives the following example:

	9	8	4	**5**	**6**	**7**	1	3	2	0	(parent 1)
	8	7	1	**2**	**3**	**0**	9	5	4	6	(parent 2)
				↑	two		↑ crossover points				
→											
	9	8	4	5	6	7	1	X	X	X	
	8	X	1	2	3	0	9	X	4	X	
→											
	5	6	7	X	X	X	1	9	8	4	
	2	3	0	X	X	X	9	4	8	1	
→											
	5	6	7	**2**	**3**	**0**	1	9	8	4	(child 1)
	2	3	0	**5**	**6**	**7**	9	4	8	1	(child 2)

Cycle crossover (CX) creates children so that the position of each symbol in a child is determined by one of its parents (Fox & McMahon, 1991). Hunter (1995a) gives a description of the method. Starting at a random position in the first parent, the symbol is copied into the first child. Since that position is filled from the first parent, the corresponding symbol of the second parent cannot be used in the first child. Therefore, that symbol must be copied from the first parent as well. This pattern is followed until a closed cycle of dependencies identifies the part of the first child to be copied from the first parent. The balance is then copied from the second parent. The opposite copy is simultaneously made for the other child. Goldberg (1989) gives the following example:

	9	8	2	1	7	4	5	0	6	3	(Parent 1)
	1	2	3	4	5	6	7	8	9	0	(Parent 2)

Starting with 9 in the first string, the next closed cycle (9 → 1 → 4 → 6 → 9) can be identified:

9	-	-	1	-	4	-	-	6	-
↓			↓		↓			↓	
1	-	-	4	-	6	-	-	9	-

Copying the balance means exchanging the remaining numbers:

	-	2	3	-	5	-	7	8	-	0	
	-	8	2	-	7	-	5	0	-	3	
→											
	9	2	3	1	5	4	7	8	6	0	(Child 1)
	1	8	2	4	7	6	5	0	9	3	(Child 2)

Partially mapped crossover (PMX) creates children that preserve the order and position of symbols in a sub-sequence of one parent while preserving the order and position of many of the remaining symbols from the other parent (Fox & McMahon, 1991). Two crossover points are selected, and the two strings take each other's mid-sections (Hunter, 1995a). However, the newly inserted values are swapped with the displaced values to maintain a valid permutation. This way the crossed-over strings gain a section from each other at the cost of scrambling their retained values. Goldberg (1989) gives the following example, where both strings perform three swaps (2 with 5, 6 with 3 and 7 with 0):

	9	8	4	|	5	6	7	|	1	3	2	0	(Parent 1)
	8	7	1	|	2	3	0	|	9	5	4	6	(Parent 2)
				↑		two		↑	crossover points				
→													
	9	8	4	|	**2**	6	7	|	1	3	**5**	0	
	8	7	1	|	**5**	3	0	|	9	**2**	4	6	
→													
	9	8	4	|	2	**3**	7	|	1	**6**	5	0	
	8	7	1	|	5	**6**	0	|	9	2	4	**3**	
→													
	9	8	4	|	2	3	**0**	|	1	6	5	**7**	(Child 1)
	8	**0**	1	|	5	6	**7**	|	9	2	4	3	(Child 2)

Mutation is used to change a certain part of a string randomly. This is needed because some potentially useful parts of the string can get lost in the process of applying reproduction and crossover operators. Mutation on a binary representation changes the value of a single bit of the string from 1 to 0 or vice versa which can completely change a solution. The most primitive mutation operator for the permutation representation is a *swap mutation* of two adjacent integer numbers (Dorn, 1995). Mutation can also exchange groups of integer numbers in one permutation. *Position-based* swap mutation selects two positions randomly and exchanges the integer numbers at these positions. For example:

9	**8**	4	2	3	0	**1**	6	5	7	(Parent 1)
→										
9	**1**	4	2	3	0	**8**	6	5	7	(Child 1)

Order-based mutation selects two positions and puts the integer number from the second position in front of the integer number at the first position. For example:

9	**8**	4	2	3	0	**1**	6	5	7	(Parent 1)
→										
9	**1**	**8**	4	2	3	0	6	5	7	(Child 1)

The *inversion mutation* operator completely inverts the sequence of the genes between two points on the chromosome (Hunter, 1995a). For example:

9	8	4	2	3	**0**	**1**	**6**	**5**	7	(Parent 1)
→										
9	8	4	2	3	**5**	**6**	**1**	**0**	7	(Child 1)

The operators crossover and mutation are applied at a certain rate to the strings in the population, depending on the specific problem at hand. A very low mutation rate is usually chosen, to avoid too much disturbance of the solutions in the population. In most applications the crossover and mutation rate are fixed and have to be specified before a run of the algorithm is started. Another possibility is to vary these rates during the search process, according to the characteristics of the population in each generation (Grefenstette, 1990). The mutation rate could be enlarged for example, when no more improvement is found during a certain number of generations. This way the diversity of the population can be increased again.

Important control parameters of the GA are population size and the stopping criterion. The *population size* has a dramatic impact on the performance of a GA. If the population is too small, it will contain insufficient information to reach all areas of the search space. On the other hand, more computation time is needed per generation when a population is large (Grefenstette, 1990; Dorn, 1995). Chen et al. (1995) found that the efficiency of the GA can be largely increased by selecting a good initial population and a reasonable population size. They found an upper limit to the size of the population (60 in their experiments) above which better results could not be guaranteed. Several *stopping criteria* are possible: the GA can be stopped after a predetermined fixed number of generations (iterations), or when a pre-set fitness value is obtained or finally when no further improvement is found during a large number of generations (Chen et al., 1995).

When no special action is taken, the best member of the population may disappear again in a next generation through the genetic operators crossover and mutation. An *elitist strategy* prevents this loss by always preserving the best string of a population from one generation to the next. Elitism can sometimes lead to the domination of a population by a super individual, but on balance it appears to improve genetic algorithm performance (Davis, 1991; Filipic, 1993; Dorn, 1995).

6.3 Simulated Annealing (SA)

Simulated annealing (SA) is based on an analogy between the physical annealing process of solids and the problem of solving large combinatorial optimization problems (Aarts & Korst, 1989; Aarts & Van Laarhoven, 1989; Eglese, 1990; Aarts et al., 1997). Physical annealing refers to the process of finding low energy states of a solid by initially increasing the temperature to melt the solid. Then the temperature is carefully lowered, spending a long time at temperatures close to freezing point until the particles arrange themselves in the ground state of the solid. The different states of the substance correspond to feasible solutions in a combinatorial optimization problem and the energy of a state corresponds to the cost of a solution which has to be minimized. Many authors have applied SA to scheduling problems (Aarts et al., 1994a; Jeffcoat & Bulfin, 1993; Johnson & McGeoch, 1997).

SA belongs to a class of local search algorithms called threshold algorithms (Aarts et al., 1997). The main characteristic of the SA algorithm is that it sometimes allows a neighbourhood move which increases the value of the objective function as an escape mechanism from a local optimum. The acceptance or rejection of a non-improving neighbourhood move is determined by an *acceptance probability*. This way SA uses a stochastic way of steering that produces good, though not necessarily optimal, solutions within a reasonable computing time.

A general description of the successive steps of SA as an approximation algorithm can be found in Aarts & Korst (1989), Heinrici (1993) and Aarts et al. (1997). The following elements are identified for solving a minimization problem:

1. Start with any initial solution i with cost f(i) and choose an initial control parameter c_k (k=0)
2. Choose a potential neighbourhood move for a transition to a new solution randomly and construct the corresponding new solution j
3. Calculate the cost f(j) of solution j, and the cost difference f(i) - f(j);
4. Determine whether solution j is accepted or not from solution i by applying the acceptance probability:

   $$P_k(\text{accept } j) = \begin{cases} 1 & \text{if } f(j) \leq f(i) \\ \exp((f(i) - f(j))/c_k) & \text{if } f(j) > f(i) \end{cases}$$

5. k=k+1; lower c_k;
6. Stop the algorithm when stop criterion applies; otherwise continue with step 2.

The variable c_k is the *control parameter* equivalent to the role of temperature in the physical annealing process. The probability of accepting a less favourable non-improving transition move is implemented by comparing the value of exp((f(i)-f(j))/c_k) with a random number generated from a uniform distribution on the interval [0,1) according to Aarts & Korst (1989). The acceptance probability implies that small increases in the objective function are more likely to be accepted than large increases. However, there is no limitation to the size of a deterioration, although a large increase in the objective function will only be accepted with a small probability. The algorithm begins with a relatively high value of c_0, to avoid being prematurely trapped in a local optimum. Then it proceeds by attempting a certain number of neighbourhood moves at each value of c_k, while c_k is gradually dropped (Eglese, 1990). When c_k is large, almost all neighbourhood moves will be accepted. As c_k decreases, less transitions will be accepted and finally, when c_k approaches zero most non-improving moves will be rejected (Aarts & Korst, 1989; Eglese, 1990).

The application of SA requires a concise problem representation, a neighbourhood function, a transition mechanism and a cooling schedule (Aarts & Korst, 1989; Aarts et al., 1997). No general guidelines exist for these components, except for the cooling schedule. Therefore experience, taste and skill play an important role when SA is applied to a specific problem. Aarts et al. (1997) do not expect that this will change in the near future.

A solution to a scheduling or sequencing problem is often coded as a permutation of integers on a string just like solutions used for the GA. An *initial solution* for SA can be produced by a heuristic (Heinrici, 1993), but it can also be generated randomly, or the solution of a previous run can also be taken.

The problem must be clearly formulated for SA so that a *neighbourhood* can be defined for each solution (Eglese, 1990). There are many different possibilities for transitions between solutions. The most obvious transitions for sequencing problems lie in shifting/inserting an element or in swapping two elements (Reeves, 1993). According to Heinrici (1993) transitions should produce solutions with similar objective function values. Experiments showed that swapping is the best transition for SA because it produces neighbours that differ little from the preceding solution in the values of the objective function.

Just like the other local search methods, SA needs a *cost function* to steer the local search process. A cost function should be as simple as possible, because it is the most time consuming part of the algorithm. The calculation of cost differences should preferably be done incrementally (only for the local rearrangement) without having to calculate the whole cost function.

In a finite-time implementation of the SA algorithm, a finite sequence of values of the control parameter should be specified (Aarts & Korst, 1989; Aarts et al., 1997):

- an initial value of the control parameter c_0;
- a decrement function for decreasing the value of the control parameter c_k;
- a final value of the control parameter specified by a stop criterion;
- a finite number of transitions at each value of the control parameter c_k.

A choice for the values of these parameters is referred to as a *cooling schedule* or an *annealing schedule*. Aarts & Korst (1989) found that the quality of the final solution mainly depends on the speed at which the control parameter c_k is lowered. According to Eglese (1990) the choice of the annealing schedule does influence the performance of the algorithm. Aarts & Korst (1989) also suggest that a cooling schedule should be chosen carefully. For this reason the search for adequate cooling schedules has been the subject of many studies. Aarts et al. (1997) distinguish between a *static* cooling schedule, where the parameters are fixed and cannot be changed during the algorithm and a *dynamic* cooling schedule, where the parameters are adaptively changed during the execution of the algorithm. For the static cooling schedule Aarts et al. (1997) suggest a simple schedule known as the geometric schedule. The initial value of the control parameter may be chosen as $c_0 = \Delta f_{\max}$, where $\Delta f_{\max}$ is the maximal difference in cost between two neighbouring solutions. This value could be estimated by evaluating a sample of the neighbouring solutions. The control parameter value is often lowered with a decrement function:

$$c_{k+1} = \alpha \cdot c_k$$

where α is a positive constant smaller than but close to 1 (mostly between 0.80 and 0.99). The final value of the control parameter is fixed at some small value (related to the smallest possible difference in cost).

Aarts & Van Laarhoven (1989) described SA mathematically by means of a Markov chain: a sequence of trials, where the outcome of each trial only depends on the outcome of the previous trial. They distinguish between two algorithms according to the method of decreasing the control parameter during the course of the algorithm. In the *inhomogeneous algorithm* the control parameter is decreased after each transition. This algorithm can be described by a single inhomogeneous Markov chain. In the *homogeneous algorithm* the control parameter decreases after a number of transitions. This can be described by a sequence of homogeneous Markov chains, each generated at a fixed value of the control parameter. By using the theory of homogeneous and inhomogeneous Markov chains, the algorithm has been proven to converge to globally optimal solutions with probability 1 (Aarts et al., 1997). However, to guarantee this convergence to a global minimum, the cooling schedule must be exponentially slow. Thus SA may require more than polynomial time to reach a global minimum. In practical implementations, SA is only an *approximation algorithm*.

Several modifications to the basic SA have been proposed, which are summarized by Eglese (1990). Simple modifications are: storing the best solution so far; sampling the neighbourhood without replacement; and alternative acceptance probabilities. Also more complex modifications are possible: combining SA with another method; problem specific modifications; and parallel versions. A combination of different local search algorithms is known as a *multilevel approach*. According to Aarts et al. (1997) a combination will lead to new variants of local search algorithms, but not to new algorithmic concepts. *Parallel simulated annealing* algorithms distribute the execution of various parts over a number of parallel processors (Aarts et al., 1997). This is a promising way of speeding up the execution of SA, but it is also a very difficult task, due to the sequential nature of SA. *Threshold accepting* is a deterministic version of SA, where a neighbour solution is accepted if the difference in cost between the neighbour and the current solution is smaller than a non-negative threshold. This threshold may vary in the course of the algorithm's execution (Aarts et al., 1994b; Dorn, 1995).

6.4 Tabu Search (TS)

Tabu search (TS) is a deterministic local search algorithm that has the ability to incorporate and guide another search procedure and therefore it may be viewed as a meta-heuristic for combinatorial problem solving. TS can be superimposed on other heuristics to prevent them from becoming trapped at locally optimal solutions. These heuristics may be high-level procedures, like other local search methods, or nothing more than a description of the available moves for transforming one solution into another. TS can be used to guide any process that employs a set of moves for transforming one solution into another and that provides an evaluation function for measuring the attractiveness of these transition moves (Glover, 1989, 1990; Glover et al., 1993; Aarts et al., 1994b; Glover & Laguna, 1997). TS is founded on three primary themes:

- the use of flexible attribute-based memory structures designed to permit evaluation criteria and historical search information to be exploited more thoroughly than, for example, a memory-less system as SA;
- an associated mechanism of control - for employing the memory structures - based on the interplay between conditions that constrain and free the search process (embodied in tabu restrictions and aspiration criteria);
- the incorporation of memory functions of different time spans, from short-term to long-term, to implement strategies for intensifying and diversifying the search.

TS has been shown to be a remarkably effective approach in the area of production scheduling (Barnes & Laguna, 1993; Glover & Greenberg, 1989; Heinrici, 1993; Hertz, 1991; Punnen & Aneja, 1993 & 1995; Reeves, 1995; Rochat & Semet, 1994; Widmer, 1991). TS can obtain high quality solutions with modest computational effort (Glover, 1989) and TS allows a high degree of freedom for designing solution procedures. A general description of the essential characteristics of a TS procedure for solving a minimization problem is given by Hertz et al. (1997):

1. Choose an initial solution i. Set the best solution found so far i*=i and iteration k=0
2. Set iteration k=k+1 and generate a subset of solutions in the neighbourhood of solution i at iteration k, such that these neighbour solutions are either not tabu or at least one of the aspiration conditions holds
3. Choose the best new solution j in the subset with respect to the function value f(j), and set i=j
4. If f(i)<f(i*), then set i*=i
5. Update the tabu and aspiration conditions
6. Stop the algorithm when a chosen stop criterion applies; otherwise continue with Step 2.

An *initial solution* has to be generated, for example, with a heuristic scheduling algorithm. Obtaining one or more good starting solutions instead of a random start can be beneficial and speed up TS considerably (Barnes & Laguna,1993; Reeves, 1993). A neighbourhood of a solution is defined in terms of moves for transforming the solution into new solutions. TS may be viewed as a *dynamic neighbourhood search technique*: each iteration redefines the neighbourhood from which the next solution will be drawn, based on conditions that classify certain moves as tabu (Glover et al., 1993; Hertz et al., 1997). Suitable *change operators*, that generate alternative neighbour solutions have to be chosen. There are many different possibilities for transitions between solutions. Good change operators for scheduling problems are 'insert moves' and 'swap moves'. Insertion procedures are often preferred because a single swap may be achieved by two insert moves. Also, in the context of sequencing and partitioning problems, insert moves provide a higher degree of perturbation of the current solution than swap moves (Barnes & Laguna, 1993; Heinrici, 1993).

Tabu restrictions are designed to prevent reversal or repetition of certain moves. The primary goal of the tabu restrictions is to permit the method to go beyond points of local optimality while still making high quality moves at each step (Glover, 1990). A short term memory structure in TS for managing the tabu restrictions is the so-called *Tabu List*. In the Tabu List, one or more attributes of an applied transition move are stored after each iteration. A transition move is considered forbidden - tabu - if it is on the tabu list. The tabu list has a finite length (L) and it is handled with a FIFO strategy: at each iteration, the algorithm memorizes a new attribute and it forgets the oldest one. The basic idea is that a tabu status of a move can avoid cycles in the evolution of the search by preventing the guided algorithm from repeating the most recently made moves (Dell' Amico & Trubian, 1993). A number of tabu restrictions are possible in the context of scheduling (Barnes & Laguna, 1993). For example, after a swap move has been performed:

- both jobs may be forbidden from moving;
- the job that moves to an earlier (or later) position in the schedule may be locked into its new position;
- both jobs may be forbidden from exchanging their positions, but they may participate in swaps with other jobs;
- the jobs (or just one job) may not be allowed to return to the positions they occupied prior to the exchange.

Attention has to be given to the best *length of the tabu list*. Experimentation has indicated that the appropriate length of the tabu list is a very stable and robust parameter, that depends on the problem class (Glover & Greenberg, 1989). In applications that were studied so far a tabu list length of somewhere between 5 and 20 was chosen (Glover, 1990; Heinrici, 1993; Hertz, 1991; Punnen & Aneja, 1993 & 1995). According to Glover & Greenberg (1989) several applications of TS found the 'magic number' 7 (±2) to be a remarkably good choice for tabu list size. When the size is too small, cycling can still occur and when it is too large the solution quality will decrease because too many moves are forbidden (Glover et al., 1993). Several authors made the tabu list proportionally larger as the number of jobs increased (Barnes & Laguna, 1993; Dell' Amico & Trubian, 1993; Dorn, 1995). It is also possible to change the length of the tabu list dynamically (Glover et al., 1993; Hertz et al., 1997; Rochat & Semet, 1994).

A usual strategy in TS is the definition of an *aspiration criterion*, a rule that cancels the effect of the tabu status of a move in particular situations if the aspiration criterion is satisfied (Dell'Amico & Trubian, 1993; Glover, 1989a). The evaluation of a move can be based on the change produced in the objective function value. If the move is not tabu, it is immediately accepted as admissible. Otherwise the aspiration criteria are given an opportunity to override the tabu status, providing the move with a second chance of becoming admissible (Glover, 1990). The role of an aspiration criterion is to provide added flexibility in choosing good moves. A simple type of aspiration criterion accepts a tabu move if it produces a better solution than the best known solution so far. The tabu restrictions and the aspiration level criteria of TS play a complementary role in constraining and guiding the search process and they integrated into one common framework (Glover, 1989).

Intermediate-term and long-term memory functions are employed within TS to achieve regional intensification and global diversification of the search. Combined with the short-term memory functions, the intermediate and long-term functions provide an interplay between 'learning' and 'unlearning' (Glover, 1989). The use of *intermediate-term memory function*s involves creating a network of good solutions as a matrix for generating other solutions with good properties. The algorithm tries to observe whether the good solutions visited so far have some common properties such as the presence or absence of certain elements. The collection of good solutions is then used to define a sub-region of the search space for intensified search, and for launching explorations into neighbouring regions. This is done by restricting or penalizing moves during a subsequent period of regional search intensification (Glover, 1989; Glover et al., 1993; Glover & Greenberg, 1989). The *long-term memory function*, whose goal is to diversify the search, employs principles that are roughly the reverse of those for the intermediate-term memory. The long-term memory function guides the process to regions that markedly contrast with those examined thus far (Glover, 1989). Restarting the solution process from different solutions generated by some kind of heuristic algorithm is a common way of diversifying (Barnes & Laguna, 1993; Dell'Amico and Trubian, 1993).

A critical step is *choosing the best admissible candidate*. Given a current solution, the next solution is selected from a set of admissible neighbours. In its simplest version, this set contains all neighbours that are not on the tabu list or that satisfy some aspiration level (Aarts et al., 1994b). In this simple version, all admissible neighbours have to be evaluated to find the best transition from the current solution. In neighbourhoods with many neighbours it is often too expensive in terms of CPU-time to examine the complete neighbourhood. Therefore, the candidate list of admissible neighbours will have to be limited to a promising region - a small subset - of the neighbourhood, so that only this subset has to be examined. If there is no admissible solution in the subset it can be temporarily extended (Heinrici, 1993; Dorn, 1995). Each of the moves of the candidate list is evaluated in turn. Reeves (1993) also states that the more traditional, 'brute-force' TS approach in which complete neighbourhoods are explored is not an efficient use of computing resources. Such an approach may be adequate for small problems, but he believes that overall it is better to make fairly limited explorations in the search space and exploit the best of these, rather than pursue a lengthy search before each move is made. Some authors even use an approach that immediately accepts a non-tabu improving move when it is discovered (Widmer & Hertz, 1989; Glover, 1997), only searching the whole neighbourhood if no such move can be found. If this is the case, the best non-improving non-tabu move is used. Reeves (1993) found that having a small neighbourhood (and hence a large number of iterations) appeared to be a much better use of resources.

Glover et al. (1993) and Hertz et al. (1997) mention four possible *stopping conditions,* that may often be combined. TS can be stopped when evidence can be provided to show that an optimal solution has been found; when the neighbourhood of the current solution is empty; when a maximum number of iterations has been reached; or when the number of iterations performed since the best solution last changed is greater than a specified maximum number of iterations.

Barnes & Laguna (1993) made the following observations about the primary components of an effective production scheduling TS method:

- a procedure for obtaining ‘good’ starting solutions, as opposed to a random start, is often beneficial;
- the efficient selection and evaluation of moves is a highly important consideration;
- the most successful TS methods often use more than one class of moves, either alternating their use or simultaneously considering them at each iteration;
- attention to the best length of the tabu list is essential;
- diversification strategies are essential to successful TS approaches to large production scheduling problems;
- improvements have sometimes been realized by forming hybrid methods where TS is combined with other approaches such as SA.

6.5 Random Search (RS)

The local search method Random Search is a simple descent algorithm. A general description of the successive steps of RS when solving a minimization problem is as follows:

1. Start with any initial solution i with cost f(i)
2. Choose a potential neighbourhood move for a transition to a new solution randomly and construct the corresponding new solution j
3. Calculate the cost f(j) of solution j;
4. Determine whether solution j is accepted or not from solution i by applying the deterministic acceptance criterion: accept j if $f(j) \leq f(i)$
5. Stop the algorithm when a chosen stop criterion applies; otherwise continue with Step 2.

The solution to a scheduling problem is coded on a string as a permutation of integer numbers. A fitness function calculates the evaluation value of the solution. Simple neighbourhood moves can be applied to generate new solutions, such as a swap mutation that exchanges two integer numbers. The algorithm searches randomly through the possible neighbourhood moves, until it finds for the first time a neighbourhood move that leads to a new solution with a better fitness value than the initial solution. The new solution replaces the old one and then the random search continues until finally no further improvement can be found. The main disadvantage of RS is that sooner or later it will get trapped in a local minimum which has no neighbours with a lower fitness value. An advantage is the simplicity of the algorithm and the ease of implementation.

6.6 Performance of local search methods

Glover & Greenberg (1989) emphasize that general local search methods, which embody broadly applicable solution principles, cannot be expected to compete with specialized solution approaches evolved over a significant span of time for high performance on a given problem class.

The results of the *Genetic Algorithm* were worse than TS in a comparison carried out by Dorn (1995). The problem that he identified in the GA was the numerous design decisions that influence performance and the amount of work needed to tailor them to a specific application. Chen et al. (1995) conclude that further research is needed to develop a systematic approach to determine the values of the major ingredients of genetic algorithms for scheduling problems. A GA-based heuristic in their application yielded better results than two problem specific heuristics. They proved that their GA based heuristic was an effective and very efficient solving method for flow shop problems. The CPU time required to solve their problem was short enough to make it suitable for solving real world problems. Dorndorf & Pesch (1992) found that compared to standard heuristics, in the case of the travelling salesman problem for instance, genetic algorithms are not well suited for adjusting structures that are very close to optimal solutions. Therefore, they consider it essential to incorporate improvement operators to make the GA more competitive. Filipic (1993) discovered that a GA that included the local improvement of individuals, outperformed a blind GA on a hypothetical problem of scheduling independent tasks. He tested four versions: a pure GA with no extensions, an elitist GA, a GA with local improvement, and a GA that included both elitism and local improvement. He achieved the best performance with the version including both extensions. Disadvantages of the GA are mentioned by Muller et al. (1993): there is no guarantee it will find an optimal solution, the reasoning of the GA cannot be traced and the algorithm cannot provide explanations. Fox & McMahon (1991) believe that the primary factor that prohibits the use of GAs in real-time scheduling problems still is execution time. Improved hardware and software will decrease the amount of time needed to evaluate a solution but, in their opinion, the real time savings will have to come from minimizing the number of solutions to be evaluated. To achieve this a GA will need to find good solutions within a minimum number of generations and with a minimum population size.

Dorn (1995) mentions that experimental comparisons of *Simulated Annealing* have shown that SA can compete with a problem specific algorithm, for example, the 'shifting bottleneck procedure'. Many papers proposing SA procedures for scheduling problems report good results (Jeffcoat & Bulfin, 1993) and SA appears to be a promising robust technique for solving scheduling problems. According to Aarts et al. (1994b) SA requires unusual computation time when compared to other approximation algorithms, but it yields consistently good solutions. As far as the TSP problem is concerned, Johnson & McGeoch (1997) conclude that the number of steps at each value of the control parameter needs to be at least proportional to the neighbourhood size to obtain tours with a high quality. However, this leads to a high computation time and therefore methods are needed that substantially reduce overall running time. Important advantages of SA are mentioned by numerous authors (Aarts & Korst, 1989; Eglese, 1990; Aarts et al., 1997). These advantages include:

- easy to implement; once a neighbourhood structure has been devised, the SA algorithm only occupies a few lines of code;
- general applicable to a wide range of optimization problems;
- robust; the final solution does not strongly depend on the choice of the initial solution;
- the stochastic acceptance criterion improves the performance, compared to a deterministic acceptance criterion (strict improvement);
- little insight is needed in the combinatorial problem structure;
- ability to obtain high-quality solutions (close to a global optimum) for many problems.

Of course SA also has some disadvantages (Aarts & Van Laarhoven, 1989; Eglese, 1990; Dorn, 1995) such as:

- high computation time required for finding high-quality solutions, particularly when compared to a problem specific algorithm; the average-case running time is close to the worst-case running time;
- if a sophisticated problem specific algorithm is available, it is usually competitive with and often superior to SA.

Tabu Search algorithms can find excellent solutions in reasonable running times (Aarts et al., 1994b; Dell'Amico & Trubian, 1993; Glover & Greenberg, 1989). Dell'Amico & Trubian's (1993) algorithm found high quality results within relatively small running times - no run required more than 6 minutes - without using specific information about the problem structure. In contrast to the other local search methods, TS has provided solutions for travelling salesman problems which in several instances are superior to the best known reported in the literature (Glover & Greenberg, 1989). The application of TS usually requires a non-trivial amount of testing and tuning (Aarts et al., 1994b). One of the advantages of TS mentioned by Hertz (1991), is that it can take into account all sorts of restrictions by defining the neighbourhood of a solution as a set of solutions that can be obtained from the original solution by applying moves that satisfy the constraints. Heinrici (1993) found that TS needed more special knowledge about the problem and more code in the implementation than SA. Several authors found that

TS was systematically superior to SA (Barnes & Laguna, 1993; Glover, 1989; Heinrici, 1993; Hertz & de Werra, 1987; Reeves, 1993) which means that, on average, better solutions can be produced even when the computing effort is restricted to a value which has been found to be effective for SA. Heinrici (1993) believed that a reason for the better performance of TS when compared to SA was that its course of search spent more time in the 'interesting' part of the solution space. According to Dell' Amico & Trubian (1993) TS is quite robust i.e. the differences between the solution values of the runs are small. According to Glover (1989) TS has the tendency to produce a variety of solutions that fall into an attractive range. This characteristic can be useful in situations where a mathematical model is used to generate candidate solutions that must then be evaluated further on the basis of external criteria.

The advantages and disadvantages of the three local search methods are summarized in Table 6-1.

Table 6-1. Summary of the most important advantages and disadvantages of the three local search methods as mentioned in the literature.

Advantages	Disadvantages
All methods	
- robustness - general applicability to a wide range of optimization problems	- the reasoning of the algorithm cannot be traced and it cannot provide explanations (black box) - specialized problem-specific algorithm often superior
Genetic Algorithm	
- required computation time is short enough to solve real world problems but still has to be improved further - competitive solutions if knowledge about the problem is incorporated	- numerous design decisions and much work to tailor many different parameters to a specific application - not well-suited for adjusting structures which are very close to optimal solutions - not guaranteed to find the optimal solution
Simulated Annealing	
- easy to implement - stochastic acceptance criterion improves performance - little insight is needed in the combinatorial problem structure - high quality solution close to a global optimum for many problems	- high computation time required
Tabu Search	
- reasonable/small computation time - excellent solutions - can be integrated with methods containing optimality	- more code in implementation than SA and non-trivial amount of testing and adjustment - needs special knowledge of the problem - only a meta-heuristic, so another heuristic is needed

6.7 Implementation of local search methods in SUGAL

Four local search methods GA, SA, TS and RS were implemented using a research software tool, called SUGAL (SUnderland Genetic ALgorithm package) developed by Hunter (1995a,b) at the University of Sunderland in the UK. This tool was written in C and it was compiled with PC Borland C++ 5.0. SUGAL was designed for research and experimentation with GA and related techniques. Therefore, Hunter has emphasized the provision of a large number of options, configurability and extendibility.

The main design problems that had to be solved in order to apply these local search methods to the internal transport sequencing problem were as follows:

- determining a way to represent the problem on a string;
- deciding how to obtain initial solutions;
- chosing a suitable selection method;
- developing a suitable fitness (evaluation) function, a fast and efficient way to calculate its results and a suitable fitness normalisation method;
- chosing a suitable crossover type and rate (only for GA);
- chosing a suitable mutation type and rate;
- chosing a suitable replacement mechanism;
- chosing the best cooling schedule with an initial temperature and a decay factor (only for SA);
- chosing a suitable size for the tabu list (for TS in combination with SA or RS);
- chosing the general parameters of the algorithm (the population size and the stopping criterion).

Considering the different representation methods mentioned in literature, it was decided that the most suitable representation would be to code the solution to the internal transport planning problem as a *permutation* of integer crop batch numbers (of output and input benches) on a string. The number of transport movements that are needed to perform a transport sequence can be calculated in combination with a specific parking method, which determines the parking rows for obstructing benches during the internal transport process (Section 5.2). The permutation data type was chosen for all four local search methods. *Initial solutions* in the form of different permutations are generated randomly by the initialisation procedure of SUGAL, which uses a pseudo-random number generator to control its stochastic behaviour (Hunter, 1995a). This pseudo-random number generator has to be started with a seed value and it then produces a sequence of numbers which are not correlated and appear to be random. All experiments were performed with the *'automatic seeding'* option. This means that SUGAL generates its own seed internally, using the current system clock setting. Although the system clock is not really random, autoseeding is a reasonable approximation.

The type of *selection method* depends on the chosen local search method. Roulette selection was applied for GA, and uniform selection for the other local search methods. In *roulette selection* each string has a probability of being selected that is proportional to its normalised fitness value. In *uniform selection*, each parent has an equal probability of selection irrespective of its fitness value.

The implementation of SUGAL requires programming a problem specific *fitness function* (= evaluation function). Therefore, two parking methods, described in Chapter 5, were programmed as alternative fitness functions: ‘first available parking row’ (Parking Method 1, Section 5.4.3) and ‘first available parking row without output benches’ (Parking Method 2, Section 5.4.4). The fitness function calculated the total number of transport movements for the generated transport sequence in combination with the chosen parking method (Appendix 6.2). The same fitness function was chosen for all four local search methods. In SUGAL the raw fitness value is modified by a chosen *fitness normalization* method in combination with a certain bias value (Hunter, 1995a). The modified fitness values are scaled so that their sum over the whole population is 1.0. Thus, the normalised fitness of a string can be interpreted as the strings share of the population fitness. In all experiments the *reverse scale fitness normalisation* was used. This method applies a linear function to the raw fitness values, in such a way that the ratio between the best solution and the worst solution in the population is equal to the bias. The value of the bias was set to 2 in the experiments, so the normalised fitness value of the best solution is twice the normalised fitness value of the worst solution (see the example in Figure 6-4). Adjusting the bias adjusts the amount of selective pressure per generation.

An *evaluation counter* was added at the beginning of the source code of the fitness function, to count the number of times that the fitness function is called (and executed). The evaluation counter is needed to be able to compare the efficiency of runs of experiments with a different population size or a different number of generations. The results of two local search methods can only be compared if they are obtained using the same number of evaluations.

The crossover operator was applied within the GA (Section 6.2). Three special *crossover methods* for the permutation data type, Cycle Crossover (CX), Partially Mapped Crossover (PMX) and Order Crossover (OX), were included in an extension package of SUGAL. A crossover rate of 1.0 per chromosome indicates that the probability that crossover will be used to generate candidate chromosomes is 100%. In this case crossover will be used to generate all candidates.

Two *mutation types* were available as a standard in SUGAL: swap and inversion mutation. These mutation types can be applied to each of the four local search methods, if required. Experiments were performed to find the best mutation type for the internal transport planning problem. SUGAL knows two main classes within the mutation type: per-gene and per-chromosome. In the experiments only per-chromosome mutation was used. This is able to change the entire string, rather than only one gene (crop batch number). However, only a very small *mutation rate* of 1.0 per-chromosome was chosen in the basic setting of the experiments. This means that the average number of mutations is one gene per chromosome, although some chromosomes may receive more mutations, and some none (Hunter, 1995a). The mutation rate was changed in the experiments with the small test cases (Section 6.8) to find the best value. Alternative mutation rates were 0.1; 0.5; 1.0 and 5.0.

Tabu Search was not included in SUGAL, so additional routines had to be added to SUGAL in order to implement TS. This was done in a straightforward way by adding a so-called *Tabu Search swap mutation* type. A *tabu list* was added to each individual string for the swap mutation type. The tabu list has a variable *tabu length*, which has to be specified at the beginning of each run. The value that is stored in the tabu list is the position on the string of the second crop batch number of the pair that is being swapped. By making this position tabu, it is guaranteed that the crop batch number will remain at that position on the string for at least as many iterations as the length of the tabu list. The tabu list is circular in form. First, the empty positions in the tabu list are filled with position numbers. When the list is full, the position number that has been in the list for the longest time will be replaced with the position number of the most recently performed swap move. The effect is that the swap moves in the tabu list are forbidden for a number of generations. When a position number is encountered that is on the tabu list, a new position number has to be generated for the second crop batch of the proposed swap. The Tabu Search swap operator can be combined with SA and RS. No form of aspiration criterion was built in, neither was any form of medium- or long-term memory. In contrast with traditional TS applications the neighbourhood is not completely (or partially) searched to generate a candidate list, but only one swap is compared with the tabu list causing only one neighbouring solution. This approach of accepting a non-tabu improving move as soon as it is discovered - first improving strategy - was also mentioned by several authors (Widmer & Hertz, 1989; Glover, 1997). For their problems they succeeded in obtaining good results with this approach. With this approach data structures and selection procedures within the current version of SUGAL could remain unchanged.

When SUGAL has generated candidate solutions, it tries to insert them into the population, using a *replacement method.* The type of replacement is determined by the type of local search method. The GA uses a uniform replacement method, with unconditional replacement. In uniform replacement, the members of the old population, which have to be replaced, are randomly selected and unconditional replacement means that the child automatically replaces the old string. SA and RS use a parental replacement method, which means that a child is compared with its own parent and that the parent is replaced by the child given some conditions. The difference between SA and RS is the condition. The method of conditional replacement for RS is 'if improved': a child replaces a parent string only if it has a better fitness value than the parent. The replacement condition used for SA is 'annealed'. In annealed replacement, the child always replaces the parent if it has a better fitness value, and according to an acceptance probability when the child has a worse fitness value. In SUGAL the initial value of the control parameter c_0 is called the annealing temperature T. The value of T controls the probability of accepting the replacement of a parent by an inferior child. The initial value of T depends on the range of fitness values, and according to Hunter (1995a) it should be about three times the typical change in fitness encountered in the early stages of the algorithm. The probability that an inferior child replaces a better parent is:

$$\text{Probability} = e^{(f(C)-f(P))/T}$$

where: f(C) = the fitness of the child;
f(P) = the fitness of the parent.

The annealing decay factor (df) specifies the rate at which T should be decreased. The value of df should be less than, but close to one (preferably 0.95-0.99). A lower df causes T to drop more quickly, which means that the algorithm will settle down more quickly. The probability of accepting an inferior child is gradually reduced, so that SA effectively starts off like unconditional replacement, and then gradually becomes conditional replacement. In the long run, SA gradually behaves more and more like RS. SA explores widely in the initial stages because it has a chance of escaping from local minima by making a sequence of apparently inferior mutations until it reaches a better general area of search space. SA then settles down in the convergence stage (Hunter, 1995a). The *replacement rate* was always 1 in the experiments. The default type 'proportion' was associated with the replacement rate, which means that the rate is interpreted as the proportion of the old population to be replaced, so the complete population is replaced in the experiments.

In the experiments the simplest *stopping condition* was chosen: a pre-determined given number of generations.

The minimum value of the population in GA decreases or remains the same because the option *elitism* was chosen. Elitism overrides the replacement rate which specifies that all strings should be replaced. This implies that the transport sequence with the lowest minimum value in the population is preserved until a better transport sequence is found so that it cannot be lost through crossover or mutation operators. In RS the minimum value of the population also only decreases or remains the same, because only improved solutions are accepted. The minimum value of the population in SA can increase in the first generations of SA, when non-improving sequences are accepted.

The selection, mutation, crossover and replacement parameters of a certain local search method can be specified in SUGAL by entering the correct value in a number field or by choosing the required type in a selection box (Figure 6-1).

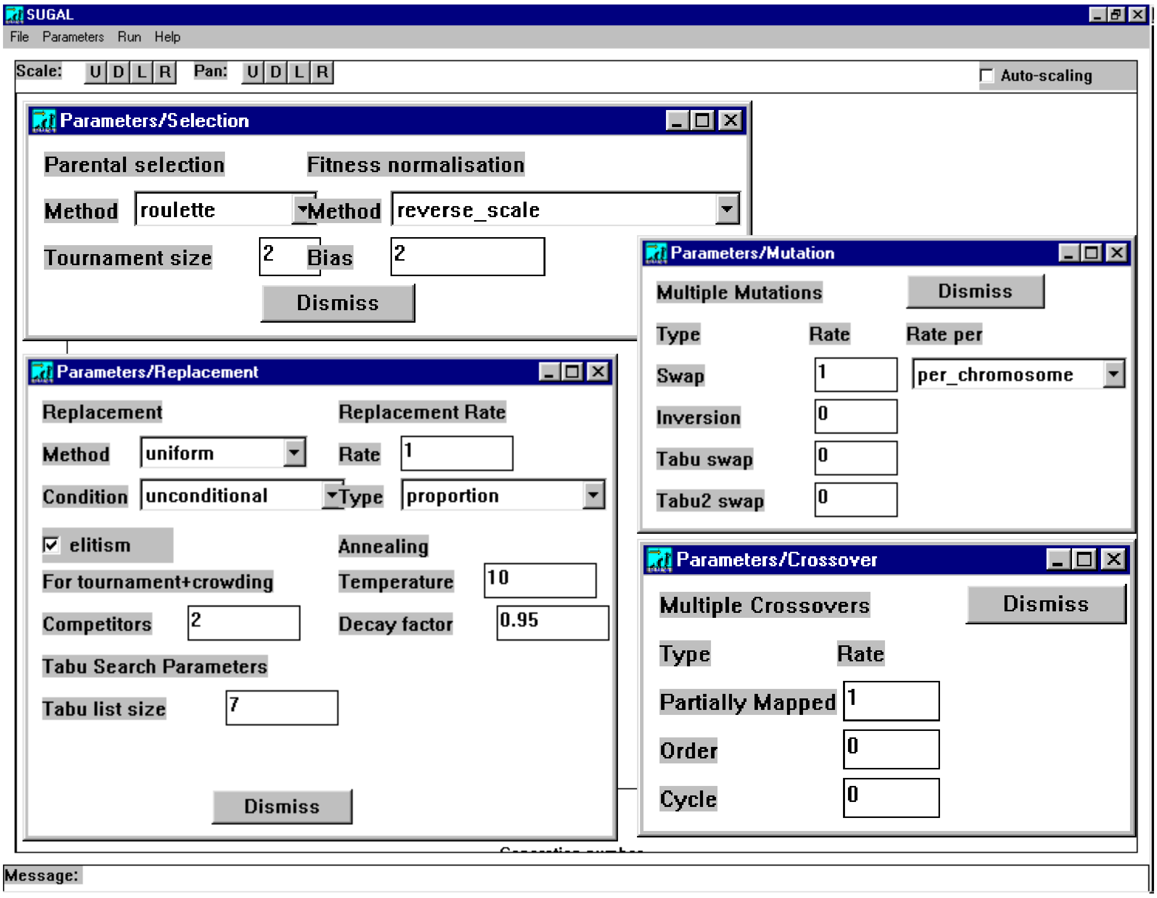


Figure 6-1. Interface to enter the parameter settings for selection, mutation, crossover and replacement in SUGAL.

When the parameters of the required local search method have been properly entered, the next action is to specify the control parameters of a run (Figure 6-2). These include the number of generations, the population size, the randomization method and a choice of the available display statistics. The display statistics specify which data about the current population have to be recorded during a run. These data can be divided into two groups: fitness statistics (minimum, maximum, mean and standard deviation of the fitness values) and diversity statistics. The diversity statistics give some measure for the differences between the strings. The diversity statistics vary in meaning according to the data-type. They were not used in the experiments. The specified statistics are shown in a graph during the run (Figure 6-3).

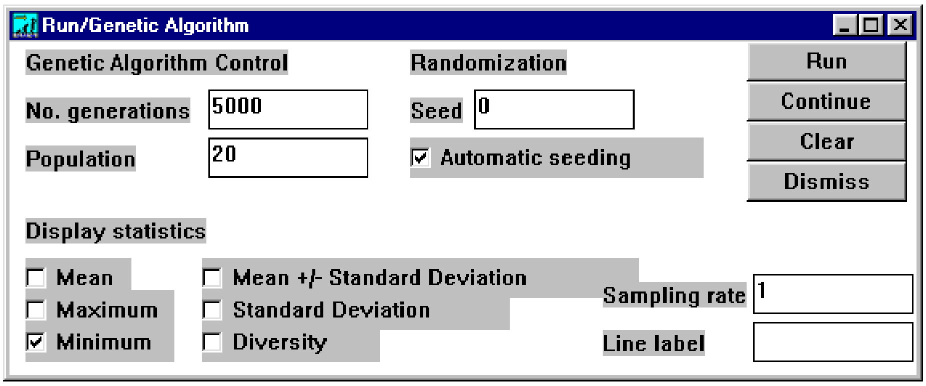


Figure 6-2. Interface to enter the control parameters that are required to start a run with a local search method in SUGAL.

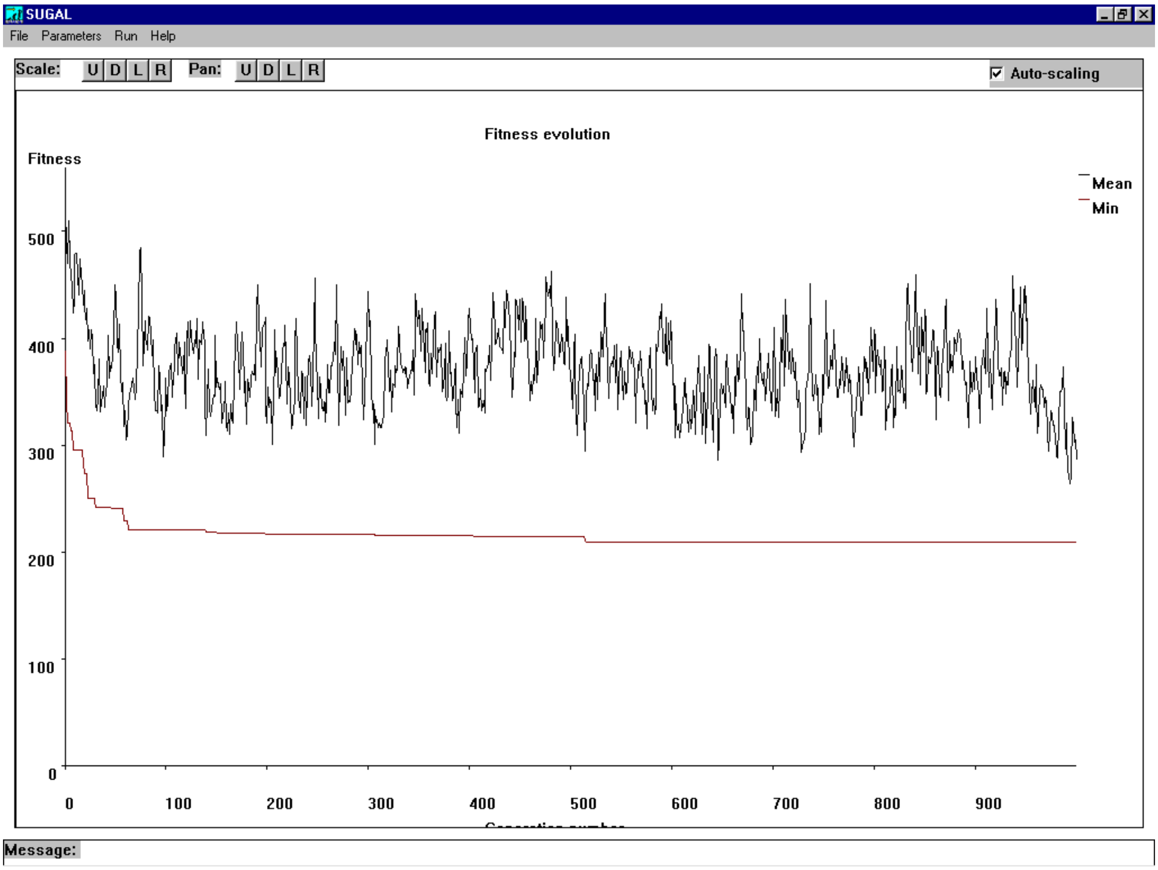
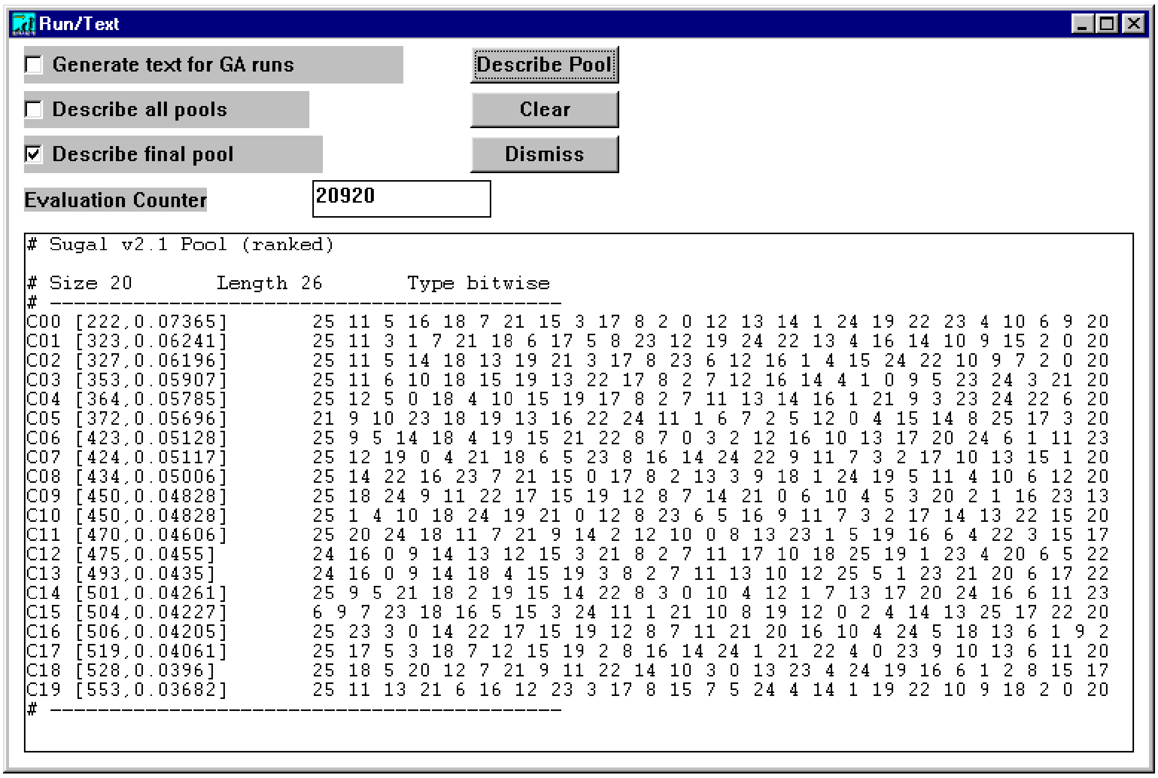


Figure 6-3. Example of the development of the minimum fitness value (lower line) and the mean fitness value (upper line) of the population during a run with GA-CX, population size 20 and 1000 generations (for the real-scale Case M1; Chapter 7).

When a run is finished the statistics can be sent to a file for analysis at a later moment, or they can be shown in a text window (Figure 6-4), which gives the exact description of each string in the population. The solutions are sorted by their fitness value. The line at the top of the screen gives the size of the population (20), the length of the string (26) and the type of string (bitwise). Then each individual string is described by an identification number for example, C00, the first string of the population; a fitness value for example, 222; a normalized fitness value for example, 0.07365 and finally the transport sequence (permutation) of the crop batch numbers, for example:

C00 [222, 0.07365]	25	11	5	16	18	7	..	..	4	10	6	9	20

Each number on the string has to be incremented by one to translate it to the crop batch number (so 0 = crop batch 1 etc.). This is a result of the zero counting coding method within SUGAL. The text window also gives the evaluation counter of the run.




```
# Sugal v2.1 Pool (ranked)

# Size 20        Length 26        Type bitwise
# -------------------------------------------
C00 [222,0.07365]      25 11 5 16 18 7 21 15 3 17 8 2 0 12 13 14 1 24 19 22 23 4 10 6 9 20
C01 [323,0.06241]      25 11 3 1 7 21 18 6 17 5 8 23 12 19 24 22 13 4 16 14 10 9 15 2 0 20
C02 [327,0.06196]      25 11 5 14 18 13 19 21 3 17 8 23 6 12 16 1 4 15 24 22 10 9 7 2 0 20
C03 [353,0.05907]      25 11 6 10 18 15 19 13 22 17 8 2 7 12 16 14 4 1 0 9 5 23 24 3 21 20
C04 [364,0.05785]      25 12 5 0 18 4 10 15 19 17 8 2 7 11 13 14 16 1 21 9 3 23 24 22 6 20
C05 [372,0.05696]      21 9 10 23 18 19 13 16 22 24 11 1 6 7 2 5 12 0 4 15 14 8 25 17 3 20
C06 [423,0.05128]      25 9 5 14 18 4 19 15 21 22 8 7 0 3 2 12 16 10 13 17 20 24 6 1 11 23
C07 [424,0.05117]      25 12 19 0 4 21 18 6 5 23 8 16 14 24 22 9 11 7 3 2 17 10 13 15 1 20
C08 [434,0.05006]      25 14 22 16 23 7 21 15 0 17 8 2 13 3 9 18 1 24 19 5 11 4 10 6 12 20
C09 [450,0.04828]      25 18 24 9 11 22 17 15 19 12 8 7 14 21 0 6 10 4 5 3 20 2 1 16 23 13
C10 [450,0.04828]      25 1 4 10 18 24 19 21 0 12 8 23 6 5 16 9 11 7 3 2 17 14 13 22 15 20
C11 [470,0.04606]      25 20 24 18 11 7 21 9 14 2 12 10 0 8 13 23 1 5 19 16 6 4 22 3 15 17
C12 [475,0.0455]       24 16 0 9 14 13 12 15 3 21 8 2 7 11 17 10 18 25 19 1 23 4 20 6 5 22
C13 [493,0.0435]       24 16 0 9 14 18 4 15 19 3 8 2 7 11 13 10 12 25 5 1 23 21 20 6 17 22
C14 [501,0.04261]      25 9 5 21 18 2 19 15 14 22 8 3 0 10 4 12 1 7 13 17 20 24 16 6 11 23
C15 [504,0.04227]      6 9 7 23 18 16 5 15 3 24 11 1 21 10 8 19 12 0 2 4 14 13 25 17 22 20
C16 [506,0.04205]      25 23 3 0 14 22 17 15 19 12 8 7 11 21 20 16 10 4 24 5 18 13 6 1 9 2
C17 [519,0.04061]      25 17 5 3 18 7 12 15 19 2 8 16 14 24 1 21 22 4 0 23 9 10 13 6 11 20
C18 [528,0.0396]       25 18 5 20 12 7 21 9 11 22 14 10 3 0 13 23 4 24 19 16 6 1 2 8 15 17
C19 [553,0.03682]      25 11 13 21 6 16 12 23 3 17 8 15 7 5 24 4 14 1 19 22 10 9 18 2 0 20
# -------------------------------------------
```

Figure 6-4. Results displayed in a text window after a run with SUGAL

6.8 Pre-selection of parameters and local search methods with simple test cases

An empirical performance analysis was performed, consisting of experimental runs with both simple and real-scale test cases, to show the effectiveness and the efficiency of different local search methods. In other words the ability of the methods to generate satisfactory internal transport sequences for transportable benches was studied and the required calculation time. The experiments with the simple test cases only looked at the effectiveness of the local search methods. An empirical performance analysis refers to an average-case situation for a certain local search method. Each different parameter setting of the studied local search algorithms (Table 6-2) was run a fixed number of times for the same problem instance (100 times for the small-scale cases mentioned in this chapter and 10 times for the real-scale cases in Chapter 7), using different initial solutions for each run. Such an experiment was performed with a fixed population size (20 for the simple test cases) and each run within an experiment was stopped after a specified fixed number of generations (2 000 for the simple test cases). This way statistical averages on the quality of the solutions and the required running time were obtained. This method of empirical performance analysis with repeated runs was suggested by several authors (Aarts & Korst,1989; Johnson & McGeoch 1997; Punnen & Aneja, 1995). The approach is not always enough to obtain tight confidence intervals, but enough to illustrate trends (Johnson & McGeoch 1997).

Table 6-2. The specifications in SUGAL of the parameters of the methods which were applied to the simple test cases (Hunter, 1995a).

Description	GA	SA	RS
Parental selection:	roulette	uniform	uniform
Fitness normalisation:	reverse scale	reverse scale	reverse scale
Bias:	2	2	2
Crossover type:	CX;PMX;OX	none	none
Rate:	1	0	0
Mutation Type:	swap	swap	swap
Rate:	0.1; 0.5; 1.0; 5.0	1	1
Rate per:	chromosome	chromosome	chromosome
Replacement method:	uniform	parental	parental
Condition:	unconditional	annealed	if improved
Rate:	1	1	1
Type:	proportion	proportion	proportion
Elitism:	yes	no	yes
Annealing temperature:	-	120	-
Decay factor:	-	0.95	-
Population size:	20	20	20

Anderson (1996) concludes that care is needed in reaching solid conclusions from these kind of experiments. He thinks that there is almost no end to the computational experiments that one might wish to carry out in order to reach a firm recommendation on the best way to carry out local search.

Six different problem instances (simple test cases) were drawn up to study the potential of three local search methods (GA, SA and RS) for generating a transport sequence for the internal transport planning problem. TS was not tested with the simple test cases. Two parking methods were used and compared in combination with the sequence generating local search methods: 'first available parking row' (Parking Method 1 = PM 1, Section 5.4.3) and 'first available parking row without output benches' (Parking Method 2 = PM 2, Section 5.4.4). Different parameter settings were applied for the GA (crossover method, swap rate, mutation method) to make a pre-selection of the most promising parameter values.

A *simple test case* was characterized by:

- 10 rows in the growing compartment;
- 10 available positions for transportable benches in each row;
- 20 (or 10 in one of the cases) output benches that have to be removed from the growing compartment;
- the exact location of the output benches (row number and position number);
- the exact location of the neutral benches (row number and position number);
- a total available free space of 10 positions;
- the available free space in each row (number of empty positions).

The exact description of each case (including the specification of the exact locations) is given in Appendix 6.1. The characteristics of a small test case remained unchanged for all experiments with the local search methods.

6.8.1 Evaluation criteria

Several criteria can be used to evaluate the quality of a local search method in relation to the internal transport control problem. These criteria concern the quality of the solutions (effectiveness) and the required calculation time (efficiency). It is advisable to look at both types of evaluation criteria, because a certain local search method might have a high score on a certain evaluation criterion, but a lower score on the other. For example, SA may always be able to find the theoretical minimum number of required transport movements, but it might take much more evaluations than GA or RS. Therefore, the criterion preferred (quality or speed) depends on the practical situation and what method has to be chosen in accordance with this criterion. Evaluation criteria for the effectiveness that were used for the simple test cases are:

- *Theoretical minimum obtained (TMO)*: Boolean indicating if a method found a transport sequence with the theoretical minimum number of transport movements (TM, Section 5.5.2) among hundred runs;
- *Frequency theoretical minimum (FTM):* the number of times that a method found a transport sequence with the theoretical minimum number of transport movements among hundred runs with 2 000 generations;
- *Mean minimum (MM):* the mean of all the minimum values for the number of transport movements that were found among hundred runs with 2 000 generations;
- *Best minimum (BM):* the best minimum value for the number of transport movements that was found among hundred runs with 2 000 generations;
- *Worst minimum (WM):* the worst minimum value for the number of transport movements that was found among hundred runs with 2 000 generations;
- *Distribution of the minimum values (DM):* the distribution between the best and the worst minimum of all minimum values that were found among hundred runs with 2 000 generations.

The simple test cases were not used to evaluate the efficiency of the local search methods. This analysis was performed with the real-scale test cases in Chapter 7.

6.8.2 Results

The experiments with the small test cases provided data on the mean of all the minimum values of 100 runs (MM; Table 6-3), about the number of times that the theoretical minimum was obtained within 100 runs (FTM; Table 6-4) and about the distribution of the minimum values between the best and the worst minimum of 100 runs (DM; Table 6-5). The distribution of the minimum values has only been reported for two cases. The other cases gave similar results.

Local search method

Comparing the best results of the three local search methods to each other shows a general trend. SA (with Parking Method 2) has a lower mean minimum value than RS (with Parking Method 2), in four out of six cases (Table 6-3). In two of these cases the mean minimum value of SA is substantially lower than RS (20.19 compared to 21.77 for Case S4 and 10.00 compared to 10.76 for Case S3). In all six cases SA has much lower mean minimum values than GA (with CX; swap rate 0.5; Parking Method 2). Furthermore the frequency of the theoretical minimum for SA (Table 6-4) was slightly higher than the frequency of the theoretical minimum for RS in two cases and dramatically higher in two other cases (100 compared to 29, and 81 compared to 2). In all cases SA found a higher frequency of the theoretical minimum than GA-CX. And finally the support of the distribution of the minimum values for SA (Table 6-5) was smaller than the support of the distribution of the minimum values for

GA. The worst minimum value for GA (with CX; swap rate 0.5; Parking Method 2) is ‘TM+4’ in both cases, ‘TM+1’ and ‘TM+3’ for RS (with Parking Method 2) and ‘TM+1’ and ‘TM’ for SA (with Parking Method 2). Thus, examining all the evaluation criteria it was found that SA was clearly more effective than GA for these small cases and slightly more effective than RS.

Parking method

All three local search methods (GA, SA and RS) gave much better results in combination with Parking Method 2 than in combination with Parking Method 1. The mean minimum value (Table 6-3) decreased substantially in most cases and the frequency of the theoretical minimum (Table 6-4) strongly increased in most cases (2.0-7.6 times higher for GA; 1.1-1.6 higher for SA and 1.2-6.0 higher for RS). In general, the results of GA and RS improve more in combination with Parking Method 2 than the results of SA. This is caused by the fact that the results of SA were already quite near to the TM in combination with Parking Method 1, which left fewer opportunities for improvement. In one case all three local search methods were able to find the theoretical minimum in combination with Parking Method 2, where as they did not succeed at all when used in combination with Parking Method 1. When SA was used in combination with Parking Method 2, the theoretical minimum was obtained in each of the 100 runs for almost all cases except one. When combined with Parking Method 1, SA only succeeded in achieving this extremely high frequency of the theoretical minimum in two cases. Using a different parking method actually means solving a different sequencing problem. The sequencing problem is easier for Parking Method 2, because Parking Method 2 solves a larger part of the parking problem than Parking Method 1. It uses a rule of thumb that finds better parking rows than Parking Method 1 and thus the total number of required transport movements is lower for Parking Method 2. Based on these better results, only Parking Method 2 was selected in the large-scale experiments.

Crossover method (GA)

Looking at the mean minimum value, the crossover method CX gave slightly better results in combination with Parking Method 1, than PMX in all six cases (Table 6-3 and 6-4). PMX in turn was slightly better than OX in five out of six cases. In four cases a transport sequence with TM was obtained by all three crossover methods at least once within the 100 runs. In one case none of the methods succeeded in finding a transport sequence with TM in combination with Parking Method 1, and in one case only CX and PMX found a transport sequence with TM. The crossover methods also produced different frequencies of the theoretical minimum.

Table 6-3. Mean minimum value (MM) of 100 runs with different parameter settings of three local search methods (Population Size 20 and 2 000 generations).

Local search method	Case:	S1	S2	S3	S4	S5	S6
	TM:	53	85	10	20	35	29
Genetic Algorithm							
CX; swap; rate 1.0; PM 1		55.83	87.98	10.77	23.23	36.49	32.03
PMX; swap; rate 1.0; PM 1		56.69	88.07	10.95	24.59	37.05	32.83
OX; swap; rate 1.0; PM 1		57.38	88.76	11.23	24.97	36.90	33.58
CX; swap; rate 0.1; PM 2		54.28	86.37	11.89	22.98	35.31	30.56
CX; swap; rate 0.5; PM 2		53.27	85.50	10.91	22.08	35.04	29.53
CX; swap; rate 1.0; PM 2		53.32	85.56	10.81	22.06	35.06	29.64
CX; swap; rate 5.0; PM 2		53.98	86.19	11.26	23.70	35.26	30.57
CX; inversion; rate 1.0; PM 2		53.65	85.97	10.66	22.71	35.08	29.84
Simulated Annealing (T=10, df=0.95)							
no X; swap; rate 1.0; PM 1		53.26	86.14	10.00	20.53	35.00	29.36
no X; swap; rate 1.0; PM 2		53.00	85.00	10.00	20.19	35.00	29.00
Random Search							
no X; swap; rate 1.0; PM 1		54.28	86.53	10.69	22.30	35.35	30.70
no X; swap; rate 1.0; PM 2		53.00	85.03	10.76	21.77	35.00	29.10

Table 6-4. The number of times that the theoretical minimum was obtained (FTM) of 100 runs with different parameter settings of three local search methods (Population Size 20 and 2 000 generations).

Local search method	Case:	S1	S2	S3	S4	S5	S6
Genetic Algorithm							
CX; swap; rate 1.0; PM 1		24	0	31	2	31	7
PMX; swap; rate 1.0; PM 1		9	0	21	1	14	2
OX; swap; rate 1.0; PM 1		8	0	14	0	21	1
CX; swap; rate 0.1; PM 2		38	35	5	2	71	27
CX; swap; rate 0.5; PM 2		80	69	26	3	96	59
CX; swap; rate 1.0; PM 2		73	60	31	4	94	53
CX; swap; rate 5.0; PM 2		36	27	10	0	75	9
CX; inversion; rate 1.0; PM 2		47	38	40	0	92	39
Simulated Annealing (T=10, df=0.95)							
no X; swap; rate 1.0; PM 1		88	0	100	51	100	64
no X; swap; rate 1.0; PM 2		100	100	100	81	100	100
Random Search							
no X; swap; rate 1.0; PM 1		46	0	34	3	81	15
no X; swap; rate 1.0; PM 2		100	97	29	2	100	90

Table 6-5. The distribution of the minimum values (DM) of 100 runs for Cases S4 and S6 with different parameter settings of three local search methods (Population Size 20 and 2 000 generations). Column 'TM' gives the number of runs, that the found minimum value was equal to the theoretical minimum (TM=20 for Case S4), the next column gives the number of runs equal to (TM+1=21 for Case S4), etc.

Minimum value: / Local search method:	TM	TM +1	TM +2	TM +3	TM +4	TM + 5	TM +6	TM +7	TM +8	TM +9	TM +10	TM +11
Case S4 (TM=20)												
Genetic Algorithm												
CX; sw 1.0; PM 1	2	12	18	31	14	16	5	1	0	1	-	-
PMX; sw 1.0; PM 1	1	5	5	11	26	22	17	10	2	1	-	-
OX; sw 1.0; PM 1	0	1	11	4	24	23	21	8	5	2	-	1
CX; sw 0.1; PM 2	2	12	22	40	9	8	5	0	2	-	-	-
CX; sw 0.5; PM 2	3	23	41	29	4	-	-	-	-	-	-	-
CX; sw 1.0; PM 2	4	21	42	31	2	-	-	-	-	-	-	-
CX; sw 5.0; PM 2	0	0	6	40	38	12	2	2	-	-	-	-
CX; inv 1.0; PM 2	0	14	34	28	17	5	2	-	-	-	-	-
SA (T=10, df=0.95)												
no X; sw 1.0; PM 1	51	45	4	-	-	-	-	-	-	-	-	-
no X; sw 1.0; PM 2	81	19	-	-	-	-	-	-	-	-	-	-
Random Search												
no X; sw 1.0; PM 1	3	22	31	32	10	2	-	-	-	-	-	-
no X; sw 1.0; PM 2	2	30	57	11	-	-	-	-	-	-	-	-
Case S6 (TM=29)												
Genetic Algorithm												
CX; sw 1.0; PM 1	7	16	18	25	8	16	7	2	1	-	-	-
PMX; sw 1.0; PM 1	2	8	16	23	13	18	10	9	0	1	-	-
OX; sw 1.0; PM 1	1	5	12	18	11	18	13	16	3	3	-	-
CX; sw 0.1; PM 2	27	23	28	16	3	2	0	1	-	-	-	-
CX; sw 0.5; PM 2	59	30	10	1	-	-	-	-	-	-	-	-
CX; sw 1.0; PM 2	53	34	10	2	1	-	-	-	-	-	-	-
CX; sw 5.0; PM 2	9	41	36	12	2	-	-	-	-	-	-	-
CX; inv 1.0; PM 2	39	40	19	2	-	-	-	-	-	-	-	-
SA (T=10, df=0.95)												
no X; sw 1.0; PM 1	64	36	-	-	-	-	-	-	-	-	-	-
no X; sw 1.0; PM 2	100	-	-	-	-	-	-	-	-	-	-	-
Random Search												
no X; sw 1.0; PM 1	15	26	37	18	4	-	-	-	-	-	-	-
no X; sw 1.0; PM 2	90	10	-	-	-	-	-	-	-	-	-	-

CX generated 1.5-3.5 times more transport sequences with TM than PMX and 1.5-7.0 times more than OX. The distribution of the minimum values also differs for these three crossover methods: the most favourable distribution is generated by CX and the least favourable by OX (Table 6-5).

Swap rate (GA)
The swap rate of the mutation operator ‘swap’ did have an impact on the results of GA. In four out of six cases, a swap rate of 0.5 gave the best results, with respect to the evaluation criteria mean minimum value and the frequency of the theoretical minimum (Table 6-3 and 6-4). However, only a little difference was found with a swap rate of 1.0, which found slightly better values for the two other cases. So the results of these two swap rates, 0.5 and 1.0, were more or less comparable. The results were not so good when the swap rate was lower (0.1) or raised (5.0) than when the swap rate was with the range 0.5-1.0. Evidently the capacity of GA to find new solutions depends partly on the mutation operator, so that its rate cannot be lowered too much. However, performing the mutation at a rate much higher than 1.0 disturbs the quality of the solutions too much. Therefore, 1.0 was chosen as the default swap rate in the real-scale experiments of Chapter 7.

Mutation method (GA)
In five out of six cases the mutation method swap clearly gave better results than the mutation method inversion, both with a rate of 1.0 per chromosome (Table 6-3 and 6-4). The mean minimum value of swap mutation was lower than inversion mutation. The frequency of the theoretical minimum of swap mutation was 1.4-1.6 times higher than inversion mutation. In one case, inversion mutation was unable to find a transport sequence with TM at all. Therefore, only the mutation method swap was considered in the real-scale experiments reported in Chapter 7.

Calculation time
The calculation time of each run of the simple test cases was measured by hand with a chronometer, because it is not possible in SUGAL to record the CPU time of a run automatically. SUGAL was the only process that was run on the PC, during the measurements. All experiments involved an equal number of fitness function evaluations, because the population size and the number of generations remained the same. So the evaluation counter was the same for all experiments. The calculation time varied moderately on a 133 MHz Pentium(r) PC with 32.0 MB of RAM memory, depending on a number of factors (Table 6-6). The length of the string had a large impact. The calculation times of the problems with a string length 20 were about 1.5 times longer than the calculation times of the problem with a string length 10. The chosen local search method had a much smaller influence on the calculation time: SA and RS (both with Parking Method 2) needed about 0.8-0.9 times the calculation time of GA (CX; swap rate 1,0; Parking Method 2). Only a very

small, negligible difference was found between Parking Method 1 and 2. The parameter settings of the GA did influence the calculation time. The slowest crossover method OX needed, on average, 1.1-1.2 times more calculation time than the fastest crossover method CX. Increasing the mutation rate also led to an increase in calculation time: mutation rate 5.0 needed 1.1-1.2 times more calculation time than mutation rate 0.1. These results show that the calculation times will differ for different parameter settings. However, part of the difference in the calculation time might be caused by inefficient coding of the different local search methods and their subroutines. Therefore, it was decided to compare large real-scale experiments on the basis of the evaluation counter alone, because this measure gives a more objective comparison of the local search methods.

Table 6-6. Calculation time in centi-minutes for a fixed number of evaluations with the simple test cases using different parameter settings for the local search methods (Population Size 20 and 2 000 generations).

Local search method Case:	S3	S1	S2	S4	S5	S6
String length:	10	20	20	20	20	20
Genetic Algorithm						
CX; swap; rate 1.0; PM 1	20	32	37	34	31	33
PMX; swap; rate 1.0; PM 1	21	34	39	39	33	36
OX; swap; rate 1.0; PM 1	21	36	40	41	36	39
CX; swap; rate 0.1; PM 2	20	30	35	26	27	27
CX; swap; rate 0.5; PM 2	21	31	37	30	29	30
CX; swap; rate 1.0; PM 2	22	34	40	34	30	32
CX; swap; rate 5.0; PM 2	24	39	44	42	35	37
CX; inversion; rate 1.0; PM 2	21	35	40	37	32	34
Simulated Annealing (T=10, df=0.95)						
PM 1	18	30	34	29	28	28
PM 2	18	31	36	28	28	28
Random Search						
PM 1	18	30	34	29	28	28
PM 2	18	31	35	28	28	28

6.9 Conclusions

Some preliminary conclusions can be drawn from experiments with small-scale test cases. These conclusions will be further verified in Chapter 7 with the experiments for the real-scale test cases.

First of all the experiments showed that it is possible to obtain optimal transport sequences with the theoretical minimum number of transport movements for these simple test cases with all three local search methods GA, SA and RS. However, it should be noted that the problem instances of the simple test cases are only very small. The effect of the three local search methods may change for larger internal transport planning problems (Chapter 7). The total available free space in the growing compartment will also influence the results of the local search methods. With a smaller manoeuvring area the problem will become more complex. Nevertheless, these results do indicate a major trend: SA generates high quality transport sequences with a much higher frequency than the other two local search methods. SA gave the best results of the three local search methods so far, followed closely by RS, which always performed better than GA in combination with CX for these simple test cases.

All three local search methods gave much better results in combination with Parking Method 2, than in combination with Parking Method 1. Based on these results only Parking Method 2 was used in the experiments with the real-scale test cases.

The chosen crossover method clearly influenced the results of GA. The crossover method CX gave better results than PMX, which in turn was superior to OX. The mutation method swap performed better for GA than the mutation method inversion. For the mutation method swap, a mutation rate of 0.5-1.0 per chromosome gave better results than a lower or a higher mutation rate.

Experiments that varied the population size, changed the number of generations or used different values for the SA cooling schedule, were only performed for one or two of the small-scale test cases. The results of these experiments have not been reported here, because they were incomplete and were not as yet sufficient to prove any substantial differences. However, the preliminary results of these few experiments did indicate that differences could be expected. Therefore, these parameters were included in the experiments with the real-scale test cases.

The calculation time depended on the length of the string in the problem, on the chosen local search method and on the setting of its parameters. Still the evaluation counter was chosen to compare the efficiency performance of different local search methods instead of the calculation time, because it is a more objective criterion.

7. Performance of local search methods applied to real-scale test cases

7.1 Introduction

The small test cases, that were studied in Chapter 6 constituted a highly simplified planning situation. More complex, real-scale test cases were therefore devised to analyse the performance (effectiveness and efficiency) of the selected local search methods in more realistic internal transport planning situations. However, these real-scale test cases were still transformed to the simplified internal transport control problem that was described in Chapter 5. The most important simplifications are that the parked obstructing transportable benches (both output and neutral) are not transported back to their original row; the AGV can only transport one transportable bench at a time; the control problem of the next period is not taken into account while determining the parking rows; and finally no problems were studied in which the transport sequence was already partly fixed before the sequencing process began.

7.1.1 Description of real-scale test cases

The real-scale test cases were constructed from practical data recorded at pot plant nursery ‘Kwekerij de Goede Hoop’ (KDGH; Section 3.5) on specific days. Specifications of the real-scale test cases are presented in Table 7-1. An example of the exact description of two real-scale test cases is given in Appendix 7-1. A real-scale test case is characterized by:

- 26 rows in the growing compartment;
- 61 available positions for transportable benches in each row;
- 47-93 output benches that have to be removed on a specific day from the growing compartment;
- 103-133 input benches that have to be added on a specific day to the growing compartment;
- the exact location of the output benches at the beginning of the day (row and position number);
- the exact location of the neutral benches at the beginning of the day (row and position number);
- the total available free space of 71-188 positions at the beginning of the day (4.5-11.9%);
- the available free space in each row at the beginning of the day.

The number of output benches in a real-scale test case refers to all transportable benches that had to be removed from the growing compartment on a specific day at pot plant nursery KDGH. The number of input benches refers to transportable benches with potted or spaced crop batches that had to be transported to the growing compartment on the same day. Each case has a theoretical minimum (TM) number of required transport movements. TM was calculated using the method described in Section 5.5.2.

Table 7-1. Specifications of *real-scale* test cases with a growing compartment of 26 rows each capable of holding 61 transportable benches. So the maximum number of transportable benches that can be allocated in the growing compartment is 1586 for all cases. However, the number of empty positions is case-specific.

Case	Date at KDGH	Number of output benches (OB)	Number of obstructing neutral benches (NB)	Number of input benches (IB)	Number of obstructing output benches (OO)	Theoretical minimum (TM)	Number of decision variables (crop batches) on string (DV)	Number of empty positions (EP)
Group O: Only output benches								
O1	1/4/94	72	79	0	0	151	72	129
O2	1/2/95	47	18	0	0	65	47	71
O3	1/9/94	77	104	0	0	181	77	181
O4	2/5/94	93	132	0	0	225	93	188
Group C: Combination of output & input benches								
C1	1/4/94	72	79	103	0	254	175	129
C2	1/2/95	47	18	116	0	181	163	71
C3	1/9/94	77	104	105	0	286	182	181
C4	2/5/94	93	132	133	0	358	226	188
Group M: More than one output bench per crop batch								
M1	1/4/94	72	79	0	17	168	26	129
M3	1/9/94	77	104	0	5	186	35	181
M4	2/5/94	93	132	0	10	235	43	188

The real-scale test cases were divided into three groups (Table 7-1). The first *Group O* of test cases constitute a planning situation where *only output benches* (Case O1-O4) have to be transported from the growing compartment to the working area. In this planning situation each crop batch contains only one output bench. An output bench may contain pot plants that have to be either spaced or harvested. Determining the internal transport sequence in this planning situation with only output benches is less complex than it is in a real situation because the internal transport sequence neglects input benches. Input benches do not belong to this planning situation and therefore input benches have to be sequenced separately either before or after the transport sequence with output benches. The number of decision variables on a string is equal to the number of crop batches with output benches that have to be sequenced. In Group O the number of crop batches is the same as the number of transportable benches.

A second, more complex *Group C* of test cases was devised with *a combination of output and input benches* (Case C1-C4). These real-scale test cases provide an example of a planning situation where the stream of output benches is integrated with the stream of input benches to form one single internal transport sequence. In this planning situation each crop batch also contains only one transportable bench. The initial situation in the growing compartment of these real-scale test cases with a combination of input and output benches was identical to the initial situation of the cases of Group O with only output benches. The only extension to the test cases of Group O were extra input benches that have to be added to the growing compartment during the removal of the output benches. The addition of input benches to the growing compartment will decrease the manoeuvring area during the internal transport process. A smaller manoeuvring area makes it more difficult to avoid the obstruction of output benches during relocation movements and therefore it is also more difficult to obtain a transport sequence with TM. The number of decision variables on a string is equal to the sum of the number of crop batches with output benches and the number of crop batches with input benches that have to be sequenced. In Group C the number of crop batches is again the same as the number of transportable benches.

The third *Group M* of real-scale test cases represent a planning situation where a single *crop batch may contain more than one output bench* (Case M1, M3 and M4). Input benches are not taken into account in this planning situation. All output benches of a specific crop batch have to be removed from the growing compartment, before the output benches of the next crop batch can be transported. This way the output benches of a crop batch stay together which is an advantage when pot plants have to be spaced or harvested. In the previous two Groups O and C it was assumed that a crop batch only contained one transportable bench. So it was not necessary to look for more transportable benches of the same crop batch in those cases. Group M with more than one output bench per crop batch was derived from the test cases in the first Group O with only output benches. This was done by clustering crop batches with only one transportable bench into larger crop batches. Case O2 was not converted into a new case, because the number of crop batches in the new situation would be too low to qualify it as a complex real-scale test case. In Group M the number of decision variables on a string is equal to the number of different crop batches. However, it should be noted that the number of crop batches in Group M is lower than in Group O at an equal number of output benches. This will reduce the number of decision variables in Group M compared to Group O.

The complexity of the cases within a group and between the groups depends on several factors, including the number of output benches, the number of obstructing neutral benches, the number of input benches, the number of decision variables, the exact location of the output and neutral benches, the number of empty positions and their distribution. It is not possible to establish the exact complexity of a case without looking at the initial situation of all transportable benches in the growing compartment. In general it can be stated that a case is more complex when its available manoeuvring area is smaller or when more transportable benches have to be transported.

With the specified real-scale test cases various experiments were performed. An experiment consisted of performing 10 different runs for a local search method using a specific set of parameter values (Table 7-2 and 7-3). An equivalent number of runs was used by several other authors (Johnson & McGeoch 1997; Punnen & Aneja, 1995) and it was found to be an appropriate number after analysing the experiments with the small test cases. Each run in an experiment consisted of a fixed number of generations (G) that was specified as a stopping criterion at the beginning of the run. The number of generations depended on the characteristics and complexity of the case, on the population size and on the local search method. One run with population size N involves N separate strings - each with a transport sequence of crop batch numbers - that develop for the specified number of generations. At each generation only (N-1) strings were evaluated with the fitness function and not all N. The best string in the population does not have to be evaluated because elitism was chosen for all local search methods. Elitism excludes the best string from being changed from one generation to the next. In Section 6.7 it was already mentioned that an evaluation counter is needed to be able to compare the efficiency of the runs of an experiment with a different population size or a different number of generations. The results of two local search methods can only be compared if they are obtained using the same number of evaluations. Comparing the evaluation counter is only useful within a test case because the required number of generations, and indirectly the number of evaluations, varied too much among the different test cases. This made it pointless to compare the evaluation counters of different cases. The evaluation counter (EC) of a run was calculated as follows:

$$EC = (N-1) * G$$

All experiments were performed with Parking Method 2 (see Section 5.4.4) because it clearly gave much better results than Parking Method 1 (see Section 5.4.3) in the experiments with the simple test cases (Section 6.8). No experiments were performed with the more sophisticated parking methods Clussort and Plussort (Section 5.4.1 and 5.4.2).

Table 7-2. The parameters of the local search methods that were applied to the real-scale test cases with *only one* output or input bench per crop batch (Case O1-O4 and C1-C4).

Description	GA	SA	RS
Parental selection:	roulette	uniform	uniform
Fitness normalisation:	reverse scale	reverse scale	reverse scale
Bias:	2	2	2
Crossover type:	CX	none	none
Rate:	1	0	0
Mutation Type:	swap	swap	swap
Rate:	1	1	1
Rate per:	chromosome	chromosome	chromosome
Replacement method:	uniform	parental	parental
Condition:	unconditional	annealed	if improved
Rate:	1	1	1
Type:	proportion	proportion	proportion
Elitism:	yes	yes	yes
Annealing temperature:	-	120	-
Decay factor:	-	0.999	-
Population size:	20	20; 5; 2	20

Table 7-3. The parameters of the local search methods that were applied to the real-scale test cases with *more than one* transportable bench per crop batch (Case M1, M3 and M4).

Description	GA	SA	RS
Parental selection:	roulette	uniform	uniform
Fitness normalisation:	reverse scale	reverse scale	reverse scale
Bias:	2	2	2
Crossover type:	CX; PMX; OX	none	none
Rate:	1	0	0
Mutation type:	swap	swap; TS-swap	swap; TS-swap
Rate:	1	1	1
Rate per:	chromosome	chromosome	chromosome
Replacement method:	uniform	parental	parental
Condition:	unconditional	annealed	if improved
Rate:	1	1	1
Type:	proportion	proportion	proportion
Elitism:	yes	yes	yes
Annealing temperature:	-	60; 120; 180	-
Decay factor:	-	0.95; 0.995; 0.999; 0.9995	-
Tabu list size:	-	3; 7; 11	3; 7; 11
Population size:	40; 20; 10	20; 5; 2	20; 5; 2

7.1.2 Evaluation criteria

Again several criteria were available to evaluate the quality of a local search method in relation to the internal transport control problem. Two groups of evaluation criteria were used for the real-scale test cases: criteria that concern the *effectiveness* of the local search method (quality of the solutions) and criteria that judge the *efficiency* of the local search method (required calculation time, or number of function evaluations).

The evaluation criteria to judge the *effectiveness* of the local search method were identical to the ones used for the experiments with the simple test cases (Section 6.8.1):

- *Frequency theoretical minimum (FTM):* the number of times that a method found a transport sequence with the theoretical minimum number of transport movements among ten runs with a fixed number of generations;
- *Mean minimum (MM):* the mean of all the minimum values for the number of transport movements that were found among ten runs with a fixed number of generations;
- *Best minimum (BM):* the best minimum value for the number of transport movements that was found among ten runs with a fixed number of generations;
- *Worst minimum (WM):* the worst minimum value for the number of transport movements that was found among ten runs with a fixed number of generations.

The best minimum (BM) and the worst minimum (WM) of the runs in an experiment were always given as an absolute distance to TM (+0, +1, +2, etc.). The relative distance to TM can be deduced from these values if necessary. The distribution of the minimum values (DM) was not used in the experiments with the real-scale test cases, because the number of runs was too low to give a clear description of the distribution. Three evaluation criteria were added to judge the *efficiency* of a local search method:

- *Mean Evaluation Counter (MEC):* the mean number of function evaluations that had to be made before a transport sequence with a certain number of transport movements was found among ten runs;
- *Best Evaluation Counter (BEC):* the lowest number of function evaluations that had to be made before a transport sequence with a certain number of transport movements was found among ten runs;
- *Worst Evaluation Counter (WEC):* the highest number of function evaluations that had to be made before a transport sequence with a certain number of transport movements was found among ten runs.

A problem occurred at the efficiency judgement when the transport sequence with TM was not found within the specified number of generations, leaving the run without an evaluation counter. When this occurred, the limited number of allowed evaluations that was specified at the start of the run as stopping condition was taken as the estimated evaluation counter for the run. However, the correct evaluation counter, that is the number of evaluations needed to really obtain a sequence with TM, will be (much) higher or might not be found at all. So, this approach only gave an estimated value for the mean evaluation counter (MEC) and this could sometimes be (much) too low. However, it is necessary to take these unsuccessful runs into account during the statistical analysis and the best guess for the evaluation counter is the limited number of allowed evaluations.

7.1.3 Statistical analysis

A statistical analysis was performed on all experiments to verify whether two local search methods or two different parameter settings of the same local search method were significantly different. A hypothesis-testing procedure was used to compare two mean minimum values (MM) or two mean evaluation counters (MEC) of 10 runs with a certain parameter setting of the local search method. The two-sample hypothesis tests of means use the data of the experimental runs to see if there is a statistically significant difference between the means of the two experiments with 10 runs (Hoshmand, 1998). The null hypothesis (H_0) and alternative hypothesis (H_1) for differences between two experiments A and B was declared as for the mean minimum values as:

H_0: $MM_A - MM_B = 0$

H_1: $MM_A - MM_B \neq 0$

and for the mean evaluation counters as:

H_0: $MEC_A - MEC_B = 0$

H_1: $MEC_A - MEC_B \neq 0$

The null hypothesis that was tested states that there is no difference between the two experiments A and B. The Student-distribution was used to draw conclusions about the population means. It was assumed that the mean minimum value and the mean evaluation counter of both sampled populations were approximately normally distributed with equal standard deviations (Sdev). A *95% confidence interval* was calculated for the difference of the means of the minimum values:

$(MM_A - MM_B) \pm t_v \times sed$

and for the means of the evaluation counters:

$(MEC_A - MEC_B) \pm t_v \times sed$

In this formula sed is the standard error of difference of the means and t_v is the critical value of the Student-distribution at $v=(n_A+n_B-2)$ degrees of freedom. Because all experiments contained 10 runs with the same parameter setting ($n_A=10$ and $n_B=10$), t_v is always the same: $v=18$ and $t_{18}=2.101$ at a level of significance $\alpha=0.05$. The null hypothesis was rejected if zero was not part of the confidence interval. This indicated that the results of the two experiments differed significantly. The observed significance level, the *p value*, was calculated in addition to the 95% confidence interval. The p value is the smallest value of α for which the test results are statistically significant (Hoshmand, 1998).

The results of the different experiments with different parameter settings of local search methods are given in Appendix 7.2. The general structure of the analysis is given in Table 7-4.

Table 7-4. The general structure of the analysis with different local search methods. A reference is given to the tables with results in Appendix 7.2.

Group of test cases	Local search method	Population size	Cooling schedule (SA)		Tabu Search swap	Crossover method (GA)
			Decay factor	Temp.		
O. Only output benches	Table 1	Table 2 (SA)	-	-	-	-
C. Combination of output & input benches	Table 3 Table 4	Table 5 (SA)	-	-	-	-
M. More than one output bench per crop batch	Table 6	Table 7 (SA) Table 8 (RS) Table 9 (GA)	Table 10	Table 11	Table 12 (SA) Table 13 (RS)	Table 14

7.2 Results Group O: only output benches

7.2.1 Local search method

Three local search methods, SA (temperature 120 and decay factor 0.999), RS and GA with cycle crossover (CX) were compared at Population Size 20 (Appendix 7.2, Table 1). SA was most effective, because it was the only local search method that found a transport sequence with TM in all 10 runs of an experiment. The mean minimum value of SA was significantly lower than the mean minimum value of both RS and GA-CX. The local search methods RS and GA-CX did not find the transport sequence with TM in any run. The exception was GA-CX that found a transport sequence with TM once in one of the four cases. In two cases RS found a significantly lower mean minimum value than GA-CX. The highest worst minimum value was always found by GA-CX, making it a less suitable local search method if bad transport sequences were to be avoided. However, GA-CX did find a lower best minimum value than RS had done in two cases.

The efficiency of SA was the best of the three local search methods, always having a lower mean evaluation counter. However, this difference was not analysed statistically, because RS and GA-CX never found a transport sequence with TM. RS and GA-CX were terminated after a fixed number of generations, which was then taken as the estimated value for the evaluation counter. Therefore, the mean evaluation counter values of RS and GA-CX were equal within a case. So the experiments with RS and GA-CX had a standard deviation of zero, which made it impossible to perform the proposed statistical analysis. An example of the development of the mean number of transport movements for the three local search methods is given in Figure 7-1. This Figure clearly shows that SA outperforms RS, which in turn outperforms GA-CX.

7.2.2 Population size

The population size was varied for SA at a constant temperature 120 and a constant decay factor 0.999 to study the effect of the population size on the efficiency of this local search method (Appendix 7.2, Table 2). Population Size 2 always gave the lowest mean evaluation counter and also the lowest best and worst evaluation counter. In all four cases Population Size 2 gave significantly better results than Population Size 20 and in three cases also better than Population Size 5. Population Size 5 had a significantly lower mean evaluation counter than Population Size 20 in all four cases. This leads to the conclusion that for SA using Population Size 2 is the most efficient way to obtain a transport sequence with TM for this group of cases. An example of the development of the mean number of transport movements for these three population sizes is given in Figure 7-2. SA with Population Size 2 descends most rapidly towards a mean minimum value that is equal to TM (which is 181 in Case O3).

7.2.3 Calculation time

The calculation time of a run with SA for the real-scale test cases in this Group O varied between 81 and 362 centi-minutes per 100 000 evaluations on a 350 MHz Pentium Pro(r) PC with 64.0 MB RAM memory (Table 7-5). The string length influenced the calculation time. A longer string needed a longer calculation time per 100 000 evaluations. When the mean evaluation counter of an experiment is taken into account this means that the mean calculation time for each run in the experiments with SA, took between 1.5 and 57 minutes. So it took about 0.3-9.5 hours to complete one experiment of 10 runs. The order of magnitude for the calculation time of one run with a mean evaluation counter in Cases O1, O3 and O4 is too high for practical applications in a pot plant nursery. In an actual planning situation the manager will need to re-calculate an internal transport sequence several times a day, which necessitates a maximum calculation time of 5-10 minutes per calculation run. Shorter calculation times will certainly be possible in the near future due to a rapid growth of the processing capacity of Personal Computers. However, this will only reduce part of the calculation time problem in this group of test cases.

Table 7-5. Calculation time in centi-minutes with Simulated Annealing for the real-scale test cases using temperature 120 and decay factor 0.999.

Time	Case:	O1	O2	O3	O4
	string length:	72	47	77	93
per 100 000 evaluations		237	81	235	362
to complete one run with mean number of evaluations		5702	156	1102	4728

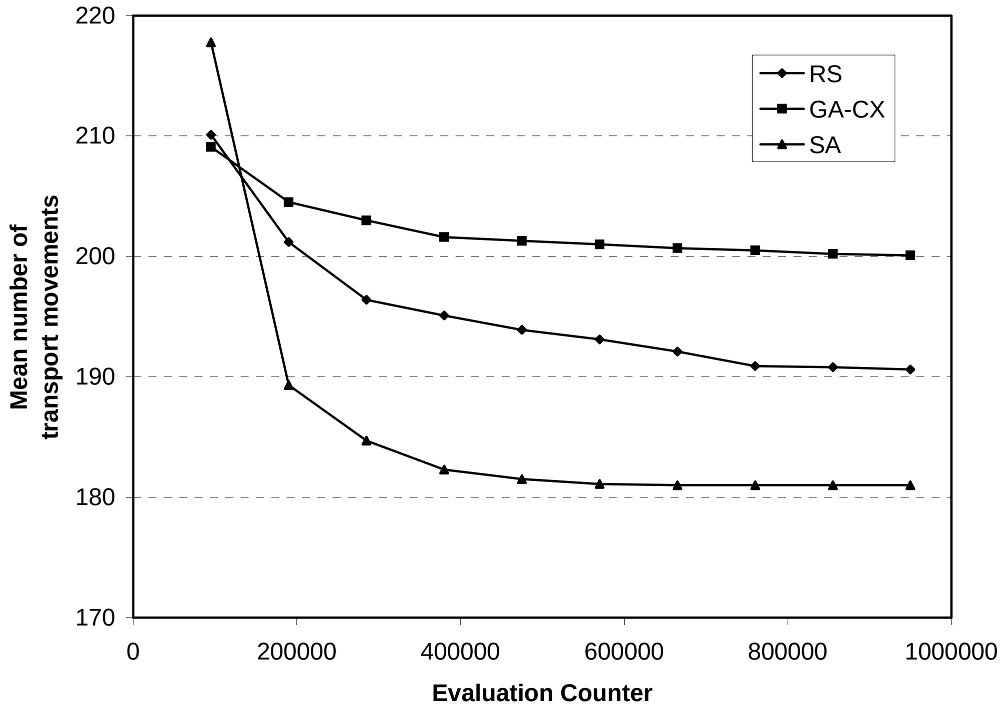


Figure 7-1. The development of the mean number of transport movements for the local search methods Simulated Annealing with temperature 120 and decay factor 0.999 (SA), Random Search (RS) and the Genetic Algorithm with cycle crossover (GA-CX) at Population Size 20 (Case O3).

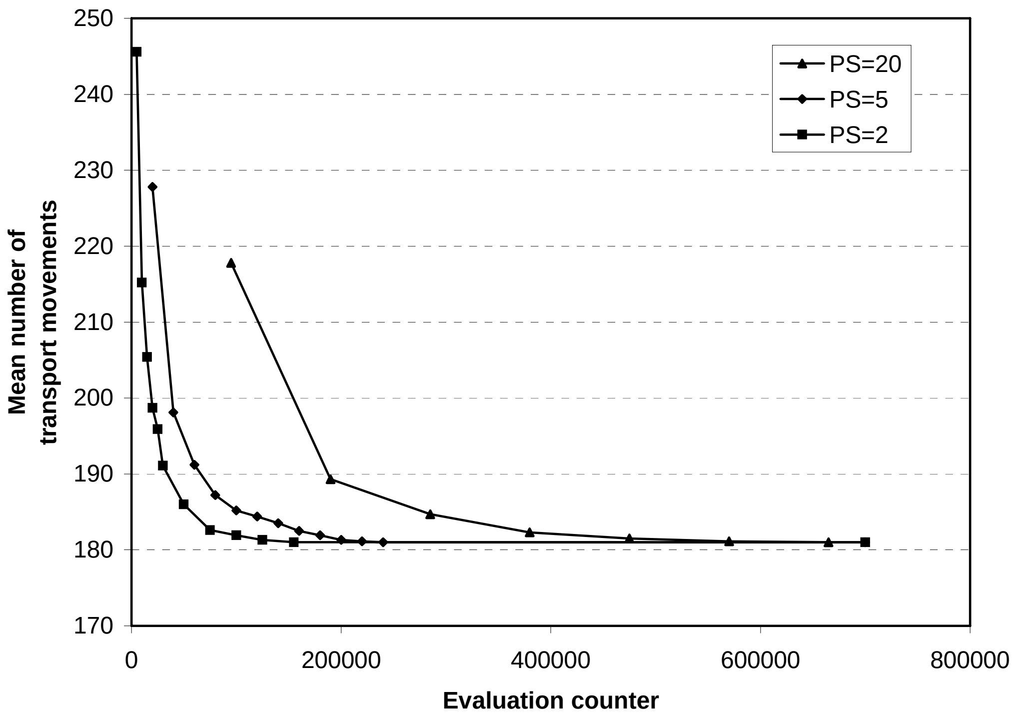


Figure 7-2. The development of the mean number of transport movements for three population sizes (PS) for Simulated Annealing with temperature 120 and decay factor 0.999 (Case O3).

7.3 Results Group C: combination of output and input benches

7.3.1 Local search method

In this Group C with a combination of output and input benches again three local search methods, SA (temperature 120 and decay factor 0.99995), RS and GA-CX were compared at Population Size 20 (Appendix 7.2, Table 3). This experiment with all three local search methods was performed for only one of the cases of Group C with a combination of output and input benches because of the extensive calculation times involved. None of the three local search methods succeeded in finding a transport sequence with TM (which is 254 for Case C1). However, SA found a significantly better mean minimum value than RS and GA-CX. RS in turn was significantly better than GA-CX. These results on the performance of the local search methods are in accordance with the results of the previous Group O with only output benches.

The development of the mean number of transport movements in Figure 7-3 shows that both RS and GA-CX start to descend more rapidly than SA, but that these methods get stuck on a much higher level than TM. SA descends more slowly due to the high decay factor 0.99995, but after 2 300 000 evaluations the mean minimum value drops below the value of RS and GA-CX, and finally SA approaches TM. The best minimum value that was obtained by SA was 'TM+7', which was 11 transport movements lower than best minimum value that was obtained by RS and 15 transport movement lower than GA-CX.

Given the better results of SA for the first case of Group C some further experiments were performed with only SA for the other three test cases in Group C. The runs were allowed a very large number of evaluations (up to 10 000 000) to obtain a transport sequence with TM as often as possible. When sufficiently long runs were taken, SA was able to find transport sequences with TM in all 10 runs for two of the three new cases and in 3 out of 10 runs for another case (Appendix 7.2, Table 4). This showed that it is possible to obtain transport sequences with TM for the difficult test cases in Group C when a sufficiently large number of evaluations is allowed.

7.3.2 Population size

In this Group C again the population size was varied for SA at a constant temperature 120 and a constant decay factor (0.99995 for case C1 and 0.999 for the other three cases) to see if this influenced the effectiveness and the efficiency of this local search method (Appendix 7.2, Table 5). The effects of the population size on the results of SA were compared at a fixed number of evaluations that was about 4-12 times lower than the number of evaluations that were used previously to obtain

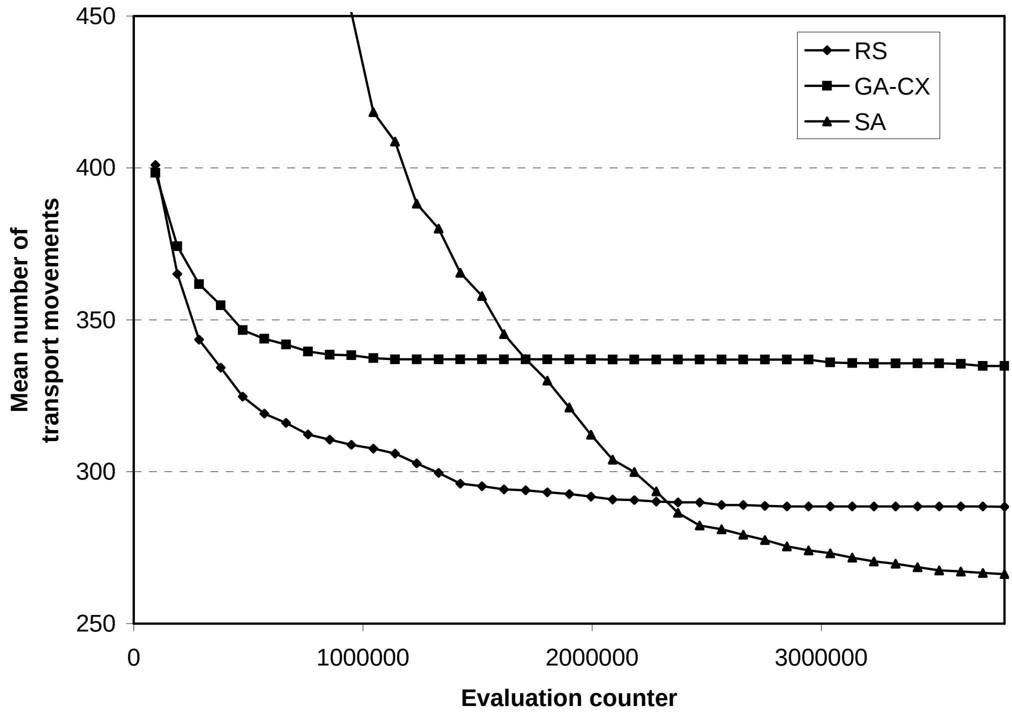


Figure 7-3. The development of the mean number of transport movements for the local search methods Simulated Annealing with temperature 120 and decay factor 0.99995 (SA), Random Search (RS) and the Genetic Algorithm with cycle crossover (GA-CX) at Population Size 20 (Case C1).

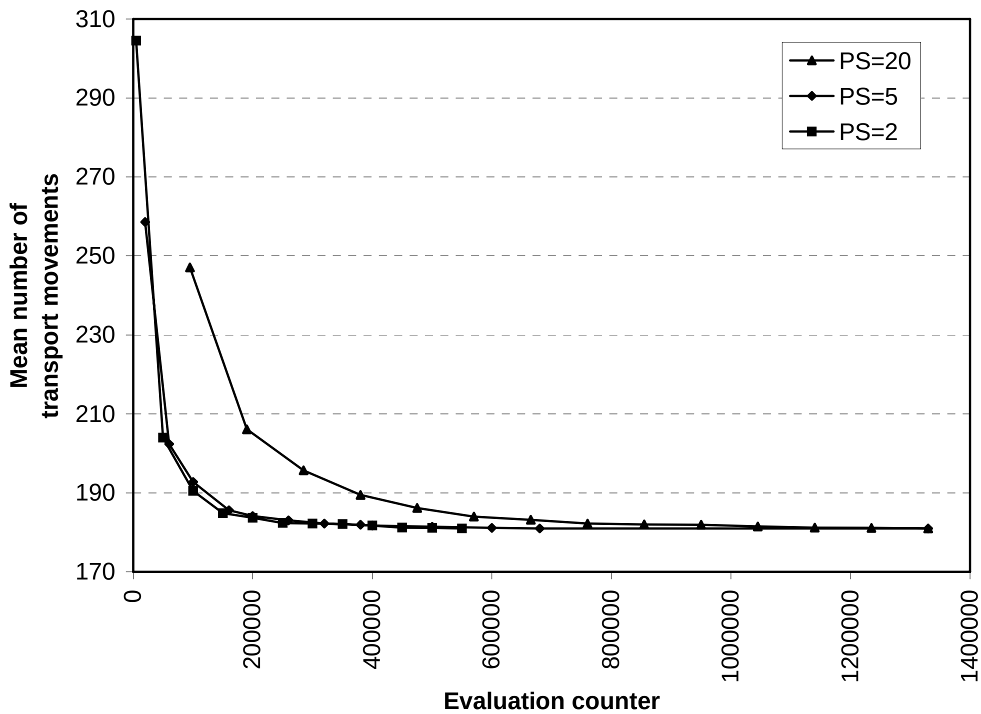


Figure 7-4. The development of the mean number of transport movements for three population sizes (PS) for Simulated Annealing with temperature 120 and decay factor 0.999 (Case C2).

transport sequences with TM for Population Size 20 (Appendix 7.2, Table 4). Therefore, the frequency of transport sequences with TM was lower for two cases compared to the longer runs in Section 7.3.1.

Population Size 2 and 5 were more effective than Population Size 20. The mean minimum values in the experiments with Population Size 2 and 5 were significantly lower than in the experiments with Population Size 20. In one case the mean minimum value of Population Size 2 was also significantly lower than Population Size 5. These results of the population size are in accordance with the results of the previous Group O with only output benches.

Efficiency could only be analysed for one case and this analysis also gave comparable results with the previous Group O. Population Size 2 and 5 gave a significantly lower mean evaluation counter than Population Size 20, but no difference was found between Population Size 2 and 5. An example of the development of the mean number of transport movements in this last instance is given in Figure 7-4, where the lines of Population Size 2 and 5 proceed close to each other and reach TM much faster than Population Size 20.

7.3.3 Calculation time

The calculation time of each run of the real-scale test cases in Group C with a combination of output and input benches were about the same as the calculation time for the cases in Group O with only output benches: between 156 and 411 centi-minutes per 100 000 evaluations with SA (Table 7-6). The string length influenced the calculation time again just like it did in Group O. A longer string length again needed a longer calculation time per 100 000 evaluations. The mean evaluation counters for the cases in Group C were much higher than those in Group O. This means that the mean calculation time for each run in the experiments with SA varied between 14 and 175 minutes. So in Group C it took about 2.5-29 hours to complete one experiment of 10 runs. Therefore, the remarks about impractical long runs are even more relevant in Group C which represents the more complex, and realistic scheduling situation where a stream of output benches has to be integrated with the stream of input benches.

Table 7-6. Calculation time in centi-minutes with Simulated Annealing for the real-scale test cases using temperature 120 and decay factor 0.999.

Time	Case:	C1	C2	C3	C4
	string length:	175	163	182	226
per 100 000 evaluations		302	156	281	411
to complete one run with mean number of evaluations		11 475	1 393	17 478	13 004

7.4 Results Group M: more than one output bench per crop batch

Large calculation times for the previous groups of test cases made it necessary to look for an alternative way to formulate and code the internal transport sequencing problem. This resulted in redefining the internal transport sequencing problem. Crop batches with only one output bench were clustered into larger crop batches that contained more than one output bench. One crop batch number coded on a string with a transport sequence now represents several output benches and replaces separate numbers of individual output benches. When a crop batch number is ready for transport - according to the transport sequence - all output benches with that specific crop batch number are collected from the growing area one after the other. The sub-sequence of the output benches within a specific crop batch could be determined according to all sorts of rules. In the experiments the transportable benches of one crop batch were collected starting at row number one and starting at the position closest to the entrance of a row. All required transport movements with a specific crop batch have to be finished completely before transport movements with the next crop batch number can be started. This approach leads to shorter strings because each string contains a transport sequence of crop batch numbers rather than all the individual output bench numbers. Less decision variables - crop batch numbers - involve the same number of output benches. The advantage of a shorter string is a shorter calculation time, which makes the approach more suitable for practical sized planning situations in pot plant nurseries. Another practical advantage of the approach is that output benches of a particular crop batch remain together because they are transported as a whole. The fact that all transportable benches of a specific crop batch arrive together in the input buffer of a work station makes it easier to perform operations on that specific crop batch without interruptions from other crop batches. In fact, a situation with more output benches per crop batch is a more realistic approach because this occurs more frequently in practice than the situation where there is only one output bench per crop batch.

For the Cases M3 and M4 the mean evaluation counter applies to the number of evaluations needed to find the transport sequence with value TM. However, for Case M1, the value ‘TM+2’ was taken to determine the mean evaluation counter. Case M1 was apparently more difficult than the other two cases, and transport sequences with TM were hardly ever found. Therefore, a statistical analysis of the mean number of evaluations needed to find a transport sequence with TM was useless and replaced by a statistical analysis of the mean number of evaluations needed to find a transport sequence with ‘TM+2’.

7.4.1 Local search method

In this Group M with more than one output bench per crop batch again three local search methods, SA (temperature 120 and decay factor 0.999), RS and GA-CX were compared at Population Size 20 (Appendix 7.2, Table 6). SA was the only local search method that always found the transport sequence with TM in all 10 runs. RS only found 10 transport sequences with TM for one case. In two cases SA was more effective and found a significantly lower mean minimum value than RS and GA-CX. In one case all three local search methods were effective and, therefore, the mean minimum values of SA, RS and GA-CX were all equal and had a standard deviation of zero, which made it impossible to perform the proposed statistical analysis. RS found significantly better results than GA-CX in one case. The highest worst minimum value was always found by GA-CX, indicating a higher probability of finding bad solutions with this method. These results are in accordance with the results in the other two groups of test cases with only one output or input bench per crop batch (Section 7.2 and 7.3).

The efficiency of SA was better in two cases where the mean evaluation counter of SA was significantly lower than the mean evaluation counters of RS and GA-CX. In those two cases no significant difference was found between the mean evaluation counters of RS and GA-CX. However, in the third case RS was significantly more efficient than SA. These deviating results were caused by a population size which was too high for SA in this less complex case. The experiments with SA at Population Size 2 (Section 7.4.2) showed that SA was significantly more efficient at that population size than RS at a population size of either 20, 5 or 2. The lowest best evaluation counter of a case was found by RS twice and once by GA-CX, indicating that these local search methods were able to find a good solution more rapidly than SA at Population Size 20 in some of the runs. The fact that the lowest best evaluation counter was never found by SA was caused by the decay factor which deliberately induced a relatively slow decrease of the number of transport movements. However, SA always found the lowest worst evaluation counter, indicating a smaller probability of finding bad solutions in the long run.

Figure 7-5 shows an example of the development of the mean number of transport movements. Figure 7-5 resembles the results in Figure 7-1 and 7-3. SA starts off at a higher mean number of transport movements than the other two local search methods. SA descends slower than RS and GA-CX, but SA is able to find a lower mean number of transport movements in the long run. After approximately 100 000 evaluations, the line of SA drops beneath the other two lines, still improving the mean number of transport movements. Finally, after approximately 200 000 evaluations SA stabilizes at a lower mean minimum value (169.0) than RS (171.0) and GA-CX (175.8). The local search methods RS and GA-CX stabilize much earlier at approximately 100 000 evaluations, and do not find any improvement after this point. That probably means that they were trapped in a local optimum from which they could no longer escape. In general, GA had serious problems finding good transport sequences for these more complex real-scale test cases. GA only found a transport sequence with TM in 8 of the 10 runs for the least complex case, where as the other two local search methods SA and RS were successful in doing so (Appendix 7.2, Table 6).

7.4.2 Population size

The local search methods SA and RS are implemented in SUGAL as a parallel algorithm, where each separate string in the population develops individually towards a better solution. These separate strings do not exchange information. It was assumed that the results of a large number of parallel strings developing with a certain parameter setting of SA or RS are normally distributed with a mean number of evaluations needed to find an internal transport sequence with TM and a standard deviation around this mean number of evaluations. A run in an experiment with, for example, a double population size also involves a double number of parallel strings and therefore such a run has a double chance that one of the strings finds a transport sequence with TM more rapidly than the mean required number of evaluations. The result of the fastest string in the run determines at what generation number a transport sequence with TM was found. However, what matters is the evaluation counter of the whole population because all other strings have gone through exactly the same number of generations. The total evaluation counter will be equal to (N-1) times the generation number of the best string (Section 7.1.1). Therefore the advantage of the lower number of evaluations for the single string is completely compensated by all the extra evaluations that are needed for the other strings that had to be run parallel to the best string and that had not yet found a transport sequence with TM.

A. Simulated Annealing

The population size was varied for SA at a constant temperature 120, and a constant decay factor 0.999, to study the effect on the efficiency of SA (Appendix 7.2, Table 7). The results showed that SA needed a certain number of evaluations to ‘cool down’ and find a transport sequence with TM. The best option for SA would be to have less parallel strings - a smaller population size - within a given limited number of allowed evaluations and thus more generations per string and sufficient time for SA to cool down. All experiments with SA were performed with a total number of allowed evaluations, sufficient to find a transport sequence with TM. So each run of the experiment was effective. The results show that the efficiency of SA was always influenced by the population size. In two cases the lowest mean

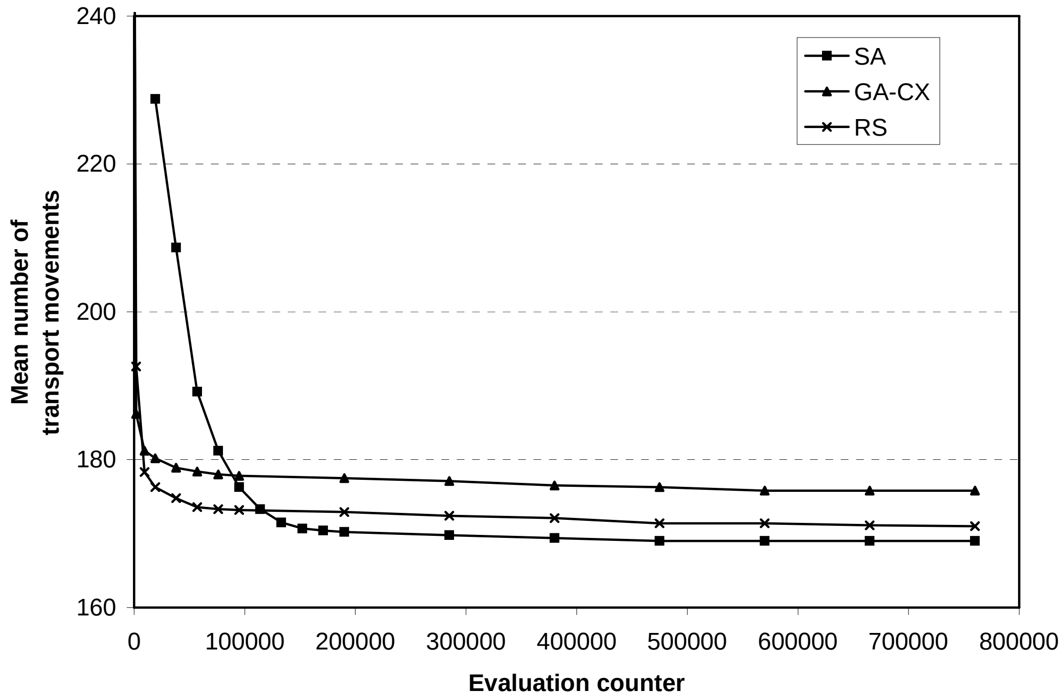


Figure 7-5. The development of the mean number of transport movements for the local search methods Simulated Annealing with temperature 120 and decay factor 0.999 (SA), Random Search (RS) and the Genetic Algorithm with cycle crossover (GA-CX) at Population Size 20 (Case M1).

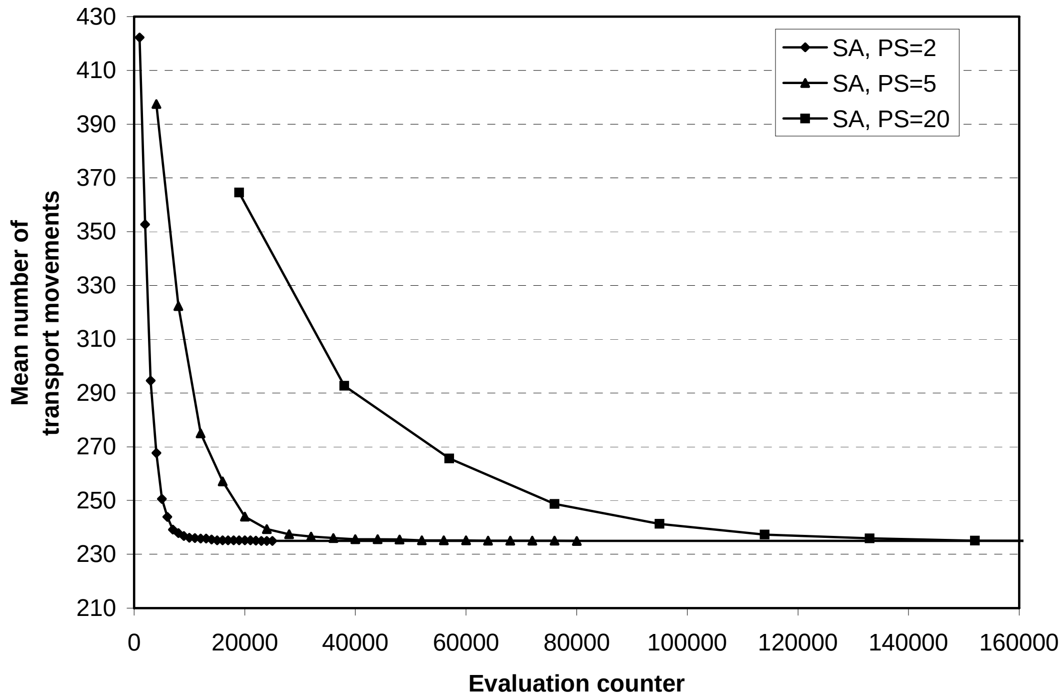


Figure 7-6. The development of the mean number of transport movements for three population sizes (PS) for Simulated Annealing with temperature 120 and decay factor 0.999 (Case M4).

evaluation counter of the experiments was found at Population Size 2, which differed significantly from Population Size 5, which in turn differed significantly from Population Size 20 (Figure 7-6). In one case the experiments with Population Size 5 found the lowest mean evaluation counter, which differed significantly from Population Size 2 and 20. These results are in accordance with the results of the other two groups of test cases. So SA in combination with Population Size 2 was often more efficient than SA in combination with Population Size 5 or 20.

B. Random Search

The population size was varied for RS to study the effect on the effectiveness and the efficiency of this local search method (Appendix 7.2, Table 8). When RS was used, the minimum value of a run decreased very rapidly after which no further improvement was found. This was because the algorithm was caught in a local optimum. All evaluations after this point were useless. Therefore, the best option for RS would be to have more parallel strings - a larger population size - within a given limited number of allowed evaluations and thus less generations per string. This is equivalent to many re-starts of the RS algorithm with one string. In the experiments with RS the mean minimum value of 10 runs was significantly lower for Population Size 20 than for Population Size 5 in two cases and Population Size 20 was significantly lower than Population Size 2 in one case. So population Size 20 was more effective in those cases. In one case no differences could be found between the mean minimum values at any of the different population sizes. The experiments of RS with Population Size 20 were significantly more efficient than those with Population Size 5 and 2 in only one case. Population Size 5 was significantly more efficient than Population Size 2 in only one case. Figure 7-7 shows that Population Size 20 gave slightly better results than Population Size 5 and 2. However, the lines are close together and the maximum distance between the final mean minimum values for Population Size 20 and 5 was only 0.5 transport movements.

C. Genetic Algorithm with cycle crossover

The GA is a local search method with an explicit exchange mechanism between the strings in the population, the crossover operator. A minimum Population Size 2 is needed to facilitate information exchange between individual solutions in a population. However, a Population Size 2 would mean a GA with very little variation in the population, thus limiting the role of the crossover operator and mainly relying on the mutation operator. Therefore, instead of choosing a very low Population Size 2, experiments with half of the original population size (Population Size 10) and twice the original population size (Population Size 40) were performed for GA-CX (Appendix 7.2, Table 9). A significant difference was only found in one case where

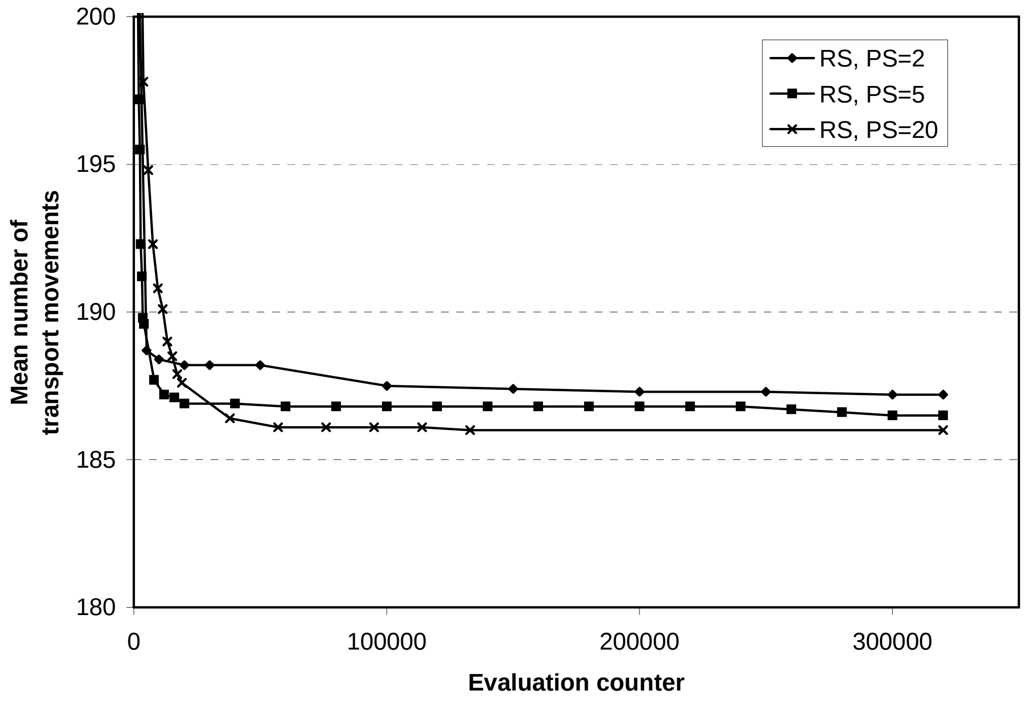


Figure 7-7. The development of the mean number of transport movements for three population sizes (PS) for Random Search (RS; Case M3).

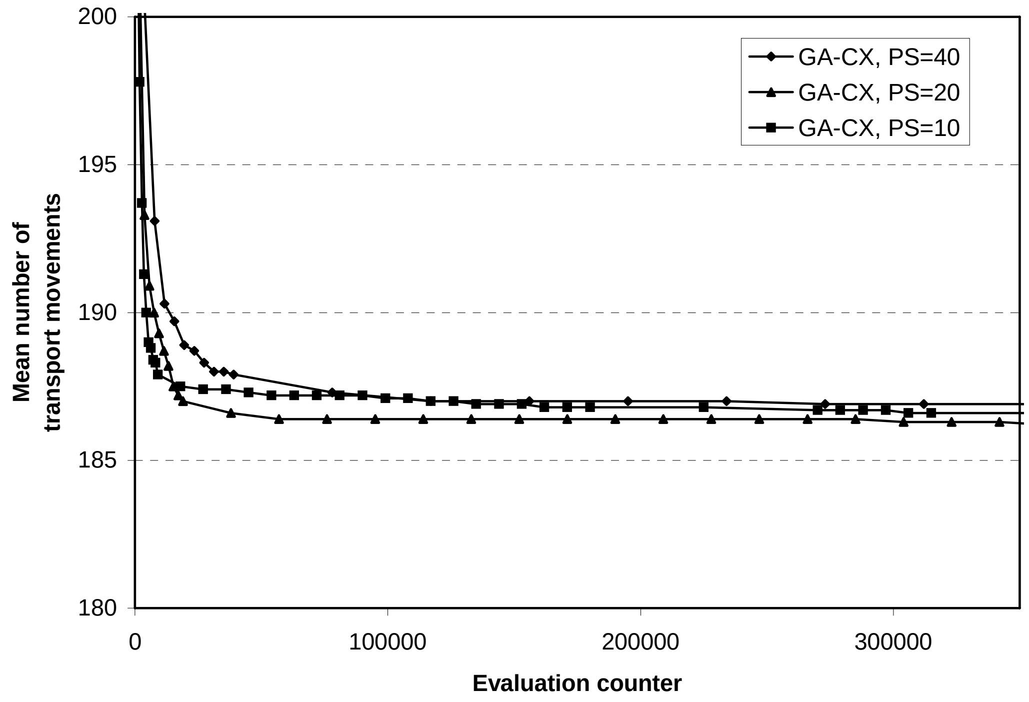


Figure 7-8. The development of the mean number of transport movements for three population sizes (PS) for the Genetic Algorithm with cycle crossover (GA-CX; Case M3).

Population Size 20 gave a lower mean minimum value than Population Size 40. No other case showed any significant difference with respect to either the effectiveness or efficiency of the population size. The development of the mean minimum value of transport movements in Figure 7-8 also shows little or no difference between the population sizes.

7.4.3 Cooling schedule of SA: decay factor and temperature

The most important control parameters of SA are specified in the cooling schedule (Section 6.3). In SUGAL these are implemented as a decay factor and an initial temperature (Section 6.7). The decay factor specifies at which rate the initial temperature should be decreased during a run of the algorithm. The temperature is needed to calculate the acceptance probability of the replacement of a better parent solution by an inferior child. Choosing the value of the decay factor quite near to 1, will give a slower, less efficient search process. However, in the long run it might be more effective when it finds a transport sequence with TM more often than with a smaller decay factor. However, when the number of allowed generations is too low for the search process to cool down, this will lead to inferior solutions. The specified temperature determines the initial value of the temperature control parameter.

A. Decay factor (df)

Four different decay factors were implemented for SA (ranging from 0.95 to 0.9995) in combination with a constant temperature 120 at Population Size 2 (Appendix 7.2, Table 10). Two cases showed a significant difference between decay factors. In general, a lower decay factor gave a significantly more efficient search process, with a lower mean evaluation counter. However, some decay factors in the range were too close to each other to find any significant differences (0.95-0.995 and 0.995-0.999). The difference in efficiency was to be expected because the search process should cool down more rapidly when a lower decay factor is chosen. However, an important result was that the high frequency of transport sequences with TM was maintained at these lower decay factors. It was still possible to find a transport sequence with TM in almost all the runs of the experiment. The difference in efficiency is also very apparent in Figure 7-9.

B. Temperature (T)

Three initial temperatures 60, 120 and 180 were compared for SA with a constant decay factor 0.999 at Population Size 2 (Appendix 7.2, Table 11). In contrast with the results for the decay factor, no significant differences were found at all concerning the efficiency of SA with these different temperatures. This is also shown in Figure 7-10 where the lines of the three temperatures run close together.

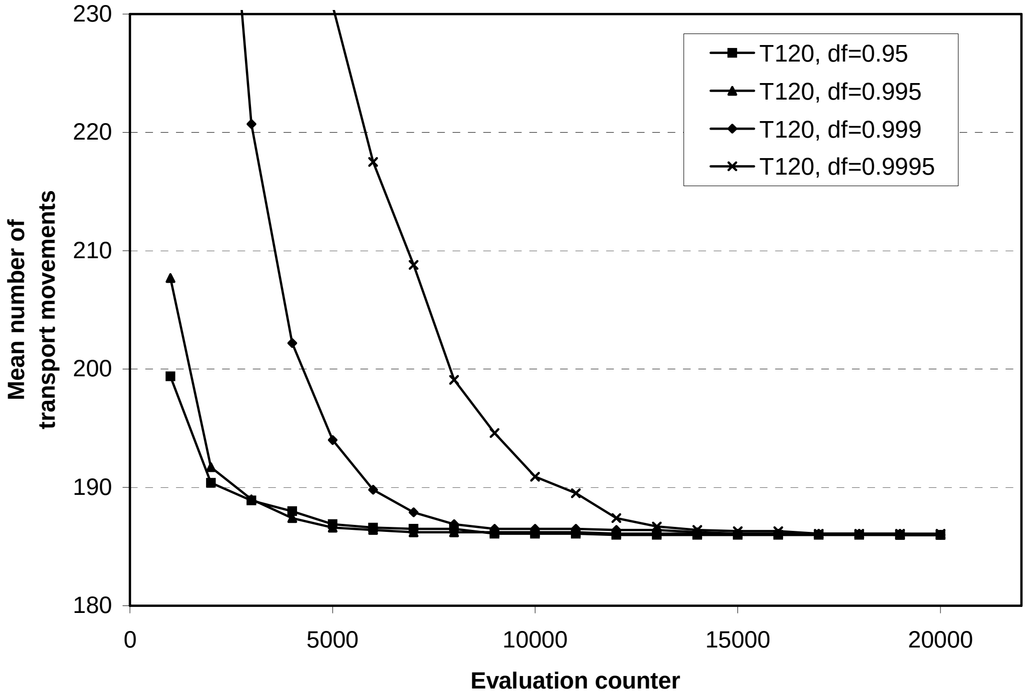


Figure 7-9. The development of the mean number of transport movements for four decay factors (df) for Simulated Annealing (temperature 120) at Population Size 2 (Case M3).

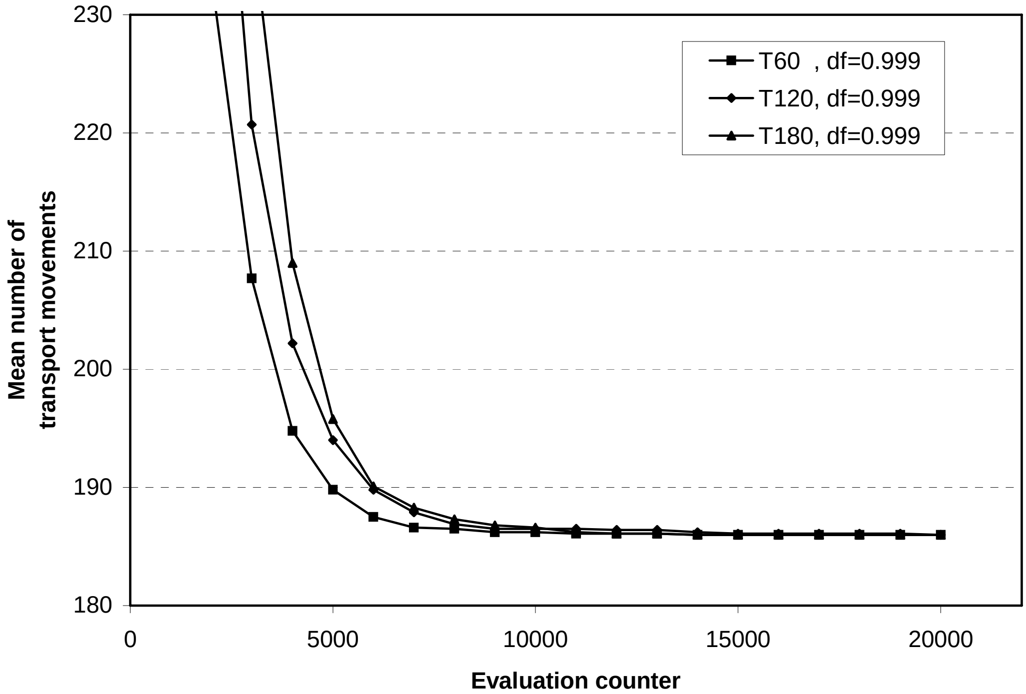


Figure 7-10. The development of the mean number of transport movements for three initial temperatures (T) for Simulated Annealing with a decay factor 0.999 (Case M3).

7.4.4 Tabu Search-swap combined with SA and RS

Tabu Search was not implemented as a separate local search method, but in combination with the two other local search methods SA and RS (Section 6.7). Fox (1993) also suggests an integration of TS, SA and GA. Glover (1989, 1990) recognized the possibilities for merging TS with other procedures, for example, SA. In his opinion TS can be integrated with methods containing optimality guarantees - such as SA in the long run - to improve their performance. Barnes & Laguna (1993) also considered the opportunity for future research to superimpose tabu search onto existing successful heuristics for specific scheduling problems. Glover et al. (1993) describe the method of performing many independent searches at a time with TS, each starting with a different initial solution or/and using a different set of parameters as a quite natural and less restricted parallelization process that works for each problem where it has been applied. The Tabu Search-swap (TS-swap) mutation operator was introduced to see if this alternative swap operator would improve the effectiveness and efficiency of the local search methods SA and RS, compared to the normal swap mutation operator. The assumption was that keeping a tabu list of the recently performed mutations on a string would create a mechanism to avoid a local optimum. This assumption had to be verified, just like the assumption that the TS-swap mutation operator would make the search process of SA and RS more efficient. The TS-swap mutation operator was not implemented in combination with GA-CX, because keeping a tabu list of the mutations of an individual string makes little sense, when parts of the strings are constantly being exchanged and recombined by the crossover operator. It would be too difficult to determine which part of a tabu list belongs to which part of the exchanged string in GA-CX.

A. Simulated Annealing and TS-swap

The normal swap mutation operator was compared with the TS-swap mutation operator for SA (temperature 120 and decay factor 0.999 at Population Size 2) and three different tabu list sizes (L=3, 7 and 11) were used (Appendix 7.2, Table 12). Unfortunately no judgement could be made about a possible increase in effectiveness of SA due to the TS-swap mutation operator, because the transport sequence with TM was already found in all runs of each experiment with the normal swap operator. Therefore, the analysis focussed on efficiency only. In one case TS-swap with a Tabu List size 3 or 11 was significantly more efficient than the normal swap mutation operator. However, no significant difference was found for the other two cases. The differences in the development of the mean number of transport movements for the case with the significant differences are shown in Figure 7-11. At 350 000 evaluations both mutation operators have reached a mean minimum number of transport movements that is lower than 170.

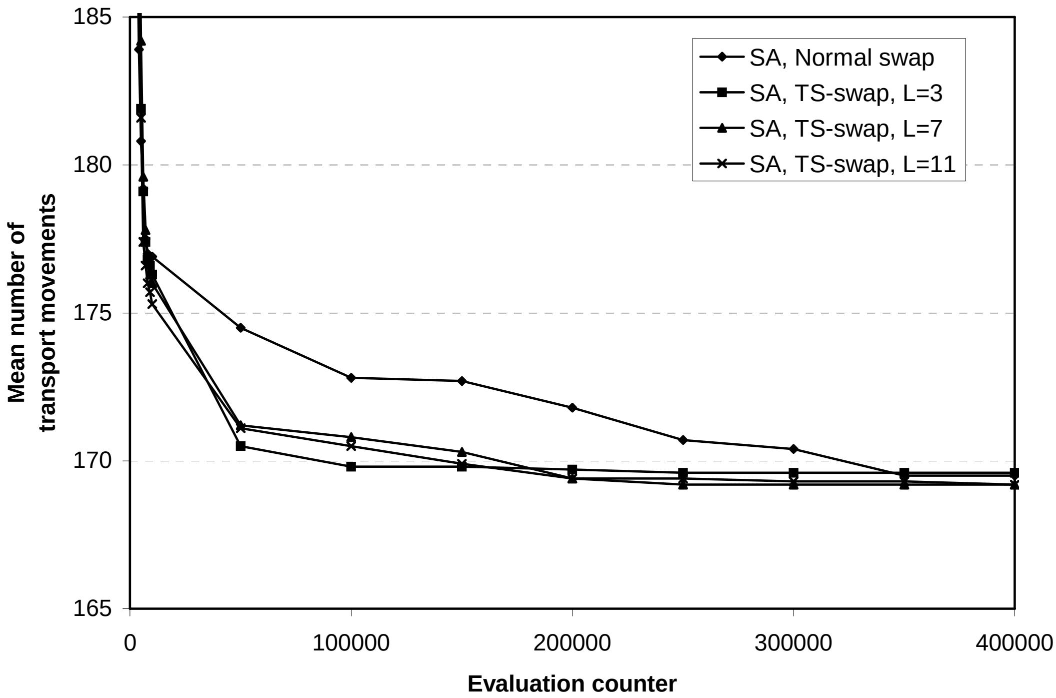


Figure 7-11. The development of the mean number of transport movements for the normal swap mutation operator and the Tabu Search-swap mutation operator (TS-swap) with three tabu list sizes (L) for Simulated Annealing (Case M1).

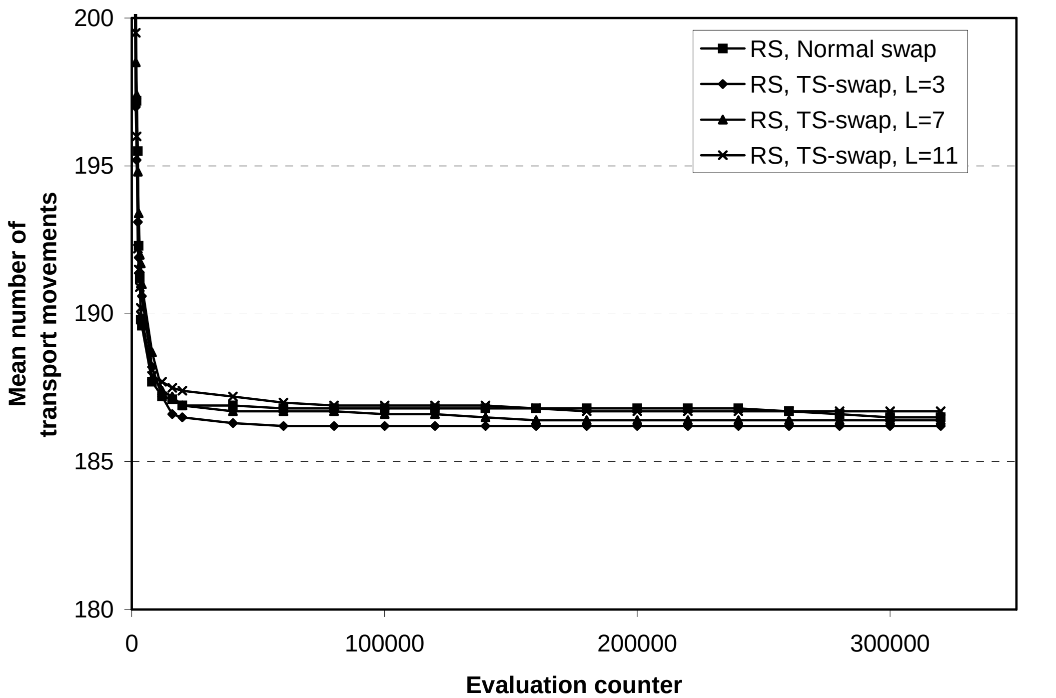


Figure 7-12. The development of the mean number of transport movements for the normal swap mutation operator and the Tabu Search-swap mutation operator (TS-swap) with three tabu list sizes (L) for Random Search (Case M3).

B. Random Search and TS-swap

The normal swap mutation operator was compared with the TS-swap mutation operator for RS at Population Size 5 and three different tabu list sizes (L=3, 7 and 11) were used (Appendix 7.2, Table 13). No significant differences were found between the two mutation methods with respect to both effectiveness and efficiency. The only exception was the significant difference in one case between a Tabu List size of 3 and 11, where Tabu List size 3 was better than Tabu List size 11. So applying the TS-swap mutation operator in combination with RS did not give any significant improvement in the results when compared to the normal swap mutation operator. One of the reasons for this could be that the implementation form of Tabu Search used was too simple to obtain significant results. Tabu Search might give better results if the tabu restrictions are combined with an aspiration criterion. Glover (1989) found that TS using the standard moves of the 2-OPT heuristic - which resembles RS -, and employing a simple tabu list and aspiration level structure for the travelling salesman problem easily found the known optimal solutions for each of the first three problems. However, an aspiration criterion was not implemented in the experiments with the transport sequences.

7.4.5 Crossover method

Three different crossover methods (Section 6.2) were tested in combination with GA (Appendix 7.2, Table 14): partially mapped crossover (PMX), order crossover (OX) and cycle crossover (CX). The effectiveness of CX was significantly better than the effectiveness of OX in two cases and significantly more effective than PMX in one case. PMX was significantly more effective than OX in one case. CX had the highest frequency of finding transport sequences with TM of the three crossover methods except for one case where none of the three methods found TM.

The efficiency of CX was significantly better than the other two crossover methods in one case. Figure 7-13 shows the difference between CX and the other two crossover methods. The lines of PMX and OX run close to each other at the end of the search, while the line of CX runs below them.

7.4.6 Calculation time

The calculation time of each run of these more complex real-scale test cases varied on a 350 MHz Pentium Pro(r) PC with 64.0 MB of RAM memory, depending on the length of the string and on the parameter settings of the chosen local search method. The calculation time was between 115 and 193 centi-minutes per 100 000 evaluations (Table 7-7). The mean evaluation counters varied between 38 000 and 760 000 (Appendix 7.2, Table 6) so the total calculation time to reach the mean evaluation counter varied between 4 and 1 079 centi-minutes. The combination of a lower calculation time and less evaluations needed enable practical applications with a total calculation time of less than 10 minutes.

Table 7-7. Calculation time in centi-minutes for the real-scale test cases using the local search methods Simulated Annealing (SA, temperature 120 and decay factor 0.999), Random Search (RS) and the Genetic Algorithm with cycle crossover (GA-CX) at Population Size 20.

Local search method	Time	M1	M3	M4
	string length:	26	35	43
SA	per 100 000 evaluations	137	145	193
	to complete one run with mean number of evaluations	230	164	270
RS	per 100 000 evaluations	116	115	155
	to complete one run with mean number of evaluations	>704	4	>491
GA-CX	per 100 000 evaluations	142	131	177
	to complete one run with mean number of evaluations	>1 079	>162	>616

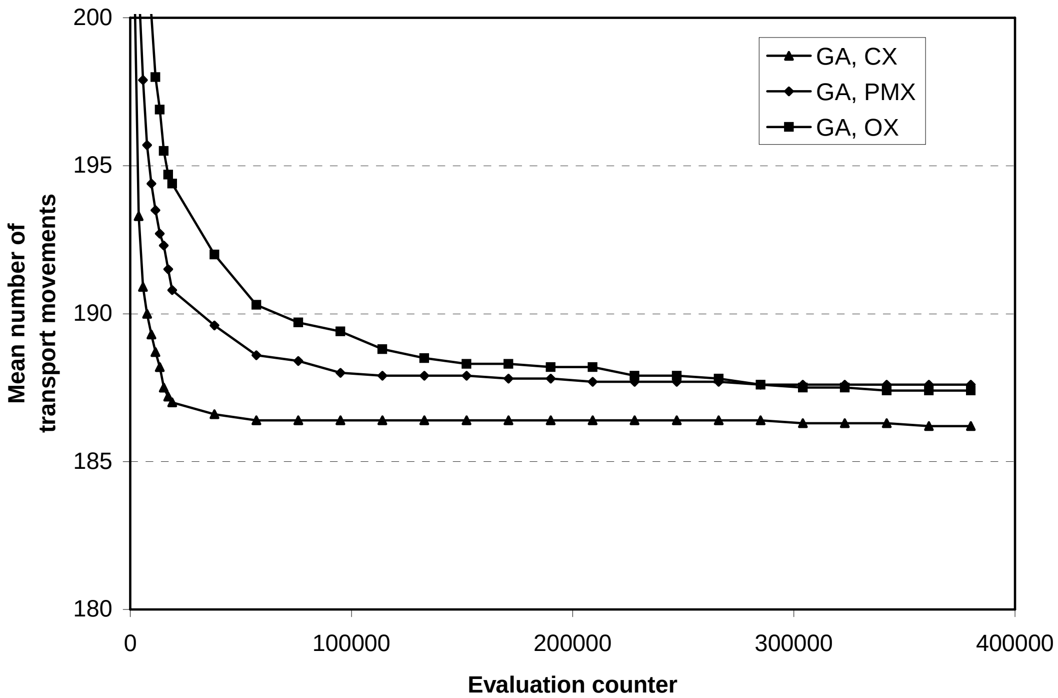


Figure 7-13. The development of the mean number of transport movements for three crossover methods: cycle crossover (CX), partially mapped crossover (PMX) and order crossover (OX) for the genetic algorithm (GA; Case M3).

7.5 Conclusions

The experiments showed that it is possible to obtain transport sequences with the theoretical minimum number of transport movements (TM) in practice for all three groups of the real-scale test cases (Section 7.2-7.4). However, it is more difficult to obtain transport sequences with TM for the real-scale test cases using the local methods Simulated Annealing (SA), Random Search (RS) and the Genetic Algorithm with cycle crossover (GA-CX), than it was for the small-scale, simple test cases (Chapter 6).

The results of comparing the three local search methods and the results of comparing the Population Size of SA, were obtained for all three groups of test cases (Group O, C and M). All other results were only obtained for test case Group M. This represented the situation with more than one output bench per crop batch.

For all three groups of test cases, SA was a significantly more effective local search method at Population Size 20 than RS and GA-CX. SA was the only local search method that found a transport sequence with TM in all ten runs of an experiment in almost all cases. RS in turn was more effective than GA-CX in half the cases: two cases out of four in Group O, one out of one in Group C and one out of three in Group M. The efficiency of the local search methods could only be analysed for the third group of test cases. SA was significantly more efficient than RS and GA-CX in two out of three cases. In the other case RS was significantly more efficient than SA at Population Size 20, but was no longer so at Population Size 5 and 2.

Population size influenced the efficiency of SA in all three groups of test cases. Population Size 2 was significantly more efficient than Population Size 20. Therefore, it can be concluded that it is better to have fewer parallel runs of SA with a larger number of allowed evaluations. The results for RS were the reverse of SA and showed that Population Size 20 is significantly more effective and efficient than Population Size 2. These results indicate that it is better to have many re-starts with a low number of allowed evaluations for RS. The population size hardly influenced the effectiveness or the efficiency of the GA-CX. There was one case in which Population Size 20 was more effective than Population Size 40.

The parameters of the cooling schedule partly influenced the efficiency of SA. The choice of the first parameter, the decay factor, had a significant impact. At lower decay factors - from 0.9995 to 0.95 - the efficiency of SA increased significantly, and SA was also able to maintain the high frequency of strings with TM (all 10 runs in an experiment). The other parameter, the initial temperature, did not influence the efficiency of SA significantly.

The Tabu Search-swap mutation operator, an alternative for the normal swap operator, did improve the efficiency of SA significantly in the most complex case in a combination with Tabu List length 3 and 11. The TS-swap mutation operator did not improve the effectiveness or the efficiency of RS.

In the case of GA the cycle crossover method (CX) gave significantly better results than the order crossover method (OX) in two cases and the partially mapped crossover method (PMX) in one case. The partially mapped crossover method in turn, was significantly better than the order crossover method in one case.

The conclusions that were given in the previous sections were based on the experiments with the described groups of test cases. It will take more experiments to prove that these conclusions always hold for quite different cases in other sorts of different pot plant nurseries. However, the fact that SA consistently showed better results with regard to both effectiveness and efficiency for all test cases (both simple and real-scale), indicates that this local search method has an excellent potential for real-life application. It is noteworthy that in the most difficult group of test cases where the stream of output and input benches were combined (Group C), SA clearly outperformed RS and GA-CX. The experiments showed that the efficiency of SA can be fine-tuned by choosing an appropriate population size and decay factor. The TS-swap mutation operator showed some significant differences in the experiments, but further research will be needed to find an appropriate way to apply this mutation operator.

No clear evidence was found in the experiments to show that the more sophisticated local search method GA achieved better results than the simple local search method RS in any of the groups of test cases. Therefore, it can be concluded that applying the GA to find a good transport sequence is a less suitable approach than applying RS.

The use of these local search methods to find transport sequences in actual pot plant nurseries needs further research. Several issues still have to be resolved. One such issue involves looking at other ways of evaluating the transport sequence, for example, by taking into account the distance the AGV travels. Another very important issue is the interaction between the current transport sequence and parking positions and transport sequences and parking positions in the near future. An optimal current transport sequence combined with a certain parking method may lead to sub-optimal transport sequences in the near future, because the current decisions on the parking positions of transportable benches may lead to the obstruction of the output benches of the next sequencing problems.

8. Final conclusions and recommendations

8.1 Final conclusions

The results of the simulation experiments show that the rules of thumb which growers use to control day-to-day internal transport constitute a valuable basis for developing methods to support the operational planning process. The rules of thumb can be used to incorporate company-specific knowledge in a control system in order to make such a control system better adapted to company-specific internal transport problems.

The performance results of the local search methods applied to real-scale test cases show that Simulated Annealing is the most effective and efficient method of finding good internal transport sequences in combination with a relatively simple parking method. Transport sequences with a theoretical minimum number of transport movements were frequently obtained by Simulated Annealing for these real-scale test cases. However, the other local search methods Genetic Algorithm, Tabu Search and Random Search also produced relatively good transport sequences in many calculation runs. In practice, the internal transport control problems will vary between different pot plant nurseries and therefore, the best approach will be to use a combination of these local search methods in order to be prepared for all sorts of internal transport control situations.

The performance results of the local search methods also indicate that the approach of combining a sophisticated transport sequence generating technique with a relatively simple parking method is a successful one. This approach has the advantage that other pot plant nurseries will only have to specify one or more simple company-specific parking methods based on their rules of thumb and that the generic transport sequence generating technique based on the local search methods can remain the same. This will enable tailor-made solution methods for other pot plant nurseries with a very low additional development effort.

Now that the described control approach has been proven successful, a completely automatic support system for internal transport in pot plant nurseries lies within reach. However, this still requires the implementation of some further extensions described in Section 8.2. In a final application of the control approach transport sequences can be generated automatically and transferred automatically to the AGV. Deviations in an executed transport sequence or the chosen parking positions can also be reported and dealt with automatically. Of course, the grower will need to have the opportunity at all times to intervene in the internal transport control process. However, he or she will no longer have to bother with taking frequent, standard transport decisions. These decisions will be suggested by the control system. The grower will only have to judge special situations that cannot be foreseen by the control system. If these situations occur more frequently they can be incorporated in new parking methods. In this way the grower will save directly on labour for the internal transport control activity.

It is very difficult to make an exact estimate of the financial revenues of an improved internal transport control approach. Decisions on the operational control level are made within the financial constraints of the tactical and strategic planning level of a specific pot plant nursery. At these two levels it is possible to distinguish between a production plan with a higher or a lower gross margin or between a design with higher or lower investment costs. However, at the operational control level the only direct costs made are the variable costs of internal transport sequences. These variable costs are proportional to the number of transport movements. The indirect revenues of an improved internal transport control approach lie in the fact that an effective and efficient internal transport process facilitates a flexible production process that is needed to satisfy all customer demands.

The control approach described is able to generate good transport sequences with a low number of transport movements, which has several advantages. One advantage is that the grower will be able to respond more rapidly to the additional demands of customers during a particular day. Now he or she will be able to collect transportable benches with pot plants required for harvesting with less movements and thus be able to supply customers more quickly. Another advantage is that the AGV will be less busy, so that it is less likely that transport movements on any particular day will be delayed. Furthermore, when fewer transport movements are needed to perform a production plan, it may be possible to design a pot plant nursery with less Automatic Guided Vehicles or other transport devices. Thus lower investment costs can be achieved.

8.2 Recommendations

Components of internal transport

The research described was focused on the transportable bench production system for reasons mentioned in Section 2.3.7. However, internal transport for the concrete floor production system cannot be neglected even if the transport control problem is of a different nature and complexity. Therefore, it is recommended that research on the control of internal transport should also be performed for this other important type of production system. Research on the concrete floor production system should take into account different types of layout and different internal transport devices. In general, the operations on pot plants will remain the same for the concrete floor production system as for the transportable bench production system, with the exception that pot plants have to be picked up from the concrete floor before they can be handled. However, the type of access to individual pot plants of a crop batch will be different, because each crop batch stands on the floor as one large group. This makes it difficult to reach individual pot plants at the back of a concrete floor.

Hierarchical planning approach

The current study mainly looked at the operational planning level. However, the interaction between the different planning levels should be analysed in more detail in order to establish a better integration of design, planning and control activities. In the described approach the internal transport problem at the operational planning level is solved within the constraints of the decisions that are made at other planning levels. The constraints of the other planning levels are only adjusted if they lead to an infeasible solution at the current planning level. However, research on some kind of simultaneous iterative approach is needed to achieve an overall optimization. Such a simultaneous iterative approach could be beneficial for a better understanding of the influence of the total planning process on the operational control level. However, such an approach depends heavily on rapid information exchange between the planning levels. Therefore, new techniques in the field of Information and Communication Technology (ICT) like data mining and neural networks will become even more important for the operational planning level in the near future than they are at present. Accurate control of the internal transport process on a minute-to-minute basis will only be possible if the automatically recorded data of the AGV, the work stations and the status of the growing compartments can be combined with data bases that include product requirements, sales predictions and the availability of resources.

TRANSIM simulation model

The current version of TRANSIM could be successfully implemented in actual day-to-day practice at pot plant nurseries to support the internal transport control process. In order to achieve this, some extensions should be made to further improve the simulation model. These extensions focus on:

- capability of handling stochastic data;
- flexible data entry for inexperienced users;
- handling other layout types and forms of the growing compartment;
- integration between TRANSIM and the AGV.

The current TRANSIM simulation model operates with deterministic data. However, in practice most data are of a stochastic nature. Certain data will be known at the beginning of the day - for example the number of plants that have to be potted on that particular day - while other data will only become (more) definite during the day, for example, the exact number of pot plants that have been sold and thus have to be transported. Other examples of stochastic data include the moment of spacing a crop batch, the moment when young plants arrive as input for the potting operation and the length of each growing phase. Therefore, an extended version of TRANSIM should be capable of taking into account the stochastic nature of the data during the simulation run.

In the current version of the TRANSIM simulation model some of the input parameters such as the processing time of different operations and the transport time of the AGV are specified within the modules of the model, where they can be changed by an experienced user. However, this should be modified to a more flexible input facility either in the opening screen of TRANSIM or straight from the data base of the pot plant nursery, so that less experienced growers can also run the simulation model easily.

Furthermore, the TRANSIM simulation model should be extended so that it can also deal with types of layout other than LIFO such as FIFO and RIRO access. An extension also applies to other forms of the growing compartment. Recently Halachmi (1999) performed an analysis in which the growing compartment was designed with a circular form. She compared a rectangular greenhouse with a circular greenhouse with regard to the time needed to harvest cucumbers, the number of harvesting robots required, the robot utilization and the cart utilization. Although her research focused more on the strategic planning level the work is a clear signal that growing compartments with another form will become of increasing importance in the near future. With some extensions to the TRANSIM simulation model it should be possible to perform this type of research at the operational control level.

In TRANSIM's implementation phase, the output of a simulation run should be linked to an automatic internal transport control system. In this way, scheduling solutions found by the simulation model could be transferred automatically to the control system of the Automatic Guided Vehicle and to the work stations. The manufacturers of automatic recording and internal transport systems need to be involved in order to establish an appropriate integration between the simulation model and an internal transport control system. A common information model - or class model in the case of an object oriented approach - is an absolute requirement for the integration of modules from different manufacturers. In this way standard interfaces can be developed.

Local search methods

The described internal transport control approach of using local search methods to determine the transport sequence in combination with a certain parking method also needs some additions and improvements before it can be implemented in practice. The behaviour of the different local search methods still has to be studied for an internal transport control situation that is less simplified than it was in the experiments in this study. The most important simplifications in this study were that the parked obstructing transportable benches (both output and neutral) were not transported back to their original row; the AGV can only transport one transportable bench at a time; the planning problem of the next period is not taken into account while determining the parking rows, and, that no problems were studied in which the transport sequence was already partly fixed before the sequencing process began. Furthermore, the approach should also be tested for other types of layout besides LIFO such as FIFO and RIRO access.

One of the most important issues that remains to be solved is how to take into account the constraints of future control problems when solving the current control problem. In this study generated transport sequences were only evaluated on the number of transport movements during the period of the current control problem. However, a transport sequence for the current control problem with a low number of transport movements, might induce a higher number of transport movements than necessary for the next control problem. So one improvement to the described approach should be that the chosen parking method for the current control problem can take into account the constraints of control problems that might be expected in the near future. This requires additional research to develop more intelligent parking methods that determine parking rows based on both current and future control problems. However, in this recommended research the balance between the effect of a sophisticated local search method to generate a transport sequence and an intelligent parking method to determine parking positions (Chapter 5) still remains an important issue, because putting too much effort into developing an intelligent parking method might interfere again with the solving capacity of intelligent transport sequence generating techniques.

9. References

Aarts, E.H.L. & J. Korst, 1989
Simulated annealing and Bolzmann machines. John Wiley & Sons, Chichester, UK, 272 pp.

Aarts, E.H.L. & P.J.M. van Laarhoven, 1989
Simulated annealing: An introduction. Statistica Neerlandica, 43, 1, 31-52.

Aarts, E.H.L., P.J.M. van Laarhoven, J.K. Lenstra & N.J.L. Ulder, 1994a
A computational study of local search algorithms for job shop scheduling. ORSA Journal on Computing, 6, 2, 118-125.

Aarts, E.H.L., J.K. Lenstra & R.J.M. Vaessens, 1994b
Local Search for Job Shop Scheduling. In: Maaren, H. van & P. van Putten (eds.), Abstracts van het Congres Operations Research & Artificial Intelligence, 10 januari, Utrecht, The Netherlands, 12-20.

Aarts, E.H.L. & J.K. Lenstra, 1997
Local search in combinatorial optimization. John Wiley & Sons, Chichester, UK, 512 pp.

Aarts, E.H.L., J.H.M. Korst & P.J.M. Laarhoven, 1997
Simulated annealing. In: Aarts, E.H.L. & J.K. Lenstra (eds.), Local search in combinatorial optimization, John Wiley & Sons, Chichester, UK, 91-120.

Aldrich, R.A. & J.W. Bartok, 1990
Greenhouse engineering. Northeast regional Agricultural Engineering Service, NREAS-33, Cornel University, Ithaca, New York, USA, 203 pp.

Ammerlaan, J.C.J. , 1984
Development in management information systems in the glasshouse horticulture in The Netherlands. Acta Horticulturae, 155, 235-241.

Anderson, E.J., 1996
Mechanisms for local search. European Journal of Operational Research, 88, 139-151.

Annevelink, E., 1989
The IMAG Production Planning system (IPP) for glass-house floriculture in its introduction phase. Acta Horticulturae, 237, 37-45.

Annevelink, E. , 1992,
Operational Planning in Horticulture: Optimal Space Allocation in Pot plant Nurseries Using Heuristic Techniques. Journal of Agricultural Engineering Research, 51, 167-177.

Annevelink, E., 1993.
Experimental results of applying the Genetic Algorithm to space allocation planning in pot-plant nurseries. In: Annevelink, E., R.K. Oving & H.W. Vos (eds.), Proceedings XXV CIOSTA-CIGR V Congress, Wageningen Pers, Wageningen, The Netherlands, 89-98.

Annevelink, E. & R.A.C.M. Broekmeulen, 1993.
The Genetic Algorithm applied to space allocation planning in pot plant nurseries. Acta Horticulturae, 340, 141-148.

Annevelink, E. & A. Vink, 1994.
A simulation model for the operational planning of internal transport in greenhouses. CIGR/AgEng ‘94 Report, N. 94-D-001, Milano, 12 pp.

Annevelink, E. & H.J.L. van der Voort, 1995
Beschrijving van het intern transport bij potplantenbedrijven met rolcontainers (Description of the internal transport systems in pot plant nurseries with rolling containers). IMAG-nota, 95-10, Instituut voor Milieu- en Agritechniek, Wageningen, The Netherlands, 47 pp.
Annevelink, E. & H.J.L. van der Voort, 1996
A detailed description of the internal transport systems in pot plant nurseries with rolling containers. Acta Horticulturae, 429, 59-66.
ASAE, 1993
Commercial greenhouse design and layout. In Engineering Standards ASAE, St Joseph, MI 49085 - 9659, USA, EP460, 598-604.
Bagchi, S., S. Uckun, Y. Miyabe & K. Kawamura, 1991
Exploring problem specific recombination operators for job shop scheduling. In: Belew, R. & L. Booker (eds.), Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, Los Altos, CA, USA, 10-17.
Ball, V., 1988
Totally automated pot plants in sight! GrowerTalks, 52, 5, 68-82.
Barnes, J.W. & M. Laguna,1993
A Tabu Search experience in production scheduling. Annals of Operations Research, 41, 141-156.
Bartok, J.W., 1984
Moveable greenhouse bench system design. ASAE Paper, No 84-1088, St Joseph, USA, 15 pp.
Basham, C.W. & J.J. Hanan , 1983
Space optimization in greenhouses with linear programming. Acta Horticulturae, 147, 45-56.
Bloch, N. , 1983
The application of production planning in practice. Acta Horticulturae, 147, 125-130.
Boot, J. & P. van Waarde, 1990
Mechanisatie van het intern transport in de potplantenteelt op betonvloeren (Mechanization of internal transport in pot plant cropping on concrete floors). IMAG-nota, P533, Instituut voor Milieu- en Agritechniek, Wageningen, The Netherlands, 64 pp (+ Appendices).
Buchwald, H.H. , 1987
A simulation model for planning and control of the potted plant production. Acta Horticulturae, 203, 39-49.
Buxey, G., 1989
Production scheduling: practice and theory. European Journal of Operational Research, 39, 17-31.
CBS, 1998
Statistisch Jaarboek 1998 (Statistical Yearbook 1998). Centraal Bureau voor de Statistiek, Voorburg/Heerlen, The Netherlands, 133-162.
Chen, C.-L., V.S. Vempati & N. Aljaber, 1995
An application of genetic algorithms for flow shop problems. European Journal of Operational Research, 80, 389-396.
Dambruin, N., 1989
Modelling and simulation of internal transport in greenhouses. MSc Report, Wageningen Agricultural University, Department of Agricultural Engineering, Wageningen, The Netherlands, 94 pp.
Davis, L., 1985
Job shop scheduling with Genetic Algorithms. In: Grefenstette, J.J. (ed.), Proceedings of the First International Conference on Genetic Algorithms, Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 136-140.

Davis, L., 1991
Handbook of genetic algorithms. Van Nostrand Reinhold, New York, USA, 385 pp.
Dell'Amico, M. & M. Trubian, 1993
Applying tabu search to the job-shop scheduling problem. Annals of Operations Research, 41, 231-252.
Dijkstra, J., 1994
Application of digital image processing for pot plant grading. PhD Thesis, Wageningen Agricultural University, Department of Agricultural Engineering and Physics, Wageningen, The Netherlands, 195 pp.
Dorn, J., 1995
Iterative improvement methods for knowledge-based scheduling. AICOM, 8, 1, 20-34.
Dorndorf, U. & E. Pesch, 1995
Evolution based learning in a job shop scheduling environment. Computers & Operations Research, 22, 1, 25-40.
Eglese, R.W., 1990
Simulated Annealing: A tool for Operations Research. European Journal of Operations Research, 47, 271-281.
Elderen, E. van & G.H. Kroeze, 1994
Operational decision making models for arable and grassland farms. IMAG-report, 94-3, Instituut voor Milieu- en Agritechniek, Wageningen, The Netherlands, 74 pp.
Elsner, B. von & C. Reuter, 1989
Model for the simulation of internal transport in pot plant nurseries. Acta Horticulturae, 248, 137-140.
Fang, W., 1989
Strategic planning through modeling of greenhouse production systems. Ph.D. dissertation, New Jersey Agricultural Experiment Station (NJAES), Publication No. T-03232-20-89, New Brunswick, USA, 244 pp.
Fang, W., K.C. Ting & G.A. Giacomelli, 1990a
Animated Simulation of Greenhouse Internal Transport Using SIMAN/CINEMA. Transactions of the ASAE, 33, 1, 336-340.
Fang, W., K.C. Ting & G.A. Giacomelli, 1990b
Optimizing resource allocation for greenhouse potted plant production. Transactions of the ASAE, 33, 4, 1377-1382.
Fang, W., K.C. Ting & G.A. Giacomelli, 1992a
Computer software development for greenhouse design and management. In: Watson, D.G., F.S. Zazueta & A.B. Bottcher (eds.), Computers in agricultural extension programs, Proceedings of the 4th International Conference, ASAE, St. Joseph, USA, 274-279.
Fang, W., G.A. Giacomelli & K.C. Ting, 1992b
Greenhouse layout design and labor planning. American Society of Agricultural Engineers, St. Joseph, USA, ASAE meeting presentation, paper no. 924071, 21 pp.
Filipic, B., 1993
Enhancing genetic search to schedule a production unit. In: Dorn, J. and K.A. Froeschl (eds.), Scheduling of production processes, Ellis Horwood, Chichester, UK, 61-69.
Fox, B.R. & M.B. McMahon, 1991
Genetic Operators for sequencing problem. In: Rawlings, G.J.E. (ed.), Foundations of Genetic Algorithms, Morgan Kaufmann, San Mateo, California, USA, 284-300.
Fox, B.L., 1993
Integrating and Accelerating Tabu Search, Simulated Annealing, and Genetic Algorithms. Annals of Operations Research, 41, 47-67.
Giacomelli, G.A., P. van Weel & M. van der Schilden, 1987
Expert systems development for greenhouse potted plant production. ASAE paper, No 87-5012, ASAE, St. Joseph, MI 49085, USA, 10 pp (+ Appendices).

Giacomelli, G.A., W. Fang, K.C. King & P. van Weel, 1991
Defining internal transport systems in greenhouses. Acta Horticulturae, 295, 233-241.
Glover, F., 1989
Tabu search - Part I. ORSA Journal on Computing, 1, 3, 190-206.
Glover, F., 1990
Tabu Search: A Tutorial. Interfaces, 20, 74-94.
Glover, F. & H.J. Greenberg, 1989
New approaches for heuristic search: A bilateral linkage with artificial intelligence. European Journal of Operations Research, 39, 119-130.
Glover, F., E. Taillard & D. de Werra, 1993
A user's guide to tabu search. Annals of Operations Research, 41, 1, 3-28.
Glover, F. & M. Laguna, 1997
Tabu Search. Kluwer Academic Publishers, Dordrecht, The Netherlands, 382 pp.
Goldberg, D.E., 1989
Genetic algorithms in Search, Optimization & Machine Learning. Addison-Wesley Publishing Company Inc., Michigan, USA, 412 pp.
Grefenstette, J.J., 1990
Genetic Algorithms and their applications. In: Kent, A. & J.G. Williams (eds.), The Encyclopedia of Computer Science and Technology, 21, 139-152.
Håkansson, B., 1983
Use of computers for economic control and planning on horticultural firms. Acta Horticulturae, 135, 97-104.
Håkansson, B., 1987
A computer based system for production planning and budgeting in horticultural firms. Acta Horticulturae, 203, 13-22.
Håkansson, B., 1991
A decision support system for production planning in horticulture. Acta Horticulturae, 295, 195-201.
Halachmi, S., 1999
Robotic harvesting time of rectangular and circular greenhouse. Internal report, Instituut voor Milieu en Agritechniek, Wageningen, The Netherlands, 16 pp.
Hamrick, D., 1988
The logistics of greenhouse efficiency: benches out, floor in? Grower talks, 52, 4, 126-143.
Hanan, J.J., W.D. Holley & K.L. Goldsberry, 1978
Greenhouse Management. Springer-Verlag Berlin, Berlin, BRD, 530 pp.
Hanan, J.J., 1998
Greenhouses; Advanced technology for protected horticulture. CRC Press, Boca Raton, Florida, USA, 684 pp.
Hax, A.C. & D. Candea, 1984
Production and inventory management. Prentice-Hall Inc., Englewood Cliffs, New Jersey, USA, 513 pp.
Heinrici, A., 1993
A comparison between simulated annealing and tabu search with an example from the production planning. In: DGOR/NSOR, Operations Research Proceedings, Springer-Verlag, Berlin, BRD, 498-503.
Hendriks, Th.H.B. & P. van Beek, 1991
Optimaliseringstechnieken, principes en toepassingen (Optimization techniques, principles and applications). Derde druk, Bohn Stafleu Van Loghum, Houten, The Netherlands, 348.
Hendrix, A.T.M., 1975
Transport systems in glasshouse horticulture. Netherlands Journal of Agricultural Science, 23, 231-237.

Hertz, A., 1991
Tabu search for large scale timetabling problems. European Journal of Operational Research, 54, 39-47.
Hertz, A., E. Taillard & D. de Werra, 1997
Tabu search. In: Aarts, E.H.L. & J.K. Lenstra (eds.), Local search in combinatorial optimization, John Wiley & Sons, Chichester, UK, 121-136.
Hoeven, J. van der, 1995
Rolcontainers of betonvloeren (Transportable benches or concrete floors)? Vakblad voor de Bloemisterij, 50, 6, 70-73.
Hofstede, G.J., 1992
Modesty in Modelling: on the Applicability of Interactive Planning Systems, with a case study in Pot Plant Cultivation. Doctoral Dissertation, Wageningen Agricultural University, Department of Computer Science, Wageningen, The Netherlands, 243 pp.
Hoorn, W. van der, 1995. Aanmaken van invoerbestanden voor het simulatiemodel intern transport (Constructing input files for the simulation model of internal transport). IMAG-nota, P 95-62, Instituut voor Milieu en Agritechniek, Wageningen, The Netherlands, 43 pp (+ Appendices).
Hoshmand, A.R., 1998
Statistical methods for environmental and agricultural sciences. CRC Press, Boca Raton, New York, USA, second edition, 439 pp.
Huijbers, J.H.A.M.,1996
Operationele planning van het intern transport op potplantenbedrijven: het clusteralgoritme (Operational planning of internal transport at pot plant nurseries: the cluster algorithm). IMAG-nota, P 96-45, Instituut voor Milieu- en Agritechniek, Wageningen, The Netherlands, 40 pp.
Hunter, A., 1995a
SUGAL User Manual, V2.1. University of Sunderland, UK, 76 pp.
Hunter, A., 1995b
SUGAL Programming Manual, V2.1. University of Sunderland, UK, 69 pp.
IKC, 1991
Opzet en inrichting van een potplantenbedrijf (Design and layout of a pot plant nursery). Informatie en Kennis Centrum Akker- en Tuinbouw, Aalsmeer, The Netherlands, 59 pp (+ Appendices).
Janssen, A.H.G., 1996
Van puzzel tot plan; Vervolgonderzoek naar de scheduling van intern transport glastuinbouwbedrijven (From puzzle to plan: Continuation of the scheduling research of internal transport at floricultural production companies). IMAG-nota, P 96-65, Instituut voor Milieu- en Agritechniek, Wageningen, The Netherlands, 51 pp.
Janssen, E., 1991
Haalbaarheidsstudie van mechanisatiesystemen in de pot- en containerteelt. Analyse van drie mechanisatiesystemen bij boomkwekerij v.d. Starre (Feasibility study of mechanization systems in pot and container growing. Analysis of three mechanization systems at nursery v.d. Starre). MSc Report, Wageningen Agricultural University, Department of Agricultural Engineering and Physics, Wageningen, The Netherlands, 67 pp (+ Appendices).
Janssen, F., 1987
Modelling and simulation of internal transport in greenhouses - a case study for the transportable bench growing system. MSc Report, Wageningen Agricultural University, Department of Agricultural Engineering, Wageningen, The Netherlands, 55 pp.
Jeffcoat, D.E. & R.L. Bulfin, 1993,
Simulated annealing for resource-constrained scheduling. European Journal of Operational Research, 70, 43-51.

Johnson, D.S. & L.A. McGeoch, 1997
The traveling salesman problem: a case study. In: Aarts, E.H.L. & J.K. Lenstra (eds.), Local search in combinatorial optimization, John Wiley & Sons, Chichester, UK, 215-310.
Kleijnen, J.P.C. & W. van Groenendaal, 1992
Simulation, a statistical perspective. John Wiley & Sons, Chichester, UK, 241 pp.
Kwaak, J. van der, 1989
Tuinbouwautomatisering schakelt versnelling hoger (Horticultural automation goes in the next gear). Vakblad voor de Bloemisterij, 44, 36, 34-36.
Kwaak, J. van der & P. van Weel, 1990
Mechanisering nu en in de toekomst (Mechanization now and in the future). Vakblad voor de Bloemisterij, 45, 9, 28-29.
KWIN, 1997
Kwantitatieve informatie voor de Glastuinbouw 1997-1998 (Quantitative information for horticulture under glass 1997-1998). Viertiende herziene druk, Proefstation voor Bloemisterij en Glasgroente, Naaldwijk, 176 pp (+ Appendices).
Langhans, R.W., 1983
Greenhouse Management - A guide to structures, environmental control, materials handling, crop programming and business analysis. Second edition, Halcyon Press of Ithaca, Ithaca, New York, USA, 270 pp.
Lawler, E.L., J.K. Lenstra, A.H.G. Rinnooy Kan & D.B. Shmoys, 1993
Sequencing and scheduling: algorithms and complexity. In: Graves, S.C. et al. (eds.), Logistics of production and inventory, Handbooks in OR & MS, 4, Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 445-522.
LEI-DLO & CBS, 1998
Land- en tuinbouwcijfers 1998 (Agricultural and horticultural figures 1998). Landbouw-Economisch Instituut en Centraal Bureau voor de Statistiek, ‘s-Gravenhage, The Netherlands, 295 pp.
Lentz, W. , 1985
Der Einsatz von Mikrocomputern bei der Planung von Produktionsprogrammen in Gartenbaubetrieben (Implementing micro computers for production planning in horticultural firms). Informationsverarbeitung Agrarwissenschaft, Band 9, Verlag Eugen Ulmer, Stuttgart, BRD, 189-198.
Lentz, W. & H.H. Buchwald , 1989
Two different computer based models for production planning in vegetable and floriculture firms. Acta Horticulturae, 237, 71-79.
Leutscher, K.J., 1995
Operational management in pot plant production. Dissertation, Wageningen Agricultural University, Department of Horticulture, Wageningen, The Netherlands, 285 pp.
Looi, C.-K., 1992
Neural network methods in combinatorial optimization. Computers Ops Res., 19, 3/4, 191-208.
Lombardot, G., 1989
“Adi serre”: support system for greenhouse investment. Acta Horticulturae, 237, 81-83.
Ludwig, S. , 1989
Consideration of risk in production planning. Acta Horticulturae, 248, 423-430.
Ludwig, S., 1993
Planning of cultivation periods under uncertainty in pot plant production. Acta Horticulturae, 340, 227-234.
Luijks, J.D.G.F., 1993
Analyse van het intern transportsysteem van Kwekerij De Goede Hoop (Analysis of the internal transport system of nursery ‘De Goede Hoop’). IMAG-nota, P 93-22, Instituut voor Milieu- en Agritechniek, Wageningen, The Netherlands, 50 pp.

Meuleman, J. & P. van Weel, 1997
Ontwikkelingen in wetenschap en technologie: Agrofysica, Informatie- en Communicatietechnologie in de plantaardige productie (Dynamics in science and technology: agrophysics, information and communication technology in crop production). NRLO report 97/24, National Council for Agricultural Research, The Hague, The Netherlands, 62 pp.
Misset, 1996
De landbouwtelling 1996 (The agricultural census 1996). Misset, Doetinchem, The Netherlands, 101-110.
Molen, M. van der, 1992
Automatisering van het intern transport in de rozen- en tomatenteelt gebruik makend van transporttabletten (Automation of internal transport in the production of roses and tomatos using transportable benches). Scriptie, Landbouw Universiteit, Vakgroep Landbouwtechniek, Wageningen, The Netherlands, 65 pp (+ Appendices).
Muller, C., E.H. Magill, P. Prosser & D.G. Smith, 1993
Distributed Genetic Algorithms for resource allocation. In Dorn, J. & K.A. Froeschl (eds.), Scheduling of production processes, Ellis Horwood, Chichester, UK, 61-69.
Nelson, P.V., 1991
Greenhouse operation and management. 4th edition, Prentice-Hall, Englewood Cliffs, New Jersey, USA, 612 pp.
Nienhuis, J.K. & E. van Rijssel, 1988
Developments in management information systems in the glasshouse horticulture in The Netherlands. Acta Horticulturae, 223, 193-202.
Oosten, H.J. van, 1998
Horticultural research in the Netherlands: changes and challenges for 2010. NRLO Report 98/26, National Council for Agricultural Research, The Hague, The Netherlands, 16 pp (+ Appendices).
Peeters, J., 1990
Validation of the model for planning greenhouse operations. MSc Report, Wageningen Agricultural University, Department of Agricultural Engineering, Wageningen, The Netherlands, 81 pp.
Ploeger, C., 1992
De structuur van de kamerplantenteelt in Nederland (The structure of pot plant growing in the Netherlands). Publicatie 4.130, Landbouw-Economisch Instituut, ‘s-Gravenhage, The Netherlands, 96 pp.
Projectgroep Evaluatie Simulatie Software, 1989
Evaluatie simulatie-software (Evaluation simulation software). GLW-rapport LWA/89-23, Groep Landbouwwiskunde, Wageningen, The Netherlands, 21 pp.
Punnen, A.P. & Y.P. Aneja, 1993
Categorized assignment scheduling: a tabu search approach. Journal of the Operational Research Society, 44, 7, 673-679.
Punnen, A.P. & Y.P. Aneja, 1995
A tabu search algorithm for the resource-constrained assignment problem. Journal of the Operational Research Society, 46, 214-220.
Reeves, C.R., 1993
Improving the efficiency of tabu search for machine sequencing problems. Journal of the Operational Research Society, 44, 4, 375-382.
Reeves, C.R., 1995
Heuristics for scheduling a single machine subject to unequal job release times. European Journal of Operational Research, 80, 397-403.
Reilly, A., 1981
Using greenhouse space efficiency with rolling-pallet system. Florist's Review, June 4, 32-33.

Rijssel, E. van & J.K. Nienhuis , 1988
An information model as first step to design management info-systems in glasshouse horticulture. Acta Horticulturae, 223, 224-228.
Reuter, C., 1989
Simulation des innerbetrieblichen Transportes im Topfpflanzenbau (Simulation of internal transport in pot plant nurseries). Universität Hannover, Institut für Technik in Gartenbau und Landwirtschaft, Gartenbautechnische Informationen, Heft 29, Hannover, BRD, 145 pp.
Rochat, Y. & F. Semet, 1994
A tabu search approach for delivering pet food and flour in Switserland. Journal of the Operational Research Society, 45, 11, 1233-1246.
Rosier, J.M., 1994
Planning van het intern transport op potplantenbedrijven - Probleemomschrijving en wiskundige modellering (Planning internal transport in pot plant nurseries – Problem description and mathematical modelling). IMAG-nota, P 94-46, Instituut voor Milieu- en Agritechniek, Wageningen, The Netherlands, 53 pp.
Saedt, A.P.H.,1982
Decision support by linear programming under changing circumstances in a pot plant nursery. In: Hanf, C.-H. & G.W. Schiefer (eds.), Proceedings of the Third Symposium of the European Association of Agricultural Economists (EAAE), Kieler Wissenschaftsverlag Vauk, Kiel, BRD, 134-144.
Saedt, A.P.H. & E. Annevelink, 1988
A Decision Support System based upon a Transition Model for Production Planning in Pot plant Nurseries. Acta Horticulturae, 223, 234-241.
Saedt, A.P.H., Th.H.B. Hendriks & F.M. Smits, 1991
A transition planning method applied in a decision support system for pot plant nurseries. European Journal of Operational Research, 52, 2, 142-154.
Schumacher, S. & F.C. Weston , 1983
Linear programming: a tool for production planning in glasshouse floriculture. Acta Horticulturae, 147, 83-88.
Sierenberg & de Gans, 1994
PROSIM modelling language, reference manual, user guide and toolbox. Sierenberg & de Gans bv., Waddinxveen, The Netherlands.
Slack, N., S. Chambers, C. Harland, A. Harrison & R. Johnson, 1998
Operations Management. Second Edition, Pitman Publishing, London, UK, 862 pp.
Smeenge, A., 1991
Haalbaarheidsstudie van mechanisatiesystemen in de pot- en containerteelt. Analyse van vier mechanisatiesystemen bij kwekerij gebr. Jonkers Elshout bv. (Feasibility study of mechanization systems in pot and container growing. Analysis of four mechanization systems at nursery gebr. Jonkers Elshout bv). MSc Report, Wageningen Agricultural University, Department of Agricultural Engineering and Physics, Wageningen, The Netherlands, 61 pp (+ Appendices).
Sowell, R.S., D.H. Willits, P.V. Nelson, R.C. Brumfield, D.G. Sanford & A.J. Coutu, 1982
Optimizing greenhouse production management decisions using linear programming. Paper 82-5021, ASAE, St. Joseph, Michigan, USA, 18 pp (+ Figures).
Stein, A. & P.A. van Weel, 1980
Enkele technische en bedrijfseconomische aspecten van verrolbare tabletten bij de teelt van potplanten (Some technical and economical aspects of rolling benches used for growing pot plants). Rapport 4.91, Instituut voor Mechanisatie, Arbeid en Gebouwen, Wageningen, Landbouw Economisch Instituut, 's-Gravenhage, The Netherlands, 45 pp.

Stein, A., 1987
Een kostenvergelijking van de verschillende bedrijfssystemen in de potplantenteelt (A cost comparison of different growing systems in pot plant growing). Landbouw-Economisch Instituut (LEI-DLO), Publicatie 4.199, ‘s-Gravenhage, The Netherlands, 43 pp.
Stoffert, G. & H.-R. Rohlfing , 1984
Handling and transportation of potplants in glasshouses. Acta Horticulturae, 155, 437-442.
Tongeren, P. van & H.T.J. Peelen, 1984
Bedrijfskundige aspecten roltafels, transporttabletten en betonvloeren (Management aspects of rolling tables, transportable benches and concrete floors). Vakblad voor de Bloemisterij, 39, 5, 102-105.
Voort, H.L.J. van der, 1995
Vervolganalyse van het intern transportsysteem van Kwekerij De Goede Hoop ten behoeve van het onderzoek naar de scheduling van intern transport bij glastuinbouwbedrijven (Continuation of the analysis of the internal transport system at Kwekerij De Goede Hoop for scheduling research at horticultural companies). IMAG-nota, P 95-11, Instituut voor Milieu- en Agritechniek, Wageningen, The Netherlands, 49 pp.
Weel, P.A. van, 1986
Mechanization aspects of the culture on concrete floors. Acta Horticulturae, 187,129-135.
Weel, P.A. van, 1988
Goed intern transport brengt potplanten in stroomversnelling (Good internal transport brings pot plants in rapid movement). Vakblad voor de Bloemisterij, 43, 4, 74-75.
Weel, P.A. van, 1991
Integrated crop production systems - Transport and materials handling in the greenhouse. In: Automated Agriculture for the 21st century, Proceedings of the 1991 Symposium, ASAE Publication 11-91, ASAE, St. Joseph, USA, 458-467.
Weel, P.A. van, G.A. Giacomelli & K.C. Ting, 1991
Design procedure for integrated greenhouse production systems, with special emphasis on internal transport systems. Acta Horticulturae, 295, 177-184.
Weel, P.A. van, 1992
Overdrachtsrapportage project 12326: automatisering van de logistiek op het potplantenbedrijf (Transfer report project 12326: automation of logistics at a pot plant nursery). Institute of Agricultural and Environmental Engineering, Internal report, Wageningen, The Netherlands, 10 pp.
Weston, F.C. & S. Schumacher , 1983
Linear programming, economic interpretation of model results in a glasshouse floriculture environment. Acta Horticulturae, 147, 89-98.
Widmer, M., 1991
Job shop scheduling with tooling constraints: a tabu search approach. Journal of the Operational Research Society, 42, 1, 75-82.
Widmer, M. & A. Hertz, 1989
A new heuristic for the flow shop sequencing problem. European journal of Operational Research, 41, 186-193.
Zabeltitz, C. von, 1986
Gewächshäuser, Planung und bau (Glasshouses, planning and building). 2. Auflage, Verlag Eugen Ulmer, Stuttgart, BRD, 280 pp.

Appendix 1 Detailed results of company survey

This appendix contains the detailed results of a company survey of eleven pot plant nurseries in practice, which was conducted by Annevelink & van der Voort (1995 & 1996). The objective of the company survey was to describe the characteristics of internal transport systems in pot plant nurseries with transportable benches.

Table 1. Number of transportable benches and their size.

Company	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
Number of transportable benches	8 428	2 400	2 700	2 132	2 500	1 800	2 800	1 050	2 000	1 200	600
Length (m)	3.03	4.50	6.09	6.00	3.75	6.20	4.20	5.66	3.90	3.00	4.45
Width (m)	1.61	1.63	1.60	1.75	1.80	1.80	1.60	1.64	1.56	1.60	1.62
Area (m^2)	4.88	7.33	9.74	10.50	6.75	11.16	6.72	9.28	6.08	4.80	7.21

Table 2. Distribution of the number of transportable benches per row.

Number of transportable benches in a row	Number of these rows	Number of transportable benches in a row	Number of these rows	Number of transportable benches in a row	Number of these rows
7	2	22	80	53	44
10	18	23	34	54	18
11	4	26	43	61	26
12	24	28	36	70	23
13	34	29	137	104	26
20	80	34	44	113	13
21	6	36	113		

Table 3. Capacity of the buffers in the processing compartment: maximum number of transportable benches.

Buffer	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
Output potting	18	20	416	0	20	24	15	9	12	13	8
input spacing/sorting	6	15	11	-	20	35	25	35	30	13	19
Output spacing/sorting	18	20	416	-	20	59	25	13	20	13	8
input harvesting	10	10	22	11	20	131	16	35	5	18	18
Empty transportable benches	252	320	210	126	420	150	270	160	143	120	72

Table 4. Area of different components of each pot plant nursery.

Company	Total area (m^2)	Total growing area (m^2)	Total net area transportable benches (m^2)	Total area paths (m^2)	Work area (m^2)
(1)	60 000	56 000	48 920	2 150	2 360
(2)	35 000	33 520	18 320	1 230	5 000
(3)	31 000	30 900	28 134	570	2 340
(4)	30 100	24 680	22 320	1 820	3 870
(5)	29 000	26 560	22 720	1 280	1 300
(6)	26 000	22 680	21 031	580	2 500
(7)	22 000	18 170	15 660	1 500	1 700
(8)	16 400	16 020	9 020	810	2 420
(9)	14 000	13 000	12 000	690	1 600
(10)	7 600	6 500	5 850	270	450
(11)	6 300	5 200	4 290	480	1 150

Table 5. Area of different components as a percentage of the total growing area of each pot plant nursery (including concrete floors).

Company	Total net area transportable benches (%)	Area paths (%)	Work area (%)	Company	Total net area transportable benches (%)	Area paths (%)	Work area (%)
(1)	87	4	4	(7)	86	8	9
(2)	55	4	15	(8)	56	5	15
(3)	91	2	8	(9)	92	5	12
(4)	90	7	16	(10)	90	4	7
(5)	86	5	5	(11)	83	9	22
(6)	93	3	11				

Table 6. The number of growing compartments in the growing area per company.

Number of growing compartments	Number of companies	Number of growing compartments	Number of companies
1	1	7	1
2	3	8	1
4	1	10	1
6	3		

Table 7. Accessibility of transportable benches in rows of a growing compartment.

Type	Number of companies
Last in First Out (LIFO)	4
First in First Out (FIFO)	4
Combination (LIFO/FIFO)	3

Table 8. Presence of half or fully automatic guided vehicles.

Number of automatic guided vehicles	Number of companies
0	2
1	5
2	3
3	1

Table 9. Capacity of automatic guided vehicles.

Number of transportable benches per automatic guided vehicle	Number of companies
1	5
2	2
3	4
4	2
5	0
6	1

Table 10. Maximum transportation distance of automatic guided vehicles.

Company	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
Maximum transport distance automatic guided vehicle (m)	500	185	-	70	160	192	230	144	44	108	-

Table 11. Range of the total crop duration and duration of each growing phase.

Company	Range total crop duration (week)	Phase 1 (week)	Phase 2 (week)	Phase 3 (week)	Phase 4 (week)	Phase 5 (week)
(10)	8-10	4	4-6	-	-	-
(10)	8-10	3	3	2	-	-
(4)	9	3	6	-	-	-
(6)	9-10	4-5	4-5	-	-	-
(7)	10	4	4	2	-	-
(6)	10-12	6	4-6	-	-	-
(9)	10-12	4-5	6-7	-	-	-
(3)	10-18	3-6	3-6	3-6	-	-
(11)	11	4	4	3	-	-
(8)	11-12	5-6	3	2-3	-	-
(6)	13-17	5-6	4-5	4-5	-	-
(10)	16-18	4	4	4	4	-
(3)	16-26	6-8	6-8	6-8	-	-
(2)	18-22	4	14-18	-	-	-
(6)	20-28	4	4	4-6	4-7	4-7
(1)	39-52	9-13	13-17	17-22	-	-
(5)	52-60	26	26	-	-	-
(5)	52-60	13	13	17	17	-

Table 12. Pot size and the number of pot plants per m^2 in each growing phase.

Company	Pot size (cm)	Phase 1 (plants/m^2)	Phase 2 (plants/m^2)	Phase 3 (plants/m^2)	Phase 4 (plants/m^2)	Phase 5 (plants/m^2)
(6)	8.5	143	47	-	-	-
(6)	8.5	143	60	-	-	-
(6)	8.5 -> 17	143	60	27	13	6
(4)	9	50	22	-	-	-
(5)	9	108	54	-	-	-
(5)	9 -> 15	108	54	27	14	-
(10)	10.5	83	42	-	-	-
(10)	10.5 -> 14	83	35	21	10	-
(2)	11	88	88	-	-	-
(6)	11	80	39	20	-	-
(3)	12	72	36	18	-	-
(7)	12	64	31	26	-	-
(9)	12	75	28-32	-	-	-
(1)	13	seedlings	56	31	-	-
(11)	13	55	22	15	-	-
(2)	14	55	55	-	-	-
(10)	14	52	22	13	-	-
(2)	15	46	46	-	-	-
(2)	17	37	37	-	-	-
(8)	17	26	11-13	6-9	-	-
(2)	19	29	29	-	-	-
(2)	21	23	23	-	-	-

Table 13. Number of pot plants produced per year.

Company	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
Number of plants/year (million)	3.5	2.8	3.0	3.2	1.3	6.5	2.3	0.5	2.6	1.3	0.5

Table 14. Number of pot plant species per company.

Number of species	Number of companies
1	4
2	4
10	1
11	1
25	1

APPENDIX 4.1 Example of an input file with the dates of potting, spacing and selling (*HndIdate.udf*)

```
@ SIMOPER  : 02/03/94 - 30/04/94                                        @
@ MADE ON  : 30/06/95                                                   @

@ number of rows, transportable benches in a row and on AGV and         @
@ critical length output buffer potting, output buffer spacing 1 and 2  @
31 61 3 22 25 28

@ start date simulation: dd mm yy                                       @
01 03 94

@Start_date   End_date   Crop_id   Num   Phase   Grade   Stage  @
  02 03 94    25 03 94    3801      2      2       1       40
  02 03 94    28 03 94    3801      1      2       2       43
  02 03 94    23 03 94    3801      1      2       3       36
  02 03 94    28 03 94    3803      2      2       2       43
  02 03 94    25 03 94    3803      2      2       1       40
  02 03 94    27 04 94    4071      3      1       0       20
  02 03 94    25 03 94    3802      1      2       1       40
  02 03 94    01 04 94    3841      3      2       1       40
  02 03 94    28 03 94    3841      1      2       2       43
  02 03 94    02 04 94    3841      1      2       1       40
  02 03 94    01 04 94    3841      1      2       2       43
  02 03 94    05 04 94    3840      1      2       2       43
  02 03 94    02 04 94    3839      1      2       1       40
  02 03 94    01 04 94    3839      2      2       1       40
  02 03 94    01 05 94    4072      5      1       0       20
  02 03 94    05 04 94    3839      1      2       2       43
  02 03 94    01 04 94    3838      1      2       1       40
  02 03 94    02 04 94    3837      1      2       1       40
  02 03 94    01 04 94    3837      1      2       2       43
  02 03 94    01 04 94    3837      1      2       1       40
  02 03 94    01 04 94    3836      2      2       1       40
  02 03 94    01 04 94    3836      1      2       2       43
  02 03 94    02 04 94    3836      4      2       1       40
  02 03 94    05 04 94    3836      2      2       2       43
    ....        ....      ....      .      .       .       ..
  30 04 94    30 05 94    4063      3      1       0       20
  30 04 94    30 05 94    4088      5      1       0       20
    -1
```

APPENDIX 4.2 Example of an input file with the filling of the rows (*Startpos.udf*)

@ SIMOCCUP : 01/03/94 @
@ MADE ON : 30/06/95 @

@ Row	Position	Container_id	Date	Crop	Phase	Grade	Stage
1	1	836	12 04 94	4066	1	0	20
1	2	252	12 04 94	4066	1	0	20
1	3	1026	04 04 94	4065	1	0	20
1	4	416	04 04 94	4065	1	0	20
1	5	830	04 04 94	4065	1	0	20
1	6	1136	04 04 94	4065	1	0	20
1	7	262	04 04 94	4065	1	0	20
1	8	688	12 04 94	4064	1	0	20
1	9	641	12 04 94	4064	1	0	20
1	10	66	12 04 94	4064	1	0	20
1	11	1561	12 04 94	4064	1	0	20
1	12	672	12 04 94	4064	1	0	20
1	13	1600	12 04 94	4064	1	0	20
1	14	665	29 03 94	4055	1	0	20
1	15	781	29 03 94	4055	1	0	20
1	16	320	29 03 94	4055	1	0	20
1	17	371	29 03 94	4055	1	0	20
1	18	1371	29 03 94	4055	1	0	20
1	19	1207	11 04 94	4063	1	0	20
1	20	603	11 04 94	4063	1	0	20
1	21	923	11 04 94	4063	1	0	20
1	22	948	29 03 94	4062	1	0	20
2	1	465	24 03 94	3786	2	3	46
2	2	41	18 03 94	3786	2	2	43
2	3	505	18 03 94	3786	2	2	43
2	4	1492	22 03 94	3809	2	2	43
2	5	1708	24 03 94	3809	2	3	46
2	6	622	30 03 94	3809	2	2	43
..	..	....	....	....	.	.	..
31	56	1770	11 04 94	3925	1	0	20
31	57	1760	21 03 94	3930	1	0	20
31	58	219	21 03 94	3930	1	0	20
31	59	469	21 03 94	3930	1	0	20
31	60	1800	21 03 94	3930	1	0	20
31	61	998	21 03 94	3930	1	0	20
-1							

APPENDIX 4.3 Example of an output file with a report of the transport times (*Report.udf*)

Input file : E940301H
Date : 18/10/1996 Time: 16:50:38
Strategy : 3
Cont.pos.in : Y
Cont.pos.out : N

Execut. date	Fill %	Day time	Total time	Input %	Reloc. %	Output %
02 03 94	95.46	17.34	17.34	31.46	40.22	28.32
03 03 94	93.19	14.77	32.11	24.93	47.54	27.53
04 03 94	93.44	14.29	46.41	22.61	53.43	23.96
05 03 94	94.39	7.02	53.43	22.51	55.01	22.48
06 03 94	93.82	3.06	56.49	21.29	56.78	21.93
07 03 94	94.14	19.93	76.42	19.34	61.01	19.65
08 03 94	93.38	18.15	94.57	17.57	63.40	19.04
09 03 94	93.44	18.46	113.03	17.65	63.39	18.96
10 03 94	91.80	14.16	127.19	17.38	63.15	19.47
11 03 94	91.61	20.96	148.16	16.91	64.09	19.00
12 03 94	91.93	12.33	160.49	16.34	65.32	18.35
13 03 94	90.10	11.25	171.74	15.27	66.71	18.02
14 03 94	91.87	14.80	186.54	16.05	66.11	17.84
15 03 94	91.17	20.25	206.79	15.74	66.44	17.81
16 03 94	90.54	16.73	223.52	16.34	65.36	18.31
17 03 94	90.86	22.03	245.55	16.57	65.38	18.05
18 03 94	84.87	23.88	269.43	15.42	66.23	18.35
19 03 94	90.73	7.08	276.51	16.55	65.36	18.10
20 03 94	90.04	2.71	279.22	16.39	65.52	18.09
21 03 94	85.69	23.72	302.93	15.68	65.99	18.33
22 03 94	90.23	15.05	317.99	16.93	64.78	18.29
23 03 94	85.62	23.50	341.49	16.21	65.27	18.52
24 03 94	84.24	24.00	365.49	16.10	65.27	18.63
25 03 94	84.68	23.91	389.40	16.14	65.35	18.51
26 03 94	87.89	9.28	398.69	16.75	64.83	18.42
27 03 94	87.89	0.00	398.69	16.75	64.83	18.42
28 03 94	86.32	23.77	422.46	16.38	65.33	18.29
29 03 94	88.71	13.06	435.52	17.00	64.70	18.30
30 03 94	91.55	18.13	453.64	17.32	64.65	18.04
31 03 94	92.69	10.58	464.22	17.60	64.43	17.97
01 04 94	88.71	23.72	487.94	16.74	65.58	17.68

APPENDIX 4.4 Components and classes in TRANSIM

This appendix gives an overview of the components and classes that are used in the simulation model TRANSIM. They are placed in alphabetical order for easy access in the source of the computer program. Attributes in Prosim can be of the type integer, real, character, macro and reference. The attributes of the components and classes are listed in the order of these types.

Cart

This component describes the Automatic Guided Vehicle (AGV) that moves the transportable benches.

Attributes of the type integer:

maxload	the maximum number of transportable benches that may be transported in one load;
speed	attribute to correct the computation of the distance the AGV travelled in case a different time unit is used (the standard time unit for this simulation is day).

Attributes of the type reference:

load	reference to the set of transportable benches that is on the AGV;
workcont	reference to the transportable bench that is being (un)loaded.

Container

This class describes the transportable benches on which the pot plants are grown and transported. In TRANSIM an instance of this class is created for each transportable bench in the growing compartment. The attributes contain different values for each transportable bench.

Attributes of the type integer:

number	unique identification of the transportable bench;
xval	value to identify the horizontal position of the AGV, necessary for animation;
yval	value to identify the vertical position of the AGV, necessary for animation.

Attributes of the type reference:

destination	reference to the destination row of the transportable bench;
crop_id	reference to the crop where the transportable bench (and crop batch) belongs to. This reference is necessary to find the correct crop for a transportable bench in expressions.

Control

This component assigns which transportable benches must be transported to which rows and when this must occur. This component is not directly related to a machine, but created to simulate the transport decisions.

Attributes of the type integer:

help	this attribute serves as a counter for the number of transportable bench-figures that are used in the animation;
crit_length1	if the output buffer of the potting machine contains more transportable benches than the value of *crit_length*, the output buffer must be emptied. The value is read from the file *Hndldate.udf*;
crit_length2	like crit_length_1 but for the output buffer of the sorting machine with grade 1;
crit_length3	like crit_length_1 but for the output buffer of the sorting machine with grade 2 and 3;
help4	used in ‘contrmod’ to store a temporary result when searching for a row where a certain crop batch is located.

Attribute of the type real:

help3	used in ‘contrmod’ to store a temporary result during the calculation of the transportation time;

Attributes of the type reference:

to_greenh	reference to the crop batch that is being transported to the growing compartment at the moment;
reloc_row	reference to the row where the relocated transportable benches must be placed;
rowmemory[40]	used in macro ‘Strat4’ and ‘Strat5’ to make a temporary backup of the set *rows*;
to_outbuf	reference to the crop batch that is being transported from the growing compartment to the input buffers for spacing and selling;
testcont	reference to container, used to find out if there are still transportable benches to be transported from the relocation row;
front_cont	reference to the first transportable bench of a specified row.

Crop

This class describes a crop. A crop contains one or more crop batches and each crop batch contains one or more transportable benches. This way it is possible to pot, space or sell one crop at different dates, which is common practice.

Attribute of the type integer:

cropnumber	unique identification of a crop.

Attribute of the type reference:

crop batches	the set of transportable benches that form a part of a crop that may be handled as one unit, all transportable benches have the same potting, spacing and selling days.

Crop batch

This class identifies a group of transportable benches, belonging to one crop, that may be handled as one unit with the same potting, spacing and selling dates and the same grade and phase.

Attributes of the type integer:

grade	indication of the quality of the crop batch (1=best quality, 2 or 3);
phase	phase of the crop batch (1=after potting, 2=after spacing);
potday	day this crop batch must be potted at the potting machine;
potted	indicates if the crop batch is already potted or not, used to select the next crop batch;
sellday	day this crop batch must be sold for potting;
stage	stretch of time between two operations on the crop batch.

Attributes of the type reference:

containers	reference to the set of transportable benches that belong to the crop batch.

Greenhouse

This component contains all attributes concerning the general layout of the growing compartment.

Attributes of the type integer:

maxrows	the maximum number of rows in the growing compartment, including buffer rows;
totalcont	number of transportable benches in the growing compartment, used to calculate the filling percentage of the growing compartment.

Attributes of the type reference:

rows	reference to the total set of rows in the growing compartment;
crops	reference to the total set of crops including the crops that are not yet potted;
buffer_out[3]	reference to the rows that are used as output buffers from the machines in the working area. Row 1 is the output buffer for the potting machine, rows 2 and 3 are output buffers for spacing;
buffer_in[2]	reference to the rows that are used as supply buffers. Row 4 is the supply buffer for spacing and row 5 for selling.

Main

This global component is automatically initiated when the simulation program starts. From this component all other components are activated. The attributes declared in ‘main’ are used for global operations, that are not directly connected to other components.

Attributes of the type integer:

beginday	contains the daynumber at the start of the simulation; it is necessary to do the day to date conversion;
beginyear	year the simulation was started, read from *Hndldate.udf* file;
cnum	container_id when reading from *Startpos.udf* file (filling of growing compartment);
colorcount	to assign a colour to every crop, depending from the crop_id;

counter	counter in for-to loop when initializing new transportable benches;
cropnum	crop_id when reading from file;
date[4]	array to store the day, month and year of an experiment;
date_fetched	used as an aid in the procedure to get the date of the experiment;
daynr	used in the ‘daydate’ macro for conversion;
daytime	used to determine the transport time on each day of the simulation run;
distance	distance between two stopping positions of the AGV;
dummy	used in macro *daydate* as a dummy for the return attribute;
endday	used to specify the number of days in a simulation run;
i	index in for-to loop when initializing transportable benches;
i2	index in for-to loop counting the number of rows;
i3	index in for-to loop counting the number of transportable benches in a row;
i5	index of *buffer_out[3]*;
i6	index in for-to loop when spacing pots;
maketime	daynumber when a crop is to be processed, converted from the date formed by the next three attributes;
maketime1	day (in date) when a crop is to be processed, read from file;
maketime2	month when a crop is to be processed, read from file;
maketime3	year when a crop is to be processed, read from file;
month1	used in the ‘daydate’ macro for conversion;
last_stop	point where the AGV stopped the last time;
name_fetched	used to store the actual name of the input file *Hndldate.udf*;
num_in	index of input buffers for selling and spacing;
num_out	index of output buffers of potting and spacing;
posnum	position of transportable bench in the row, read from file *Startpos.udf*;
row_counter	index to count the rows;
rowlength	length of the rows, read from file *Hndldate.udf*;
rownum	row_id of the row that contains a certain transportable bench;
selltime	daynumber when the crop will be sold, converted from the next three attributes;
selltime1	day (in date) when the crop will be sold, read from file;
selltime2	month when the crop will be sold, read from file;
selltime3	year when the crop will be sold, read from file;
this_stop	point where the AGV stopped this time;
time[4]	used to store the hours, minutes and seconds of the starting time of an experiment;
time_fetched	used as an aid in the procedure to get the time;
travelled	Boolean, indicates if the AGV has already travelled to its destination;
x	holds the answer to question which rule of thumb must be used;
xgrade	grade of the crop, read from file *Startpos.udf*;
xphase	phase of the crop, read from file *Startpos.udf*;
xsellday	day to sell the crop, read from file *Startpos.udf*;
xstage	stage of the crop, read from file *Startpos.udf*;
year1	used in the ‘daydate’ macro for conversion.

Attributes of the type real:

reloctime	time the AGV is busy relocating transportable benches;
start_in	start time for input of transportable benches into the growing compartment;
start_out	start time for output of transportable benches from the growing compartment;
start_reloc	start time for relocating transportable benches in the growing compartment;
transtime	total time for all transport movements;
traveltime	time the AGV is transporting transportable benches;
traveltime_in	time the AGV is busy with input of transportable benches into the growing compartment;
traveltime_out	time the AGV is busy with output of transportable benches from the growing compartment.

Attributes of the type character:

contpos	holds the answer on the question if the filling should be read from file *Startpos.udf*;
contpos_out	holds the answer on the question if the filling must be written to file *Finalpos.udf*;
fname	actual name of the input file *Hndldate.udf*;
pname	extension of the input file *Hndldate.udf*.

Attributes of type macro:

strategy[6]	contains a link to the rule of thumb to be used when parking transportable benches in the rows.

Attributes of the type reference:

usedcont	reference to the set of container_id's that are already in use, necessary to assign a unique id to each transportable bench;
to_shed	reference to the set of transportable benches that must be transported to the working area on this day, either for spacing or for selling;
test_cp	reference to a part of a crop;
outset	reference to the set of transportable benches that must be placed in the output buffer after spacing (rows 2 and 3);
potset	reference to the set of transportable benches that must be placed in the output buffer of the potting machine (row 1).

Potmachine

This component describes the potting process. It attaches an identification to the transportable benches when pot plants are potted.

Attributes of the type integer:

help2	number of transportable benches in a crop when potting;
potm_busy	a boolean that indicates whether the potting machine is working or not.

Attributes of the type reference:

to_potbuf	reference to the crop batch that is being processed;
potcont	reference to the transportable bench that is being processed.

Removal
This component is used to start the process of the output movements of transportable benches from the growing compartment to the working area. The component does not contain any attributes.

Row
This component contains attributes concerning the filling and identification of rows in the growing compartment.

Attributes of the type integer:

maxoccupation	maximum number of transportable benches allowed in a row;
rownumber	unique identification of the row;
outday	day when the first crop in the row must be transported.

Attribute of the type reference:

occupation	reference to the set of transportable benches that is placed in a row.

Seller
This component describes the selling process. The seller component does not contain any attributes.

Set-date
This component shows the date in the upper right-hand corner of the screen.

Spacing
This component describes the spacing process.

Attributes of the type integer:

help5	number of transportable benches in the crop that is being spaced;
spacing_busy	boolean that indicates whether the spacing machine is working or not.

Attributes of the type reference:

to_spacingbuf	reference to the crop batch that is being spaced;
spacingcont	reference to the transportable bench that is being spaced.

Write-filling
This component creates an output file (*Finalpos.udf*) that contains the positions of all transportable benches in the growing compartment. The component does not own attributes.

APPENDIX 4.5 Results of simulation experiments

This appendix gives the results of the simulation experiments with five different rules of thumb on internal transport.

Table 1. The interruption day of experiments with different starting dates and five different rules of thumb. Day 31 indicates an interruption that was requested by the settings of the experiment.

Starting date experiment	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
01/03/94	20	23	31	31	31
31/03/94	8	8	8	8	8
30/04/94	31	31	31	31	31
01/06/94	14	14	31	14	14
30/06/94	15	5	5	5	10
31/07/94	22	22	22	22	22
31/08/94	31	31	31	31	31
29/09/94	31	31	31	31	31
31/10/94	26	31	26	26	26
30/11/94	31	31	21	31	21
31/12/94	10	10	24	10	10
03/04/95	31	14	14	31	31
Mean	22.5	20.9	22.9	22.6	22.2

Table 2. The percentage of the growing compartment, that is filled with transportable benches after an unrequested interruption of the experiment, while applying five different rules of thumb for experiments with different starting dates. No percentage is given when the experiment was not interrupted.

Starting date experiment	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
01/03/94	97.48	97.41	-	-	-
31/03/94	96.72	96.72	96.72	96.72	96.72
30/04/94	-	-	-	-	-
01/06/94	78.31	78.06	-	78.94	80.08
30/06/94	99.56	98.30	99.18	96.41	96.85
31/07/94	96.60	96.97	96.91	96.97	96.85
31/08/94	-	-	-	-	-
29/09/94	-	-	-	-	-
31/10/94	96.91	-	96.28	96.34	97.10
30/11/94	-	-	96.47	-	96.34
31/12/94	97.29	96.78	99.37	97.60	96.60
03/04/95	-	97.73	97.86	-	-
Mean	94.70	94.57	97.54	93.83	94.36

Table 3. *Total transport time* after 7, 14, 21 and 28 days during the same simulation run (depending on the moment of interruption), while applying five different rules of thumb for experiments with different starting dates.

Starting date experiment	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
	Total transport time (hours) after 7 days				
01/03/94	107.13	108.75	94.57	93.91	104.87
31/03/94	62.64	59.48	49.15	52.98	53.67
30/04/94	77.08	82.77	57.06	63.40	63.59
01/06/94	87.32	85.07	72.69	81.08	77.10
31/07/94	83.12	79.87	63.27	67.57	69.67
31/08/94	73.93	72.47	60.78	56.62	58.21
29/09/94	48.16	52.16	41.36	46.25	47.18
31/10/94	74.90	70.49	45.79	59.64	56.18
30/11/94	45.34	41.80	34.41	41.02	39.14
31/12/94	48.33	42.16	36.27	38.60	39.97
03/04/95	106.39	112.46	87.30	98.08	95.38
Mean	74.03	73.41	58.42	63.56	64.09
	Total transport time (hours) after 14 days				
01/03/94	246.48	257.41	206.79	226.36	234.96
30/04/94	179.07	177.64	139.46	150.64	150.89
01/06/94	225.46	223.97	196.62	208.99	201.02
31/07/94	176.36	163.69	132.72	153.59	152.73
31/08/94	167.26	160.45	136.33	121.21	117.67
29/09/94	116.17	113.55	80.42	98.72	96.23
31/10/94	164.92	162.47	114.17	134.54	129.27
30/11/94	126.48	120.17	106.38	116.96	115.29
03/04/95	216.05	221.13	187.11	204.12	194.61
Mean	179.81	177.83	144.44	157.24	154.74
	Total transport time (hours) after 21 days				
30/04/94	298.97	286.27	225.76	247.92	237.31
31/07/94	305.26	278.68	229.68	251.74	258.71
31/08/94	256.86	249.98	193.24	193.19	185.74
29/09/94	199.06	186.93	129.40	153.99	154.75
31/10/94	267.22	255.03	181.40	213.29	208.40
30/11/94	203.00	193.05	166.99	187.20	191.22
Mean	255.06	241.66	187.75	207.89	206.02
	Total transport time (hours) after 28 days				
30/04/94	417.31	414.22	340.60	358.64	351.86
31/08/94	358.48	343.70	252.54	267.66	258.28
29/09/94	277.04	265.66	188.88	221.77	217.87
Mean	350.94	341.19	260.67	282.69	276.00

Table 4. *Indexed total transport time* after 7, 14, 21 and 28 days during the same simulation run (depending on the moment of interruption), while applying five different rules of thumb for experiments with different starting dates. The total transport time of rule 3 always has the index 1.00.

Starting date experiment	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
		Indexed total transport time after 7 days			
01/03/94	1.13	1.15	1.00	0.99	1.11
31/03/94	1.27	1.21	1.00	1.08	1.09
30/04/94	1.35	1.45	1.00	1.11	1.11
01/06/94	1.20	1.17	1.00	1.12	1.06
31/07/94	1.31	1.26	1.00	1.07	1.10
31/08/94	1.22	1.19	1.00	0.93	0.96
29/09/94	1.16	1.26	1.00	1.12	1.14
31/10/94	1.64	1.54	1.00	1.30	1.23
30/11/94	1.32	1.21	1.00	1.19	1.14
31/12/94	1.33	1.16	1.00	1.06	1.10
03/04/95	1.22	1.29	1.00	1.12	1.09
Mean	1.29	1.26	1.00	1.10	1.10
		Indexed total transport time after 14 days			
01/03/94	1.19	1.24	1.00	1.09	1.14
30/04/94	1.28	1.27	1.00	1.08	1.08
01/06/94	1.15	1.14	1.00	1.06	1.02
31/07/94	1.33	1.23	1.00	1.16	1.15
31/08/94	1.23	1.18	1.00	0.89	0.86
29/09/94	1.44	1.41	1.00	1.23	1.20
31/10/94	1.44	1.42	1.00	1.18	1.13
30/11/94	1.19	1.13	1.00	1.10	1.08
03/04/95	1.15	1.18	1.00	1.09	1.04
Mean	1.27	1.25	1.00	1.10	1.08
		Indexed total transport time after 21 days			
30/04/94	1.32	1.27	1.00	1.10	1.05
31/07/94	1.33	1.21	1.00	1.10	1.13
31/08/94	1.33	1.29	1.00	1.00	0.96
29/09/94	1.54	1.44	1.00	1.19	1.20
31/10/94	1.47	1.41	1.00	1.18	1.15
30/11/94	1.22	1.16	1.00	1.12	1.15
Mean	1.37	1.30	1.00	1.11	1.10
		Indexed total transport time after 28 days			
30/04/94	1.23	1.22	1.00	1.05	1.03
31/08/94	1.42	1.36	1.00	1.06	1.02
29/09/94	1.47	1.41	1.00	1.17	1.15
Mean	1.37	1.33	1.00	1.10	1.07

Table 5. *Relocation time* after 7, 14, 21 and 28 days during the same simulation run (depending on the moment of interruption), while applying five different rules of thumb for experiments with different starting dates.

Starting date experiment	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
		Relocation time (hours) after 7 days			
01/03/94	72.22	73.90	59.96	57.90	69.56
31/03/94	33.81	30.44	21.46	24.04	24.84
30/04/94	47.19	53.26	27.39	33.66	34.39
01/06/94	50.59	48.52	36.85	44.50	40.03
31/07/94	53.33	50.21	32.51	37.47	39.54
31/08/94	46.10	44.85	33.70	29.28	30.17
29/09/94	19.43	24.05	11.71	17.59	18.57
31/10/94	43.97	42.15	14.37	28.39	25.03
30/11/94	22.17	19.11	12.29	18.53	16.51
31/12/94	26.74	20.33	15.01	17.07	18.21
03/04/95	73.78	79.68	54.22	65.11	63.10
Mean	44.49	44.23	29.04	33.96	34.54
		Relocation time (hours) after 14 days			
01/03/94	180.45	192.18	137.39	156.66	167.93
30/04/94	120.41	119.94	81.14	92.90	93.49
01/06/94	143.39	142.04	111.13	125.23	114.62
31/07/94	116.26	103.04	71.77	92.49	91.09
31/08/94	117.92	111.24	86.81	71.50	67.71
29/09/94	68.46	67.38	31.49	51.62	49.37
31/10/94	105.66	101.93	53.93	73.47	69.21
30/11/94	72.84	67.24	52.73	63.75	62.63
03/04/95	153.96	158.99	124.07	140.86	132.92
Mean	119.93	118.22	83.39	96.50	94.33
		Relocation time (hours) after 21 days			
30/04/94	210.50	197.96	138.28	160.16	149.39
31/07/94	211.15	184.12	134.27	155.63	162.24
31/08/94	186.10	179.21	121.72	122.33	113.65
29/09/94	130.28	119.80	58.14	85.13	86.54
31/10/94	176.50	163.17	90.36	120.27	116.39
30/11/94	128.44	118.17	92.56	113.09	117.06
Mean	173.83	160.40	105.89	126.10	124.21
		Relocation time (hours) after 28 days			
30/04/94	305.01	303.66	230.59	248.75	241.34
31/08/94	263.45	248.43	156.04	172.16	161.61
29/09/94	183.51	173.50	93.14	129.00	125.82
Mean	250.66	241.86	159.92	183.31	176.26

Table 6. *Indexed relocation time* after 7, 14, 21 and 28 days during the same simulation run (depending on the moment of interruption), while applying five different rules of thumb for experiments with different starting dates. The relocation time of rule 3 always has the index 1.00.

Starting date experiment	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
		Indexed relocation time after 7 days			
01/03/94	1.20	1.23	1.00	0.97	1.16
31/03/94	1.58	1.42	1.00	1.12	1.16
30/04/94	1.72	1.94	1.00	1.23	1.26
01/06/94	1.37	1.32	1.00	1.21	1.09
31/07/94	1.64	1.54	1.00	1.15	1.22
31/08/94	1.37	1.33	1.00	0.87	0.90
29/09/94	1.66	2.05	1.00	1.50	1.58
31/10/94	3.06	2.93	1.00	1.98	1.74
30/11/94	1.80	1.55	1.00	1.51	1.34
31/12/94	1.78	1.35	1.00	1.14	1.21
03/04/95	1.36	1.47	1.00	1.20	1.16
Mean	1.69	1.65	1.00	1.26	1.26
		Indexed relocation time after 14 days			
01/03/94	1.31	1.40	1.00	1.14	1.22
30/04/94	1.48	1.48	1.00	1.14	1.15
01/06/94	1.29	1.28	1.00	1.13	1.03
31/07/94	1.62	1.44	1.00	1.29	1.27
31/08/94	1.36	1.28	1.00	0.82	0.78
29/09/94	2.17	2.14	1.00	1.64	1.57
31/10/94	1.96	1.89	1.00	1.36	1.28
30/11/94	1.38	1.28	1.00	1.21	1.19
03/04/95	1.24	1.28	1.00	1.14	1.07
Mean	1.54	1.50	1.00	1.21	1.17
		Indexed relocation time after 21 days			
30/04/94	1.52	1.43	1.00	1.16	1.08
31/07/94	1.57	1.37	1.00	1.16	1.21
31/08/94	1.53	1.47	1.00	1.00	0.93
29/09/94	2.24	2.06	1.00	1.46	1.49
31/10/94	1.95	1.81	1.00	1.33	1.29
30/11/94	1.39	1.28	1.00	1.22	1.26
Mean	1.70	1.57	1.00	1.22	1.21
		Indexed relocation time after 28 days			
30/04/94	1.32	1.32	1.00	1.08	1.05
31/08/94	1.69	1.59	1.00	1.10	1.04
29/09/94	1.97	1.86	1.00	1.39	1.35
Mean	1.66	1.59	1.00	1.19	1.14

Table 7. *Total input-output time* after 7, 14, 21 and 28 days during the same simulation run (depending on the moment of interruption), while applying five different rules of thumb for experiments with different starting dates.

Starting date experiment	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
	Total input-output time (hours) after 7 days				
01/03/94	34.91	34.85	34.61	36.01	35.31
31/03/94	28.83	29.04	27.69	28.94	28.83
30/04/94	29.89	29.51	29.67	29.74	29.20
01/06/94	36.73	36.55	35.84	36.58	37.07
31/07/94	29.79	29.66	30.76	30.10	30.13
31/08/94	27.83	27.62	27.08	27.34	28.04
29/09/94	28.73	28.11	29.65	28.66	28.61
31/10/94	30.93	28.34	31.42	31.25	31.15
30/11/94	23.17	22.69	22.12	22.49	22.63
31/12/94	21.59	21.83	21.26	21.53	21.76
03/04/95	32.61	32.78	33.08	32.97	32.28
Mean	29.55	29.18	29.38	29.60	29.55
	Total input-output time (hours) after 14 days				
01/03/94	66.03	65.23	69.40	69.70	67.03
30/04/94	58.66	57.70	58.32	57.74	57.40
01/06/94	82.07	81.93	85.49	83.76	86.40
31/07/94	60.10	60.65	60.95	61.10	61.64
31/08/94	49.34	49.21	49.52	49.71	49.96
29/09/94	47.71	46.17	48.93	47.10	46.86
31/10/94	59.26	60.54	60.24	61.07	60.06
30/11/94	53.64	52.93	53.65	53.21	52.66
03/04/95	62.09	62.14	63.04	63.26	61.69
Mean	59.88	59.61	61.06	60.74	60.41
	Total input-output time (hours) after 21 days				
30/04/94	88.47	88.31	87.48	87.76	87.92
31/07/94	94.11	94.56	95.41	96.11	96.47
31/08/94	70.76	70.77	71.52	70.86	72.09
29/09/94	68.78	67.13	71.26	68.86	68.21
31/10/94	90.72	91.86	91.04	93.02	92.01
30/11/94	74.56	74.88	74.43	74.11	74.16
Mean	81.23	81.25	81.86	81.79	81.81
	Total input-output time (hours) after 28 days				
30/04/94	112.30	110.56	110.01	109.89	110.52
31/08/94	95.03	95.27	96.50	95.50	96.67
29/09/94	93.53	92.16	95.74	92.77	92.05
Mean	100.29	99.33	100.75	99.38	99.75

Table 8. *Indexed total input-output time* after 7, 14, 21 and 28 days during the same simulation run (depending on the moment of interruption), while applying five different rules of thumb for experiments with different starting dates. The total input-output time of rule 3 always has the index 1.00.

Starting date experiment	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
		Indexed total input-output time after 7 days			
01/03/94	1.01	1.01	1.00	1.04	1.02
31/03/94	1.04	1.05	1.00	1.05	1.04
30/04/94	1.01	0.99	1.00	1.00	0.98
01/06/94	1.02	1.02	1.00	1.02	1.03
31/07/94	0.97	0.96	1.00	0.98	0.98
31/08/94	1.03	1.02	1.00	1.01	1.04
29/09/94	0.97	0.95	1.00	0.97	0.97
31/10/94	0.98	0.90	1.00	0.99	0.99
30/11/94	1.05	1.03	1.00	1.02	1.02
31/12/94	1.02	1.03	1.00	1.01	1.02
03/04/95	0.99	0.99	1.00	1.00	0.98
Mean	1.01	1.00	1.00	1.01	1.01
		Indexed total input-output time after 14 days			
01/03/94	0.95	0.94	1.00	1.00	0.97
30/04/94	1.01	0.99	1.00	0.99	0.98
01/06/94	0.96	0.96	1.00	0.98	1.01
31/07/94	0.99	1.00	1.00	1.00	1.01
31/08/94	1.00	0.99	1.00	1.00	1.01
29/09/94	0.98	0.94	1.00	0.96	0.96
31/10/94	0.98	1.00	1.00	1.01	1.00
30/11/94	1.00	0.99	1.00	0.99	0.98
03/04/95	0.99	0.99	1.00	1.00	0.98
Mean	0.98	0.98	1.00	0.99	0.99
		Indexed total input-output time after 21 days			
30/04/94	1.01	1.01	1.00	1.00	1.01
31/07/94	0.99	0.99	1.00	1.01	1.01
31/08/94	0.99	0.99	1.00	0.99	1.01
29/09/94	0.97	0.94	1.00	0.97	0.96
31/10/94	1.00	1.01	1.00	1.02	1.01
30/11/94	1.00	1.01	1.00	1.00	1.00
Mean	0.99	0.99	1.00	1.00	1.00
		Indexed total input-output time after 28 days			
30/04/94	1.02	1.00	1.00	1.00	1.00
31/08/94	0.98	0.99	1.00	0.99	1.00
29/09/94	0.98	0.96	1.00	0.97	0.96
Mean	0.99	0.98	1.00	0.99	0.99

Appendix 6.1 Simple test cases

This appendix describes the initial situation of the simple test cases that were used for the experiments in Chapter 6.

S1: 20 crop batches distributed over 10 rows, 10 empty positions concentrated in 5 rows with crop batches (TM = 53)

	working area									
	-1	-1	c1	-1	-1	c2	-1	0	0	0
	c3	c4	-1	-1	-1	-1	-1	-1	0	0
	-1	-1	-1	-1	c5	-1	c6	-1	c7	0
p	-1	-1	-1	c8	c9	-1	-1	-1	0	0
a	c10	-1	-1	-1	-1	-1	c11	-1	-1	-1
t	-1	-1	-1	c12	-1	-1	-1	-1	-1	c13
h	-1	-1	-1	c14	-1	-1	-1	-1	-1	-1
	c15	c16	c17	-1	-1	-1	-1	-1	-1	-1
	c18	-1	-1	c19	-1	-1	c20	-1	0	0
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

S2: 20 crop batches distributed over 10 rows, 10 empty positions concentrated in 6 rows with crop batches (TM = 85)

	working area									
	-1	-1	-1	-1	-1	c1	-1	0	0	0
	c2	-1	-1	-1	-1	-1	c3	-1	0	0
	-1	-1	-1	-1	c4	-1	-1	-1	c5	0
p	-1	-1	-1	c6	c7	-1	-1	-1	0	0
a	c8	-1	-1	-1	-1	-1	c9	-1	-1	0
t	-1	-1	-1	c10	-1	-1	-1	c11	-1	c12
h	-1	-1	-1	c13	-1	-1	-1	-1	-1	c14
	-1	c15	-1	-1	-1	-1	-1	-1	c16	-1
	c17	-1	-1	-1	-1	-1	-1	-1	c18	c19
	-1	-1	-1	-1	-1	-1	-1	-1	c20	0

S3: 10 crop batches in one row, 10 empty positions concentrated in 3 rows without crop batches (TM = 10)

	working area									
	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
p	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
a	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
t	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
h	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	-1	-1	-1	-1	-1	-1	-1	-1	-1	0
	-1	-1	-1	-1	-1	-1	-1	-1	0	0
	-1	-1	-1	0	0	0	0	0	0	0

S4: 20 crop batches distributed over two rows, 10 empty positions concentrated in 3 rows without crop batches (TM = 20)

	working area									
	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	c11	c12	c13	c14	c15	c16	c17	c18	c19	c20
p	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
a	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
t	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
h	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	-1	-1	-1	-1	-1	-1	-1	-1	-1	0
	-1	-1	-1	-1	-1	-1	-1	-1	0	0
	-1	-1	-1	0	0	0	0	0	0	0

S5: 20 crop batches distributed over 7 rows, 10 empty positions concentrated in 5 rows, 3 of which contain crop batches (TM = 35)

	working area									
	-1	c1	-1	c2	-1	c3	-1	-1	-1	-1
	-1	c4	-1	c5	-1	-1	-1	-1	0	0
	-1	c6	-1	c7	-1	c8	-1	-1	-1	-1
p	-1	-1	-1	-1	-1	-1	-1	-1	0	0
a	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
t	-1	c9	-1	c10	-1	-1	-1	-1	0	0
h	-1	c11	-1	c12	-1	-1	-1	-1	-1	-1
	-1	c13	-1	c14	-1	c15	-1	-1	0	0
	c16	c17	c18	c19	c20	-1	-1	-1	-1	-1
	-1	-1	-1	-1	-1	-1	-1	-1	0	0

S6: 20 crop batches distributed over 6 rows, 10 empty positions concentrated in 6 rows, 4 of which contain crop batches (TM = 29)

	working area									
	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
	c11	-1	-1	c12	-1	-1	-1	-1	0	0
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
p	-1	-1	-1	c13	c14	c15	-1	-1	0	0
a	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
t	c16	-1	-1	-1	-1	-1	-1	-1	0	0
h	-1	-1	-1	c17	-1	-1	-1	-1	-1	-1
	-1	-1	-1	-1	-1	-1	-1	-1	-1	0
	c18	-1	c19	c20	-1	-1	-1	-1	-1	0
	-1	-1	-1	-1	-1	-1	-1	-1	0	0

Appendix 6.2 Example of the calculation of the number of transport movements for a given transport sequence

This appendix shows an example of the calculation of the number of transport movements, for a given transport sequence, in combination with Parking Method 1. The initial situation of a case with 10 crop batches distributed over 5 rows, 8 empty positions distributed over 4 rows and a Theoretical Minimum = 23 is:

	working area									
	-1	-1	c1	-1	-1	c6	-1	0	0	0
p	c2	c3	-1	-1	-1	-1	-1	-1	0	0
a	-1	-1	-1	-1	c4	-1	c7	-1	c10	0
t	-1	-1	-1	c9	c8	-1	-1	-1	0	0
h	c5	-1	-1	-1	-1	-1	-1	-1	-1	-1

In the application of the initial situation in SUGAL three columns are added to store extra information about the rows in the growing compartment:

	working area										Free	Low	High
	-1	-1	c1	-1	-1	c6	-1	0	0	0	3	1	6
p	c2	c3	-1	-1	-1	-1	-1	-1	0	0	2	2	3
a	-1	-1	-1	-1	c4	-1	c7	-1	c10	0	1	4	10
t	-1	-1	-1	c9	c8	-1	-1	-1	0	0	2	8	9
h	c5	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	5	5

Each line in the position-matrix describes the situation of a single row. The first 10 numbers on the line describe the status of each position in that row (0 = empty, -1 = neutral bench and ci = output bench i). The last three numbers of a line supply information about the number of empty positions in the row (Free), the lowest output bench number in the row (Low) and the highest transportable bench number in the row (High). These numbers are used internally in the fitness function to increase its efficiency. An ideal transport sequence of the crop batches in this simple test case is:

c2 c3 c1 c6 c4 c7 c9 c10 c5 c8

To calculate the fitness value, the output benches are removed from the growing compartment one by one, according to the transport sequence. Obstructing benches are moved to parking rows according to Parking Method 1. The next pages give an overview of the successive steps of the calculation. Each time, when an output bench has been removed, the new situation in the growing compartment is shown. The total number of transport movements and the number of the crop batch which has been removed are shown in the heading.

Total number of transport movements: 1 Crop removed: c2

	working area										Free	Low	High
	-1	-1	c1	-1	-1	c6	-1	0	0	0	3	1	6
p	c3	-1	-1	-1	-1	-1	-1	0	0	0	3	3	3
a	-1	-1	-1	-1	c4	-1	c7	-1	c10	0	1	4	10
t	-1	-1	-1	c9	c8	-1	-1	-1	0	0	2	8	9
h	c5	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	5	5

Total number of transport movements: 2 Crop removed: c3

	working area										Free	Low	High
	-1	-1	c1	-1	-1	c6	-1	0	0	0	3	1	6
p	-1	-1	-1	-1	-1	-1	0	0	0	0	4	0	0
a	-1	-1	-1	-1	c4	-1	c7	-1	c10	0	1	4	10
t	-1	-1	-1	c9	c8	-1	-1	-1	0	0	2	8	9
h	c5	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	5	5

Total number of transport movements: 5 Crop removed: c1

	working area										Free	Low	High
	-1	-1	c6	-1	0	0	0	0	0	0	6	6	6
p	-1	-1	-1	-1	-1	-1	-1	-1	0	0	2	0	0
a	-1	-1	-1	-1	c4	-1	c7	-1	c10	0	1	4	10
t	-1	-1	-1	c9	c8	-1	-1	-1	0	0	2	8	9
h	c5	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	5	5

Total number of transport movements: 8 Crop removed: c6

	working area										Free	Low	High
	-1	0	0	0	0	0	0	0	0	0	9	0	0
p	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0
a	-1	-1	-1	-1	c4	-1	c7	-1	c10	0	1	4	10
t	-1	-1	-1	c9	c8	-1	-1	-1	0	0	2	8	9
h	c5	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	5	5

Total number of transport movements: 13 Crop removed: c4

	working area										Free	Low	High
	-1	-1	-1	-1	-1	0	0	0	0	0	5	0	0
p	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0
a	-1	c7	-1	c10	0	0	0	0	0	0	6	7	10
t	-1	-1	-1	c9	c8	-1	-1	-1	0	0	2	8	9
h	c5	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	5	5

Total number of transport movements: 15 Crop removed: c7

path	working area										Free	Low	High
	-1	-1	-1	-1	-1	-1	0	0	0	0	4	0	0
p	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0
a	-1	c10	0	0	0	0	0	0	0	0	8	10	10
t	-1	-1	-1	c9	c8	-1	-1	-1	0	0	2	8	9
h	c5	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	5	5

Total number of transport movements: 19 Crop removed: c9

path	working area										Free	Low	High
	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	1	0	0
p	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0
a	-1	c10	0	0	0	0	0	0	0	0	8	10	10
t	c8	-1	-1	-1	0	0	0	0	0	0	6	8	8
h	c5	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	5	5

Total number of transport movements: 21 Crop removed: c10

path	working area										Free	Low	High
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0
p	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0
a	0	0	0	0	0	0	0	0	0	0	10	0	0
t	c8	-1	-1	-1	0	0	0	0	0	0	6	8	8
h	c5	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	5	5

Total number of transport movements: 22 Crop removed: c5

path	working area										Free	Low	High
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0
p	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0
a	0	0	0	0	0	0	0	0	0	0	10	0	0
t	c8	-1	-1	-1	0	0	0	0	0	0	6	8	8
h	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	1	0	0

Total number of transport movements: 23 Crop removed: c8

path	working area										Free	Low	High
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0
p	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0
a	0	0	0	0	0	0	0	0	0	0	10	0	0
t	-1	-1	-1	0	0	0	0	0	0	0	7	0	0
h	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	1	0	0

Appendix 7.1 Examples of the exact description of a real-scale test case

The real-scale test cases have been derived from data that were recorded at ‘Kwekerij de Goede Hoop’ (Section 3.5). The next pages show two examples of a real-scale test case.

The path is on the left side of the compartment. Each row represents a row in the growing compartment. Because it was difficult to represent rows on one line, they have been cut in two halves: the upper block specifies the situation at the front of the rows in the growing compartment and the lower block gives the situation from position 35 until the end of the rows. Transportable benches with a relevant crop batch are identified by integer numbers, neutral benches by ‘-1’ and free positions by ‘0’. The empty positions are shown at the left side of the row in this representation (because of the required format within the SUGAL application). In reality they are at the end of the row, behind the transportable benches. The last three numbers of a line supply information about the number of empty positions in the row, the lowest output bench number in the row and the highest transportable bench number in the row.

The first example of a real-scale test case (Case O1) belongs to Group O, which constitutes a planning situation where only output benches have to be transported. Each crop batch contains only one transportable bench. The second example (Case M1) belongs to Group M which represents a planning situation where a single crop batch may contain more than one output bench. Case M1 is based on Case O1. For example: bench number 1, 3 and 4 of Case O1 all belong to crop batch 1 in Case M1.

Case O1: 72 crop batches (one output bench per crop batch), 26 rows and 61 transportable benches per row

```
O1_Number_containers_in_pottingbuffer: 0
O1_Number_containers_in_spacingbuffer: 0

Lay_out_at_time_0:
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0  0  0  0  0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0  0  0  0  0  0  0  0 -1 -1 -1 -1 -1 -1  1  2  3  4 -1 -1 -1 -1 -1  5  6  7  8  9 -1 -1 -1 10
 0  0  0  0  0 -1 24 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0 25 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0  0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 26 27 28 29 30 31 32 33 34 35 36 37
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0  0  0 48 49 50 51 52 -1 53 54 55 56 57 -1 -1 -1 -1 -1 58 59 60 61 62 63 64 65 66 67 68 69 -1
 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 70 71 72 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

Lay_out_at_time_0: Continued from position 35 in each row:
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0  0  0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    3  0  0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    8  0  0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    3  0  0
11 -1 -1 12 13 14 -1 15 16 -1 17 18 -1 19 20 -1 -1 -1 -1 -1 21 22 -1 23 -1 -1 -1   10  1 23
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    5 24 24
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1 -1 -1 -1 -1   56  0  0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    5 25 25
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    5  0  0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0  0  0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    3  0  0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0  0  0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    2  0  0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    1  0  0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    1  0  0
38 39 40 41 42 43 44 45 46 47 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    1 26 47
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0  0  0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    5 48 69
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    1  0  0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1   18  0  0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    2  0  0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0  0  0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0 70 72
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0  0  0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0  0  0
```

Case M1: 26 crop batches (more than one output bench per crop batch; 72 output benches in total), 26 rows and 61 transportable benches per row; this case was derived from case O1: the initial situation is the same, but the output benches have been clustered in crop batches

```
M1_Number_containers_in_pottingbuffer:  0
M1_Number_containers_in_spacingbuffer:  0

Lay_out_at_time_0:
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0  0  0  0  0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0  0  0  0  0  0  0  0 -1 -1 -1 -1 -1 -1  1  2  1  1 -1 -1 -1 -1 -1  3  3  4  4  4 -1 -1 -1  4
 0  0  0  0  0 -1 10 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0 11 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0  0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 12 12 12 12 12 12 12 12 13 13 13 13
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0  0  0 16 17 17 18 17 -1 19 19 20 21 22 -1 -1 -1 -1 -1 23 23 23 23 23 23 23 24 23 23 24 23 -1
 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 25 26 25 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

Lay_out_at_time_0: Continued from position 35 in each row:
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0   0   0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    3   0   0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    8   0   0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    3   0   0
 4 -1 -1  5  6  7 -1  8  8 -1  5  2 -1  8  1 -1 -1 -1 -1 -1  4  9 -1  6 -1 -1 -1   10   1   9
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    5  10  10
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0   0   0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1 -1 -1 -1 -1   56   0   0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    5  11  11
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    5   0   0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0   0   0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    3   0   0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0   0   0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    2   0   0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    1   0   0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    1   0   0
14 14 14 14 15 15 15 15 15 15 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    1  12  15
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0   0   0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    5  16  24
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    1   0   0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1   18   0   0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    2   0   0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0   0   0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0  25  26
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0   0   0
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1    0   0   0
```

Appendix 7.2: Results of the experiments with local search methods

The Tables on the next pages give the results of the experiments performed with different parameter settings of local search methods. The results were divided into results on the effectiveness and on the efficiency of a method. The evaluation criteria to judge the *effectiveness* of the local search method were:

- *Frequency theoretical minimum (FTM):* the number of times that a method found a transport sequence with the theoretical minimum number of transport movements among ten runs with a fixed number of generations;
- *Mean minimum (MM):* the mean of all the minimum values for the number of transport movements that were found among ten runs with a fixed number of generations;
- *Best minimum (BM):* the best minimum value for the number of transport movements that was found among ten runs with a fixed number of generations;
- *Worst minimum (WM):* the worst minimum value for the number of transport movements that was found among ten runs with a fixed number of generations.

The best minimum (BM) and the worst minimum (WM) of the runs in an experiment were always given as an absolute distance to TM (+0, +1, +2, etc.).

Three evaluation criteria to judge the efficiency of a local search method were:

- *Mean Evaluation Counter (MEC):* the mean number of function evaluations that had to be made before a transport sequence with a certain number of transport movements was found among ten runs;
- *Best Evaluation Counter (BEC):* the lowest number of function evaluations that had to be made before a transport sequence with a certain number of transport movements was found among ten runs;
- *Worst Evaluation Counter (WEC):* the highest number of function evaluations that had to be made before a transport sequence with a certain number of transport movements was found among ten runs.

The statistical analysis specified if there was a significant difference between two local search methods at a level of significance $\alpha=0.05$. The compared methods are put into one of the two columns: a significant different or no significant difference. The ‘<’ sign indicates if the first method found a smaller value (MM or MEC) than the second method in the comparison. The p value is the smallest value of α for which the test results are statistically significant.

Group O. Only output benches

Table 1. Effectiveness and efficiency of the *local search methods* Simulated Annealing (temperature 120 and decay factor 0.999), Random Search and the Genetic Algorithm with cycle crossover, at Population Size 20.

Case	Method	FTM	MM	Sdev	BM	WM	Statistical analysis ($\alpha = 0.05$) significant difference	p	non-sign. difference	p
O1	SA	10	151.0	0.00	+0	+0	SA<RS	<0.01		
	RS	0	170.5	3.75	+13	+25	SA<GA	<0.01		
	GA-CX	0	186.4	18.88	+20	+74	RS<GA	0.02		
O2	SA	10	65.0	0.00	+0	+0	SA<RS	<0.01	RS<GA	0.12
	RS	0	68.4	0.70	+3	+5	SA<GA	<0.01		
	GA -CX	1	70.2	3.43	+0	+12				
O3	SA	10	181.0	0.00	+0	+0	SA<RS	<0.01	RS<GA	0.07
	RS	0	190.6	2.95	+5	+14	SA<GA	<0.01		
	GA-CX	0	200.1	15.46	+4	+58				
O4	SA	10	225.0	0.00	+0	+0	SA<RS	<0.01		
	RS	0	247.4	11.91	+10	+45	SA<GA	<0.01		
	GA-CX	0	270.8	14.51	+25	+75	RS<GA	<0.01		

Case	Method	MEC (x1000)	Sdev (x1000)	BEC (x1000)	WEC (x1000)	Statistical analysis ($\alpha = 0.05$) significant difference	p	non-sign. difference	p
O1	SA	2 406	390.0	1 948	3 178				
	RS	>3 800	0.0	>3 800	>3 800	no analysis performed			
	GA-CX	>3 800	0.0	>3 800	>3 800				
O2	SA	192	25.7	152	228				
	RS	>760	0.0	>760	>760	no analysis performed			
	GA-CX	>745	47.4	610	>760				
O3	SA	469	84.6	291	585				
	RS	>950	0.0	>950	>950	no analysis performed			
	GA-CX	>950	0.0	950	>950				
O4	SA	1 306	281.8	668	1 805				
	RS	>2 185	0.0	>2 185	>2 185	no analysis performed			
	GA-CX	>2 185	0.0	>2 185	>2 185				

the '>' sign combined with an evaluation counter indicates that the real evaluation counter is higher than the given value; this is caused by the fact that the maximum allowed number of evaluations had to be taken as an estimate of the evaluation counter for some of the runs; this occured when the transport sequence with TM was not yet found when the algorithm was stopped

Table 2. Influence of the *population size* on the efficiency of Simulated Annealing (temperature 120 and decay factor 0.999).

Case	Population size	FTM	MEC (x1000)	Sdev (x1000)	BEC (x1000)	WEC (x1000)	Statistical analysis ($\alpha = 0.05$) significant difference	p	non-sign. difference	p
O1	20	10	2 406	39.0	1 948	3 178	2< 5	<0.01		
	5	10	789	250.0	459	1 236	2<20	<0.01		
	2	10	435	99.5	313	615	5<20	<0.01		
O2	20	10	192	25.7	152	228	2< 5	<0.01		
	5	10	61	14.3	41	87	2<20	<0.01		
	2	10	28	4.7	20	34	5<20	<0.01		
O3	20	10	469	84.6	292	585	2< 5	<0.01		
	5	10	165	38.3	120	233	2<20	<0.01		
	2	10	99	35.5	50	150	5<20	<0.01		
O4	20	10	1 306	281.8	668	1 805	2<20	<0.01	2<5	0.08
	5	10	421	151.0	246	709	5<20	<0.01		
	2	10	307	114.3	115	510				

Group C. Combination of output and input benches

Table 3. Effectiveness of the *local search methods* Simulated Annealing (temperature 120 and decay factor 0.99995), Random Search and the Genetic Algorithm with cycle crossover at Population Size 20 after 3 800 000 evaluations.

Case	Method	FTM	MM	Sdev	BM	WM	Statistical analysis ($\alpha = 0.05$) significant difference	p	non-sign. difference	p
C1	SA	0	266.3	4.72	+7	+22	SA<RS	<0.01		
	RS	0	288.5	9.64	+18	+44	SA<GA	<0.01		
	GA-CX	0	334.8	38.05	+22	+132	RS<GA	<0.01		

Table 4. The best results of Simulated Annealing with temperature 120 and decay factor 0.999 (0.9995 for Case C1) after a large number of evaluations.

Case	Population size	FTM	MEC (x1000)	Sdev (x1000)	BEC (x1000)	WEC (x1000)	Statistical analysis ($\alpha = 0.05$) significant difference	p	non-sign. difference	p
C1	20	0	>3 800	0.0	>3 800	>3 800	no analysis performed			
C2	20	10	893	255.5	540	1 240	no analysis performed			
C3	20	10	6 220	1 811.4	3 659	9 912	no analysis performed			
C4	5	3	>3 164	746.3	1 691	>3 600	no analysis performed			

Table 5. Influence of the *population size* on the effectiveness and efficiency of Simulated Annealing (with temperature 120 and decay factor 0.999 or 0.99995 for Case C1).

Case	Population size	FTM	MM	Sdev	BM	WM	Statistical analysis (α = 0.05) significant difference	p	non-sign. difference	p
C1	20	0	515.2	14.63	+236	+313	2<20	<0.01	2<5	0.14
	5	0	280.4	10.02	+12	+47	5<20	<0.01		
	2	0	272.1	13.45	+4	+40				
C2	20	10	181.0	0.00	+0	+0				
	5	10	181.0	0.00	+0	+0	no analysis performed			
	2	10	181.0	0.00	+0	+0				
C3	20	0	494.5	28.66	+168	+252	2<20	<0.01	2<5	0.12
	5	0	297.9	13.69	+1	+49	5<20	<0.01		
	2	3	290.3	5.1	+0	+17				
C4	20	0	641.5	19.73	+225	+316	2<20	<0.01		
	5	0	384.2	13.03	+5	+50	5<20	<0.01		
	2	0	372.5	10.33	+5	+37	2< 5	0.04		

Case	Population size	MEC (x1000)	Sdev (x1000)	BEC (x1000)	WEC (x1000)	Statistical analysis (α = 0.05) significant difference	p	non-sign. difference	p
C1	20	>700	0.0	>700	>700				
	5	>700	0.0	>700	>700	no analysis performed			
	2	>700	0.0	>700	>700				
C2	20	893	255.5	540	1 240	2<20	<0.01	2<5	0.12
	5	363	170.4	144	667	5<20	<0.01		
	2	248	138.8	72	543				
C3	20	>800	0.0	>800	>800				
	5	>800	0.0	>800	>800	no analysis performed			
	2	>727	131.9	472	>800				
C4	20	>900	0.0	>900	>900				
	5	>900	0.0	>900	>900	no analysis performed			
	2	>900	0.0	>900	>900				

Group M. More than one output bench per crop batch

Table 6. Effectiveness and efficiency of the *local search methods* Simulated Annealing (temperature 120 and decay factor 0.999), Random Search and the Genetic Algorithm with cycle crossover at Population Size 20.

Case	Method	FTM	MM	Sdev	BM	WM	Statistical analysis ($\alpha = 0.05$) significant difference	p	non-sign. difference	p
M1	SA	10	169.0	0.47	+0	+2	SA<RS	<0.01		
	RS	3	171.0	1.15	+1	+5	SA<GA	<0.01		
	GA-CX	0	175.8	3.68	+3	+13	RS<GA	<0.01		
M3	SA	10	186.0	0.00	+0	+0				
	RS	10	186.0	0.00	+0	+0	no analysis performed			
	GA-CX	8	186.2	0.42	+0	+1				
M4	SA	10	235.0	0.00	+0	+0	SA<RS	<0.01	RS<GA	0.11
	RS	3	235.7	0.48	+0	+1	SA<GA	0.02		
	GA-CX	3	237.2	2.82	+0	+9				

Case	Method	MEC (x1000)	Sdev (x1000)	BEC (x1000)	WEC (x1000)	Statistical analysis ($\alpha = 0.05$) significant difference	p	non-sign. difference	p
M1	SA	168	65.6	125	343	SA<RS	<0.01	RS<GA	0.08
	RS	>607	263.9	75	>760	SA<GA	<0.01		
	GA-CX	>760	0.0	>760	>760				
M3	SA	113	5.2	106	123	RS<SA	<0.01	SA<GA	0.83
	RS	38	31.4	19	123			RS<GA	0.13
	GA-CX	>124	169.0	9	>380				
M4	SA	140	6.6	133	155	SA<RS	<0.01	RS<GA	0.41
	RS	>315	113.1	79	>380	SA<GA	<0.01		
	GA-CX	>348	54.5	228	>380				

Table 7. Influence of the *population size* on the efficiency of Simulated Annealing (temperature 120 and decay factor 0.999).

Case	Population size	FTM	MEC (x1000)	Sdev (x1000)	BEC (x1000)	WEC (x1000)	Statistical analysis ($\alpha = 0.05$) significant difference	p	non-sign. difference	p
M1	20	10	168	65.6	125	343	5<20	<0.01	2<20	0.74
	5	10	75	37.3	31	134	5< 2	0.04		
	2	10	155	109.9	8	313				
M3	20	10	113	5.2	106	123	2< 5	<0.01		
	5	10	28	2.1	25	32	2<20	<0.01		
	2	10	11	4.1	7	19	5<20	<0.01		
M4	20	10	140	6.6	133	155	2< 5	<0.01		
	5	10	45	15.5	26	80	2<20	<0.01		
	2	10	14	4.3	10	22	5<20	<0.01		

Table 8. Influence of the *population size* on the effectiveness and efficiency of Random Search.

Case	Population size	FTM	MM	Sdev	BM	WM	Statistical analysis ($\alpha = 0.05$) significant difference	p	non-sign. difference	p
M1	20	3	171.0	1.15	+1	+5	20<5	0.01	20<2	0.06
	5	2	173.4	2.17	+2	+7			2<5	0.43
	2	2	172.6	2.22	+1	+7				
M3	20	10	186.0	0.00	+0	+0	20<5	0.04	5<2	0.14
	5	6	186.5	0.71	+0	+2	20<2	0.01		
	2	3	187.2	1.23	+0	+4				
M4	20	3	235.7	0.48	+0	+1			20< 2	0.11
	5	5	235.6	0.70	+0	+2			5<20	0.72
	2	1	237.9	4.07	+0	+13			5< 2	0.10

Case	Population size	MEC (x1000)	Sdev (x1000)	BEC (x1000)	WEC (x1000)	Statistical analysis ($\alpha = 0.05$) significant difference	p	non-sign. difference	p
M1	20	>607	263.9	75	>760			20<5	0.71
	5	>649	234.4	167	>760			20<2	0.50
	2	>682	224.6	46	>760			5<2	0.75
M3	20	38	31.4	19	123	20<5	0.01	5<2	0.27
	5	>186	150.2	6	>320	20<2	<0.01		
	2	>255	120.6	2	>320				
M4	20	>315	113.1	79	>380	5<2	0.03	5<20	0.40
	5	>262	155.1	21	>380			20< 2	0.09
	2	>380	0.7	378	>380				

Table 9. Influence of the *population size* on the effectiveness and efficiency of the Genetic Algorithm with cycle crossover.

Case	Population size	FTM	MM	Sdev	BM	WM	Statistical analysis ($\alpha = 0.05$) significant difference	p	non-sign. difference	p
M1	40	0	175.1	2.92	+4	+12			10<20	0.62
	20	0	175.8	3.68	+3	+13			10<40	0.95
	10	0	175.0	3.43	+3	+12			40<20	0.64
M3	40	4	186.9	0.88	+0	+2	20<40	0.04	20<10	0.28
	20	8	186.2	0.42	+0	+1			10<40	0.50
	10	7	186.6	1.07	+0	+3				
M4	40	2	238.5	4.84	+0	+16			20<40	0.47
	20	3	237.2	2.82	+0	+9			20<10	0.62
	10	3	238.1	4.79	+0	+15			10<40	0.85

Case	Population size	MEC (x1000)	Sdev (x1000)	BEC (x1000)	WEC (x1000)	Statistical analysis ($\alpha = 0.05$) significant difference	p	non-sign. difference	p
M1	40	>720	0.0	>720	>720				
	20	>720	0.0	>720	>720	no analysis performed			
	10	>720	0.0	>720	>720				
M3	40	>243	152.5	26	>360			20<40	0.12
	20	>120	169.0	9	>360			20<10	0.60
	10	>163	158.9	6	>360			10<40	0.27
M4	40	>326	87.7	82	>360			10<20	0.20
	20	>334	54.5	228	>360			10<40	0.49
	10	>294	116.8	21	>360			40<20	0.51

Table 10. Influence of the *decay factor* on the efficiency of Simulated Annealing (temperature 120) at Population Size 2.

Case	Decay factor	FTM	MEC (x1000)	Sdev (x1000)	BEC (x1000)	WEC (x1000)	Statistical analysis ($\alpha = 0.05$) significant difference	p	non-sign. difference	p
M1	0.995	9	>83	36.2	21	>140			995<9995	0.51
	0.999	10	155	109.9	8	313			995< 999	0.07
	0.9995	10	102	82.9	13	246			999<9995	0.24
M3	0.95	10	6	3.0	2	12	95< 999	0.01	95< 995	0.63
	0.995	10	7	4.7	3	18	95<9995	<0.01	995< 999	0.07
	0.999	10	11	4.1	7	19	995<9995	<0.01		
	0.9995	10	14	3.0	12	21	999<9995	0.04		
M4	0.95	7	>11	2.9	5	>14	95< 999	0.04	95< 995	0.07
	0.995	10	24	20.5	5	61	95<9995	<0.01	999< 995	0.18
	0.999	10	14	4.4	10	22	999<9995	0.02	995<9995	0.88
	0.9995	10	25	11.4	13	46				

Table 11. Influence of the *temperature* on the efficiency of Simulated Annealing (decay factor 0.999) at Population Size 2.

Case	Temperature	FTM	MEC (x1000)	Sdev (x1000)	BEC (x1000)	WEC (x1000)	Statistical analysis ($\alpha = 0.05$) significant difference	p	non-sign. difference	p
M1	60	10	82	61.8	15	232			180<120	0.11
	120	10	155	109.9	8	313			180< 60	0.91
	180	10	78	93.6	14	329			60<120	0.09
M3	60	10	8	2.6	5	13			60<120	0.06
	120	10	11	4.1	7	19			60<180	0.052
	180	10	10	1.9	7	14			180<120	0.48
M4	60	10	16	4.0	10	22			120<180	0.20
	120	10	14	4.4	10	22			120< 60	0.50
	180	10	17	5.1	10	26			60<180	0.45

Table 12. Influence of the *Tabu Search-swap operator* (TS) and the tabu list size (L) on the performance of Simulated Annealing (temperature 120 and decay factor 0.999) at Population Size 2.

Case	Swap method/ Tabu list size (L)	FTM	MEC (x1000)	Sdev (x1000)	BEC (x1000)	WEC (x1000)	Statistical analysis (α = 0.05) significant difference	p	non-sign. difference	p
M1	Normal/0	10	155	109.9	8	313	L3<L0	0.01	L3< L7	0.16
	TS/3	10	46	18.4	15	75	L11<L0	0.03	L3<L11	0.40
	TS/7	10	82	76.0	16	207			L11< L7	0.51
	TS/11	10	62	56.5	17	189			L7< L0	0.11
M3	Normal/0	10	11	4.1	7	19			L7<L11	0.47
	TS/3	10	10	3.7	7	20			L7< L3	0.44
	TS/7	10	9	1.1	7	11			L7< L0	0.18
	TS/11	10	9	1.9	6	12			L11< L3	0.73
									L11< L0	0.34
									L3< L0	0.60
M4	Normal/0	10	14	4.4	10	22			L7< L0	0.54
	TS/3	10	26	19.6	10	74			L7<L11	0.30
	TS/7	10	13	4.3	8	20			L7< L3	0.06
	TS/11	10	16	7.8	9	33			L0<L11	0.54
									L0< L3	0.08
									L11< L3	0.16

Table 13. Influence of the *Tabu Search-swap operator* (TS) and the tabu list size on the effectiveness of Random Search at Population Size 5.

Case	Swap method/ Tabu list size (L)	FTM	MM	Sdev	BM	WM	Statistical analysis (α = 0.05) significant difference	p	non-sign. difference	p
M1	Normal/0	2	173.4	2.17	+2	+7			L7<L11	1.00
	TS/3	0	173.9	1.73	+3	+8			L7< L0	0.41
	TS/7	2	172.6	2.01	+1	+8			L7< L3	0.14
	TS/11	2	172.6	2.17	+2	+7			L11< L0	0.42
									L11< L3	0.16
									L0< L3	0.58
M3	Normal/0	6	186.5	0.71	+0	+2	L3<L11	0.02	L3< L7	0.45
	TS/3	8	186.2	0.42	+0	+1			L3< L0	0.26
	TS/7	7	186.4	0.70	+0	+2			L7< L0	0.76
	TS/11	3	186.7	0.48	+0	+1			L7<L11	0.28
									L0<L11	0.47
M4	Normal/0	5	235.6	0.70	+0	+2			L11< L0	0.28
	TS/3	4	235.8	0.79	+0	+2			L11< L7	0.26
	TS/7	5	235.7	0.95	+0	+3			L11< L3	0.10
	TS/11	7	235.3	0.48	+0	+1			L0< L7	0.79
									L0< L3	0.55
									L7< L3	0.80

Case	Swap method/ Tabu list size (L)	MEC (x1000)	Sdev (x1000)	BEC (x1000)	WEC (x1000)	Statistical analysis (α = 0.05) significant difference	p	non-sign. difference	p
M1	Normal/0	>649	234.4	167	>760			L0<L11	0.93
	TS/3	>760	0.0	>760	>760			L0< L7	0.77
	TS/7	>677	175.8	334	>760			L0< L3	0.15
	TS/11	>658	231.9	67	>760			L11< L7	0.84
								L11< L3	0.18
								L7< L3	0.15
M3	Normal/0	>186	150.2	6	>320	L3<L11	0.03	L3< L7	0.32
	TS/3	>79	127.6	5	>320			L3< L0	0.11
	TS/7	>140	134.5	8	>320			L7< L0	0.48
	TS/11	>228	148.4	6	>320			L7<L11	0.18
								L0<L11	0.54
M4	Normal/0	>262	155.1	21	>380			L11< L7	0.56
	TS/3	>296	129.2	45	>380			L11< L0	0.26
	TS/7	>227	170.5	28	>380			L11< L3	0.09
	TS/11	>184	149.4	15	>380			L7< L0	0.63
								L7< L3	0.32
								L0< L3	0.60

Table 14. Influence of the *crossover method* on the effectiveness and efficiency of the Genetic Algorithm at Population Size 20.

Case	Crossover Method	FTM	MM	Sdev	BM	WM	Statistical analysis (α = 0.05) significant difference	p	non-sign. difference	p
M1	CX	0	175.8	3.68	+3	+13			CX<PMX	0.17
	PMX	0	177.8	2.39	+6	+13			CX< OX	0.55
	OX	0	176.8	3.58	+3	+14			OX<PMX	0.47
M3	CX	8	186.2	0.42	+0	+1	CX< OX	<0.01	OX<PMX	0.75
	PMX	2	187.6	1.65	+0	+5	CX<PMX	0.02		
	OX	2	187.4	1.07	+0	+3				
M4	CX	3	237.2	2.82	+0	+9	CX< OX	<0.01	CX<PMX	0.15
	PMX	1	240.4	6.10	+0	+16	PMX< OX	<0.01		
	OX	0	249.8	6.00	+6	+25				

Case	Crossover Method	MEC (x1000)	Sdev (x1000)	BEC (x1000)	WEC (x1000)	Statistical analysis (α = 0.05) significant difference	p	non-sign. difference	p
M1	CX	>760	0.0	>760	>760				
	PMX	>760	0.0	>760	>760	no analysis performed			
	OX	>760	0.0	>760	>760				
M3	CX	>124	169.0	9	>380	CX< OX	0.01	OX<PMX	0.80
	PMX	>330	112.9	47	>380	CX<PMX	0.01		
	OX	>315	136.2	46	>380				
M4	CX	>348		228	>380			CX< OX	0.08
	PMX	>375		>330	>380			CX<PMX	0.15
	OX	>380		>380	>380			PMX<OX	0.33

Summary

Drawing up internal transport schedules in pot plant production is a very complex task. The amount of internal transport needed in pot plant nurseries is already high at the moment and this will continue to grow in the near future as a result of several external and internal developments. External developments that relate to the pot plant nursery include, for example, the change from a production driven push-market towards a customer driven pull-market and the fact that individual pot plant nurseries have to operate more and more as a part of larger production chains. These developments require a very flexible and responsive pot plant production system. Technical developments in the pot plant nursery itself also make the production and internal transport process more complex. Examples include the new Information and Communication Technologies (ICT) that lead to more data on the internal transport process and potentially to more information for executing controlling tasks. Another example is the use of vision systems for the automated sorting of plants. Here the sorting operation easily distinguishes between quality classes, and results in more small units of pot plants with their specific characteristics having to be traced and transported through the pot plant nursery. Developments in robot or related technology make it possible to treat small production units individually. These robots or automated machines require instructions from an electronic planning system. Another development is the continuous growth in size of the enterprises with an increase in the amount of internal transport. Therefore, it is essential to study and develop a new approach and methods that can support the grower during the internal transport planning process. Scheduling internal transport at the operational level and providing control on a day-to-day or even hour-to-hour basis in particular requires a new approach.

The main components that influence internal transport in pot plant nurseries were studied and described. These components include the layout - including the working area, paths and growing area - the production system, the internal transport device and the internal transport process. A company survey of eleven pot plant nurseries showed the dimensions of these components in practice. The description was used to choose a pot plant nursery with a specific configuration of internal transport components. This was studied more thoroughly and was used as the basis for the development of a new control approach. First, a *layout* was chosen where there was a separation between the working area and the growing area. This has major advantages as far as working conditions are concerned and raises possibilities for mechanization. However, this type of layout requires more internal transport than a

layout in which work and plant growth are mainly combined in one area. The *transportable bench production system* was chosen because it is one of the two most important contemporary production systems and is expected to remain so for some time. The transportable bench has the double function of being both a production system and a transport system. In the other important production system that was not chosen for this study, concrete floors, it is usually more difficult to transport pot plants, because pot plants first have to be lifted from the floor and then put on a transport device before they can be moved. Therefore, pot plants usually remain in the same position on a concrete floor as long as possible to avoid additional handling and transport operations. The transportable bench production system is highly mechanized. Transportable benches can easily be moved between the working area and the growing area and vice versa, and many handling systems have been developed to automate operations in the working area. However, the internal transport between the working area and the growing area has to be sequenced carefully in order to avoid inefficient transport movements with transportable benches that may cause disturbances in the production or delivery of pot plants. The main internal transport device for transportable benches chosen in this study was the *Automatic Guided Vehicle (AGV)*. The combination of an AGV with the transportable bench production system has important advantages for the control of the internal transport process. An AGV can be combined with an automatic data recording system and it can be given transport assignments automatically by software on a Personal Computer. This will make it easier to implement new control approaches in practice. The *internal transport process* is strongly influenced by the potting, spacing, sorting and harvesting operations carried out on pot plants. Operations always require the movement of transportable benches between the working area and the growing area. In the growing area a *Last In First Out (LIFO) system* of access to the rows was chosen. LIFO gives a complex scheduling situation that requires special attention when the transport sequence of transportable benches has to determined.

A *hierarchical planning approach* based on Anthony's framework was presented in this study to decompose the internal transport planning problem of pot plant nurseries into three planning levels with differentiating characteristics: strategic, tactical and operational planning. Emphasis was put on the *operational planning level* within the constraints of the tactical and strategic planning level. The operational planning level is concerned with short-term control decisions on the work sequence, the internal transport sequence and the parking positions of transportable benches. The work sequence was not considered in the context of this study: it was assumed to be fixed. In a company survey of eleven innovative pot plant nurseries it was found that at the moment growers do not make an internal transport sequence on a daily basis. They use *rules of thumb* to determine the positions of transportable benches and the internal transport sequence. These rules of thumb have been described and they have been used as a basis for the next phase of research. Their individual strength has been demonstrated with simulation experiments and they have been integrated with new optimization techniques - local search methods - in order to find good internal transport sequences. A detailed analysis of the transport data of a practical pot plant nursery showed that relocation movements with obstructing transportable benches constituted between 61% and 77% of the total amount of transport movements per week. In general the percentage of relocation movements should be kept as low as possible, although some amount of relocation movements might be unavoidable. Still, in an ideal internal transport situation only input and output movements should be performed. However, in practice, this can only be achieved at the cost of high investments in extra free space in the form of extra buffer rows or more rows in the growing area for example.

A discrete *transport simulation model TRANSIM* was designed and developed to analyse and evaluate the rules of thumb on internal transport used by growers in everyday life. The simulation model incorporated operations that included potting, spacing, selling and internal transport. Recorded data from the pot plant nursery studied were used as input for the simulation model. These input data included potting, spacing and selling dates and the starting positions of all transportable benches in the growing compartment. TRANSIM displays an animation of a growing compartment layout with rows with a LIFO-system access. An animation facility is used to visually represent the flow of transport benches during a simulation run. This animation proved to be very useful in the development and validation phase of the simulation model and also in the phase when the internal transport experiments were performed using different rules of thumb. The simulated rules of thumb focused on determining the parking rows where input benches and relocated obstructing transportable benches should be placed. The transport sequence of the input and output benches was determined on the basis of data recorded at the pot plant nursery and on the positions of the transportable benches in the rows. The simulation experiments showed clear differences between the different rules of thumb as far as the amount of relocation time required during a fixed amount of days was concerned. The best performance was shown by the rule of thumb that took into account the current contents of a parking row. Rows that contain transportable benches that have to be transported in the near future were avoided as parking rows in this rule. The worst results were found for a rule of thumb that always used the most empty row as a parking row.

A *simplified version of the internal transport planning problem* was described, because the real-scale problem is extremely complex. The real-scale internal transport planning problem was classified as a single machine shop scheduling problem. The operational planning problem with respect to internal transport was divided into three sub-problems that deal with planning the work sequence, internal transport sequence and parking positions. The *work sequence* was taken as a fixed input for the *transport sequence* sub-problem. This was solved iteratively in combination with the choice of the *parking positions*. Local search methods were used to generate acceptable transport sequences and the best rules of thumb in the simulation experiments were taken as the basis for developing two Parking Methods that determine the parking positions of relocated transportable benches and input benches. The required number of transport movements was taken as an evaluation value - performance measure - for the quality of the combination of the transport sequence and the parking positions. The variable costs of internal transport at the operational planning level are proportional to the number of transport movements and the fixed costs are determined by the results from strategic and tactical planning which are no variable in this study. A method was developed to determine the *theoretical minimum* number of transport movements (TM) needed to perform a certain internal transport sequence. The value of TM was then used as a bench mark for comparing local search and parking methods.

Four *local search methods* - Simulated Annealing (SA), Genetic Algorithm (GA), Tabu Search (TS) and Random Search (RS) - were described and applied to generate good internal transport sequences in combination with a chosen parking method. These local search methods were implemented using a standard research tool, called SUGAL. First, a pre-selection was made of the local search methods and their parameters using *simple test cases*. These experiments showed that it is possible to obtain transport sequences with the theoretical minimum number of transport movements for these simple test cases using the local search methods Genetic Algorithm, Simulated Annealing and Random Search. Tabu Search was not studied for these simple test cases. The most effective local search method proved to be Simulated Annealing. It was remarkable that Random Search always performed better than the Genetic Algorithm combined with the cycle crossover method. The results of the Genetic Algorithm were influenced by the chosen crossover method: the cycle crossover method (CX) was superior to partially mapped crossover (PMX) and order crossover (OX). The swap mutation method in combination with the Genetic Algorithm showed the best results at a mutation rate of 0.5-1.0. Parking Method 2 - first available parking row without output benches - that was derived from the best rule of thumb in the simulation experiments, showed better results than the alternative Parking Method 1 - first available parking row -. Therefore, only this best parking method was used in the next phase of the experiments.

In the next phase of the research, the performance of local search methods as applied to *real-scale test cases* was studied. Three groups of real-scale test cases were again derived from the practical data recorded at the pot plant nursery being studied. These groups represented three different planning situations. First, a situation where a transport sequence had to be generated for output benches only. Second a transport sequence for a combination of output benches and input benches and third for crop batches with more than one output bench. Two types of evaluation criteria were used to judge the performance of the local search methods: *effectiveness* and *efficiency*. A statistical analysis was performed to check whether differences between two local search methods or between two different parameter settings of the same local search method were statistically significant. The analysis focused on the effect of the local search method, population size, the cooling schedule of Simulated Annealing (decay factor and temperature), the effect of the addition of a Tabu Search swap mutation operator to Simulated Annealing and Random Search and, finally, the crossover method of the Genetic Algorithm. It was found that Simulated Annealing was significantly more effective than the local search methods Random Search and the Genetic Algorithm with the cycle crossover method in all three groups of test cases. Population size influenced efficiency depending on the type of local search method used. At the same allowed number of evaluations, Simulated Annealing performed better with a small Population Size 2, while Random Search performed better at a large Population Size 20. Simulated Annealing needed a long period to ‘cool down’ and so it benefited from less solution strings in the population, developing for a larger number of evaluations. The Genetic Algorithm did not seem to be influenced by population size. The cooling schedule only partly influenced the efficiency of Simulated Annealing. The decay factor of the cooling schedule showed a statistically significant effect, but the initial temperature did not. Adding the Tabu Search-swap mutation operator did improve the efficiency of Simulated Annealing in one case. However, it did not improve the effectiveness or efficiency of Random Search at all. Finally the crossover method did influence the effectiveness and efficiency of the Genetic Algorithm. The performance of the cycle crossover method was better than the performance of partially mapped crossover and order crossover method.

The performance results of the local search methods applied to real-scale test cases show that Simulated Annealing is the most effective and efficient method of finding good internal transport sequences in combination with a relatively simple parking method. Transport sequences with a theoretical minimum number of transport movements were frequently obtained by Simulated Annealing for these real-scale test cases. However, the other local search methods Genetic Algorithm, Tabu Search and Random Search also produced relatively good transport sequences in many calculation runs. In practice, the internal transport control problems will vary between different pot plant nurseries and therefore, the best approach will be to use a combination of these local search methods in order to be prepared for all sorts of internal transport control situations. The performance results of the local search methods also indicate that the approach of combining a sophisticated transport sequence generating technique with a relatively simple parking method is a successful one. This approach has the advantage that other pot plant nurseries will only have to specify one or more simple company-specific parking method based on their rules of thumb and that the generic transport sequence generating technique based on the local search methods can remain the same. This will enable tailor-made solution methods for other pot plant nurseries that involve very low additional development efforts. Now that the described control approach has been proven successful, a completely automatic support system for internal transport in pot plant nurseries lies within reach. However, this still requires the implementation of some further extensions.

Samenvatting

Het opstellen van een intern transportplan bij de potplantenteelt is een zeer complexe taak. De benodigde hoeveelheid intern transport in potplantenbedrijven is op het moment al hoog en zal in de nabije toekomst nog verder groeien onder invloed van verschillende externe en interne ontwikkelingen. Externe ontwikkelingen die in verband staan met een potplantenbedrijf zijn bijvoorbeeld de verandering van een productiegedreven push-markt naar een consumentengedreven pull-markt en het feit dat individuele potplantenbedrijven steeds meer moeten opereren als een onderdeel van grotere productieketens. Deze ontwikkelingen vereisen een productiesysteem voor potplanten dat zeer flexibel is en snel kan reageren. Technische ontwikkelingen binnen het potplantenbedrijf zelf maken het productie- en transportproces nog complexer. Een voorbeeld is de nieuwe Informatie- en CommunicatieTechnologie (ICT) die leidt tot meer gegevens over het intern transport proces en potentieel tot meer informatie ten behoeve van het uitvoeren van beheerstaken. Een ander voorbeeld is het gebruik van beeldverwerkings-technieken voor het automatisch sorteren van planten. Hierbij maakt de sorteerhandeling makkelijk onderscheid tussen kwaliteitsklassen en dit resulteert in meer en kleinere eenheden van potplanten met hun eigen specifieke karakteristieken, eenheden die gevolgd en getransporteerd moeten worden binnen het potplantenbedrijf. Ontwikkelingen op het gebied van robots en gerelateerde technologie maken het mogelijk om kleine productie-eenheden individueel te behandelen. Deze robots of geautomatiseerde machines vragen instructies van een geautomatiseerde planningsysteem. Een andere ontwikkeling is de continue groei van de grootte van de ondernemingen die gepaard gaat met een groei van de hoeveelheid intern transport. Daarom is het essentieel nieuwe methoden te bestuderen en te ontwikkelen, die de tuinder kunnen ondersteunen gedurende het planningsproces van het intern transport. De planning van het intern transport op het operationele niveau en vooral het realiseren van beheersing (control) op een dag-tot-dag of zelfs uur-tot-uur niveau, vereist een nieuwe aanpak.

De belangrijkste componenten die het intern transport in potplantenkwekerijen beïnvloeden, zijn bestudeerd en beschreven. Deze componenten zijn de layout (werkruimte, paden en teeltruimtes), het teeltsysteem, de interne transportmiddelen en het intern transportproces. Een inventarisatie bij elf potplantenbedrijven leverde de dimensies van deze componenten in de praktijk op. De beschrijving werd gebruikt om een potplantenbedrijf te kiezen met een specifieke configuratie van de componenten van het intern transport. Deze configuratie werd grondig bestudeerd

en vormde de basis voor een nieuwe beheersaanpak. Allereerst werd een *layout* gekozen met een scheiding tussen de werkruimte en de teeltruimte. Deze layout heeft belangrijke voordelen voor de werkomstandigheden en verhoogt de mogelijkheden voor mechanisatie. Dit type layout vraagt echter meer intern transport dan een layout waarbij werk en plantengroei voornamelijk in één ruimte worden gecombineerd. Het *rolcontainerteeltsysteem* werd gekozen omdat het op het moment één van de twee belangrijkste teeltsystemen is en dit naar verwachting de komende tijd ook zal blijven. Een rolcontainer heeft een dubbele functie omdat hij zowel de rol van teeltsysteem als die van transportsysteem vervult. Bij het andere belangrijke teeltsysteem, dat niet in deze studie werd gekozen nl. betonvloeren, is het meestal moeilijker om potplanten te transporteren, omdat de potplanten eerst van de vloer moeten worden getild en vervolgens op een transportmiddel moeten worden geplaatst voordat ze kunnen worden verplaatst. Daarom blijven potplanten meestal zo lang mogelijk op dezelfde positie op een betonvloer staan om zo extra bewerkings- en transporthandelingen te voorkomen. Het rolcontainerteeltsysteem is sterk gemechaniseerd. Rolcontainers kunnen gemakkelijk worden getransporteerd tussen de werkruimte en de teeltruimte en vice versa, en veel systemen zijn ontwikkeld om handelingen te automatiseren in de werkruimte. De volgorde van het transport tussen de werkruimte en de teeltruimte moet echter zorgvuldig worden gepland om inefficiënte transportbewegingen met rolcontainers te voorkomen. Die bewegingen zouden verstoringen van de productie of van het afleveren van potplanten kunnen veroorzaken. Het belangrijkste transportmiddel voor rolcontainers dat in deze studie werd gekozen is de *Automatische Wagen* (AW). De combinatie van een AW met het rolcontainerteeltsysteem heeft belangrijke voordelen voor de beheersing van het intern transportproces. Een AW kan worden gecombineerd met een geautomatiseerd systeem voor gegevensregistratie en een AW kan zijn transportopdrachten automatisch ontvangen van software op een bedrijfscomputer. Dit zal het makkelijker maken om een nieuwe beheersaanpak in de praktijk te implementeren. Het *intern transport proces* wordt sterk beïnvloed door de handelingen oppotten, wijderzetten, sorteren en oogsten, die met de potplanten gebeuren. Handelingen vragen altijd transport van rolcontainers tussen de werkruimte en de teeltruimte. In de teeltruimte werd een *Last In First Out (LIFO) systeem* gekozen voor de toegang tot de rijen. LIFO levert een complexe planningsituatie, waarbij speciale aandacht moet worden gegeven aan het bepalen van de transportvolgorde van de rolcontainers.

In deze studie wordt een *hiërarchische planningsbenadering* gepresenteerd, die gebaseerd is op Anthony's framework en die wordt gebruikt om een decompositie van het intern transportprobleem te bereiken in drie planningslagen met onderscheidende karakteristieken: strategische, tactische en operationele planning. De nadruk is gelegd op het operationele planningsniveau binnen de randvoorwaarden van het tactische en het strategische planningsniveau. Het operationele planningsniveau richt zich op korte termijnbeslissingen ten aanzien van de werkvolgorde, de intern transportvolgorde en de parkeerposities van de rolcontainers. De werkvolgorde is niet in beschouwing genomen in de context van deze studie: er is een vaste werkvolgorde aangenomen. Uit een inventarisatie bij elf innovatieve potplantenbedrijven is gebleken dat kwekers op het moment nog geen dagelijkse planning maken voor de intern transportvolgorde. Ze gebruiken *vuistregels* om de posities van rolcontainers en de intern transportvolgorde te bepalen. Deze vuistregels zijn in dit onderzoek beschreven en ze zijn gebruikt als basis voor de volgende fase van het onderzoek. Hun individuele kracht is aangetoond met simulatie-experimenten en ze zijn geïntegreerd met nieuwe optimalisatietechnieken (lokale zoekmethoden) om goede intern transportvolgordes te vinden. Een gedetailleerde analyse van de transportgegevens van een potplantenbedrijf toonde aan dat omrijbewegingen met blokkerende rolcontainers tussen 61% en 77% van de totale hoeveelheid transportbewegingen per week uitmaken. In het algemeen moet het percentage omrijbewegingen zo laag mogelijk worden gehouden, hoewel een zekere hoeveelheid omrijbewegingen onvermijdbaar zal zijn. In een ideale situatie zouden echter alleen aanvoer- en afvoerbewegingen moeten plaatsvinden. In de praktijk kan dit echter alleen worden bereikt tegen hoge investeringskosten in extra ruimte, bijvoorbeeld in de vorm van extra bufferrijen of meer rijen in de teeltruimte.

Het discrete *transport simulatiemodel TRANSIM* is ontworpen en ontwikkeld om de vuistregels voor het intern transport, die kwekers in de dagelijkse praktijk gebruiken, te analyseren en te evalueren. Het simulatiemodel bevat de handelingen oppotten, wijderzetten, verkoop en intern transport. De geregistreerde gegevens van het bestudeerde potplantenbedrijf zijn gebruikt als invoer van het simulatiemodel. De invoergegevens waren de oppot-, wijderzet- en verkoopdatum en de startposities van alle rolcontainers in de teeltafdeling. TRANSIM toonde een animatie van een teeltafdeling met rijen met een LIFO toegangsysteem. De animatiefaciliteit werd gebruikt om de stroom van rolcontainers gedurende een simulatierun visueel weer te geven. Deze animatie bleek zeer bruikbaar te zijn in de ontwikkel- en validatiefase van het simulatiemodel en tevens in de fase waarin intern transportexperimenten werden uitgevoerd met verschillende vuistregels. De gesimuleerde vuistregels concentreerden zich op het bepalen van de parkeerrijen, waarin de aan te voeren rolcontainers en de omgereden blokkerende rolcontainers moesten worden geplaatst. De transportvolgorde van de aan te voeren en af te voeren rolcontainers werd bepaald op basis van de geregistreerde gegevens van het potplantenbedrijf en op basis van de posities van de rolcontainers in de rijen. De simulatie-experimenten toonden duidelijke verschillen tussen de onderzochte vuistregels voor de benodigde omrijtijd gedurende een vast aantal dagen. De beste prestaties werden geleverd door de vuistregel, die rekening houdt met de actuele inhoud van een parkeerrij. Rijen die rolcontainers bevatten die in de nabije toekomst moeten worden getransporteerd, worden door deze regel vermeden als parkeerrij. De slechtste resultaten werden gevonden voor de vuistregel die steeds de meest lege rij gebruikt als parkeerrij.

Van het intern transport planningsprobleem is een *vereenvoudigde vorm* beschreven, omdat het probleem op werkelijke schaal extreem complex is. Het intern transport planningsprobleem op werkelijke schaal kan geclassificeerd worden als een single machine shop scheduling probleem. Het operationele planningsprobleem voor het intern transport is verdeeld in drie subproblemen die betrekking hebben op het plannen van de werkvolgorde, de intern transportvolgorde en de parkeerposities. De werkvolgorde is als vaste invoer genomen voor het transportvolgorde subprobleem. Dit is iteratief opgelost in combinatie met de keuze van de parkeerposities. Om acceptabele transportvolgordes te genereren zijn lokale zoekmethoden zijn gebruikt. De beste vuistregels uit de simulatie-experimenten zijn als basis genomen voor het ontwerpen van twee parkeermethoden die de parkeerposities van de omgereden en aangevoerde rolcontainers bepalen. Het vereiste aantal transportbewegingen is genomen als evaluatiewaarde (prestatiemaatstaf) voor de kwaliteit van de combinatie van de transportvolgorde en de parkeerposities. De variabele kosten van het intern transport op het operationele planningsniveau zijn evenredig aan het aantal transportbewegingen. De vaste kosten worden bepaald door de resultaten van de strategische en tactische planning. Die resultaten zijn in deze studie niet variabel. Om het *theoretisch minimum* aantal transportbewegingen (TM) te bepalen dat nodig is om een bepaalde intern transportvolgorde uit te voeren, is een methode ontwikkeld. De waarde van TM is gebruikt als maatstaf voor het vergelijken van lokale zoekmethoden en parkeermethoden.

Vier *lokale zoekmethoden*, nl. Simulated Annealing (SA), het Genetisch Algoritme (GA), Tabu Search (TS) en Random Search (RS) zijn beschreven en toegepast om goede intern transportvolgordes te genereren in combinatie met een gekozen parkeermethode. Deze lokale zoekmethoden zijn geïmplementeerd met behulp van een standaard onderzoeksgereedschap, SUGAL genaamd. Allereerst is een voorselectie gemaakt van de lokale zoekmethoden en hun parameters met behulp van *eenvoudige test cases*. Deze experimenten hebben aangetoond dat het voor deze eenvoudige test cases mogelijk is om transportvolgordes te vinden met het theoretisch minimum aantal transportbewegingen met behulp van de lokale zoekmethoden Genetisch Algoritme, Simulated Annealing en Random Search. Tabu Search is nog niet bestudeerd voor deze eenvoudige test cases. De meest effectieve lokale zoekmethode was Simulated Annealing. Het was opvallend dat Random Search steeds betere resultaten gaf dan het Genetisch Algoritme in combinatie met de cycle crossover methode. De resultaten van het Genetisch Algoritme werden beïnvloed door de gekozen crossover methode: de cycle crossover methode (CX) was superieur aan partially mapped crossover (PMX) en order crossover (OX). De swap mutatie methode in combinatie met het Genetisch Algoritme gaf de beste resultaten bij een mutatiesnelheid van 0.5-1.0. Parkeermethode 2 (‘de eerste parkeerrij die beschikbaar is zonder rolcontainers die moeten worden afgevoerd’) die is afgeleid van de beste vuistregel uit de simulatie-experimenten gaf betere resultaten dan de alternatieve Parkeermethode 1 (‘eerste beschikbare parkeerrij’). Daarom is alleen Parkeermethode 2 gebruikt in de volgende fase van de experimenten.

In de volgende fase van het onderzoek zijn de prestaties van de lokale zoekmethoden onderzocht bij het toepassen op *test cases met een reële schaal*. Drie groepen van test cases met een reële schaal zijn wederom afgeleid van de gegevens die in de praktijk zijn geregistreerd bij het bestudeerde potplantenbedrijf. Deze groepen representeerden drie verschillende planningssituaties. Allereerst een situatie, waarbij een transportvolgorde moet worden gegenereerd alleen voor af te voeren rolcontainers. Ten tweede een transportvolgorde voor een combinatie van af te voeren en aan te voeren rolcontainers en ten derde voor partijen met meer dan één af te voeren rolcontainer. Om de prestaties van de lokale zoekmethoden te beoordelen zijn twee typen evaluatiecriteria gebruikt: de *effectiviteit* en de *efficiëntie*. Om te controleren of verschillen tussen twee lokale zoekmethoden of tussen twee verschillende parameterinstellingen van dezelfde lokale zoekmethode statistisch significant waren, is een statistische analyse uitgevoerd. De analyse concentreerde zich op het effect van de lokale zoekmethode, de populatiegrootte, het koelschema van Simulated Annealing (de vervalfactor en de temperatuur), het effect van het toevoegen van een Tabu Search swap mutatie operatie aan Simulated Annealing en Random Search en tenslotte de crossover methode van het Genetisch Algoritme. Gevonden werd dat Simulated Annealing in alle drie de groepen met test cases significant effectiever was dan de lokale zoekmethoden Random Search en het Genetisch Algoritme met de cycle crossover methode. De populatiegrootte beïnvloedde de efficiëntie afhankelijk van het gebruikte type lokale zoekmethode. Bij hetzelfde aantal toegestane evaluaties presteerde Simulated Annealing beter bij een kleine Populatiegrootte 2, terwijl Random Search beter presteerde bij een grote Populatiegrootte 20. Simulated Annealing had een lange tijd nodig om ‘af te koelen’ en daarom had het baat bij minder oplossingen in de populatie, die zich gedurende een groter aantal evaluaties konden ontwikkelen. Het Genetisch Algoritme werd niet beïnvloed door de populatiegrootte. Het koelschema beïnvloedde de efficiëntie van Simulated Annealing slechts ten dele. De vervalfactor van het afkoelschema had een statistisch significant effect, maar de begintemperatuur niet. Het toevoegen van de Tabu Search swap mutatie operatie verbeterde de efficiëntie van Simulated Annealing in één geval. Het verbeterde de effectiviteit en efficiëntie van Random Search echter in het geheel niet. De crossover methode tenslotte beïnvloedde wel de effectiviteit en efficiëntie van het Genetisch Algoritme. De prestaties van de cycle crossover methode waren beter dan de prestaties van de partially mapped crossover en order crossover methode.

De prestaties van de lokale zoekmethoden, toegepast bij test cases met een reële schaal, tonen aan dat Simulated Annealing de meest effectieve en efficiënte methode is om goede intern transportvolgordes te vinden in combinatie met een relatief eenvoudige parkeermethode. Transportvolgordes met een theoretisch minimum aantal transportbewegingen voor deze test cases met een reële schaal werden frequent gevonden door Simulated Annealing. De ander lokale zoekmethoden, het Genetisch Algoritme, Tabu Search en Random Search produceerden echter ook relatief goede transportvolgordes in veel berekeningsronden. In de praktijk zullen intern transport beheersingsproblemen verschillen per potplantenbedrijf en daarom zal de beste aanpak zijn om een combinatie van deze lokale zoekmethoden te gebruiken om voorbereid te zijn op alle soorten intern transport beheersingssituaties. De prestaties van de lokale zoekmethoden geven ook aan dat de aanpak van het combineren van een geavanceerde techniek voor het genereren van transportvolgordes met een relatief eenvoudige parkeermethode succesvol is. Deze aanpak heeft het voordeel dat andere potplantenbedrijven alleen maar één of meer eenvoudige bedrijfsspecifieke parkeermethodes hoeven te specificeren, gebaseerd op hun eigen vuistregels en dat de generieke techniek voor het genereren van de transportvolgordes hetzelfde kan blijven. Dit maakt bedrijfsspecifieke oplossingsmethoden mogelijk voor andere potplantenbedrijven tegen lage additionele ontwikkelingsinspanningen. Nu de beschreven beheersingsmethode succesvol is gebleken, ligt een totaal geautomatiseerd ondersteunend systeem voor het intern transport in potplantenbedrijven binnen bereik. Dit vraagt echter eerst nog wel de implementatie van enige uitbreidingen.

Curriculum Vitae

Bert Annevelink werd op 9 juli 1960 geboren in Roosendaal (Noord-Brabant). In 1978 behaalde hij aan het St. Gertrudis Lyceum te Roosendaal het diploma gymnasium B. Door zijn interesse in de natuur (ornithologie) koos hij voor een studie aan de Landbouw Hogeschool te Wageningen. Hier volgde hij, na de N-propaedeuse, de studierichting bosbouw der gematigde luchtstreken. Zijn ervaring met het fenomeen ‘Waldsterben’ tijdens een stage in Zuid-Duitsland vormde de aanleiding van het hoofdonderwerp tijdens zijn doctoraalexamen: een studie naar de verspreiding van bastschaden bij de Amerikaanse eik en hun verband met luchtverontreinigingen. In zijn andere afstudeervakken richtte hij zich op planning in de bosbouw, bestuurlijke informatiekunde en marktkunde. In januari 1985 sloot hij zijn studie in Wageningen af. Vanaf november 1984 tot maart 1986 vervulde hij zijn militaire diensttijd als reserve-officier bij de genie in Vught, Wezep en Seedorf. In mei 1986 kwam hij in dienst bij het IMAG in Wageningen als wetenschappelijk onderzoeker bij de afdeling organisatiekunde. Zijn onderzoek richtte zich in eerste instantie op de optimalisatie van productieplanning in de potplantenteelt. Van 1988 tot 1989 volgde hij de post-doctorale opleiding Agribusiness Management in Wageningen. Naast zijn dagelijkse werkzaamheden vervulde hij van 1986 tot 1991 als mede-oprichter van de Vereniging voor Informatici werkzaam in de Agrarische Sector (VIAS) verschillende bestuursfuncties, zoals penningmeester en redacteur van het tijdschrift Agro Informatica. In 1992 won hij samen met Rob Broekmeulen de prijs voor de beste paper en presentatie tijdens het jaarlijkse VIAS-symposium. Hij was mede-organisator van het internationale XXV CIOSTA-CIGR V congres te Wageningen in 1993. Vanaf 1995 was hij betrokken bij de implementatie van het besturingsmodel bij IMAG-DLO en vervulde hij naast zijn promotie-onderzoek de rol van projectleider, programmaleider en marketeer bij het onderzoekprogramma ‘energie uit biomassa’. Vanaf januari 1999 heeft hij de functie van clusterleider bij de onderzoekcluster systeemkunde binnen de afdeling technologie open teelten van IMAG-DLO.