A two-echelon inventory routing problem for perishable products

Rohmer, S. U. K., Claassen, G. D. H., & Laporte, G.

A Two-Echelon Inventory-Routing Problem for Perishable Products

S.U.K. Rohmer [*1], G.D.H. Claassen [†1], and Gilbert Laporte [‡2]

[1]Operations Research and Logistics, Wageningen University, Hollandseweg 1, 6706 KN Wageningen, Netherlands

[2]CIRRELT and HEC Montreal, 3000 chemin de la Côte-Sainte-Catherine, H3T 2A7 Montréal, Canada

August 7, 2019

Abstract

This paper presents a two-echelon inventory-routing problem for perishable products. Products are delivered from a supplier to an intermediary depot, where storage may occur and from which they are delivered by smaller vehicles to the customer locations. Holding costs are incurred for storage at the depot. Customer availability is taken into account in the form of customer delivery patterns. The objective is to minimise the total transportation and holding costs. We formulate the problem as a mixed integer linear program and solve it by means of an adaptive large neighbourhood search metaheuristic in combination with the solution of a reduced formulation. Three variants of the heuristic are compared on a variety of randomly generated instances. Given the two-stage structure of the problem, computational results show the importance of taking the cost structure into account when choosing the most suitable solution approach.

Keywords: Perishable products; inventory-routing; adaptive large neighbourhood search; last-mile logistics; two-echelon system

[*]sonja.rohmer@wur.nl [†]frits.claassen@wur.nl [‡]gilbert.laporte@cirrelt.ca

1 Introduction

Last-mile logistics and inventory considerations can play a crucial part in supply chain operations. In fact, the last mile is often considered to be one of the most costly and least efficient stages of the whole supply chain (Gevaers et al., 2009). Transporting products to the final customer can be challenging, especially if these are perishable items, with a limited life span, for which the quality degrades over time. Storage time and time spent on the road affect the quality of the products and reduce their life time at the customer location, and may therefore reduce the value of the product or result in product loss. While perishable products can be found in many areas, the food sector presents an important example of an environment in which quality and safety aspects play an important role (Akkerman et al., 2010), and where high perishability leads to considerable losses and wastage (Yu and Nagurney, 2013). In today's competitive markets, the quality and freshness of a product are important aspects influencing the customers' decision to purchase and hence are vital for the survival of a business.

Last-mile distribution often arises in two-echelon systems that require intermediary storage of products, and needs to consider customer availability during a given time horizon, thus complicating the delivery process. Efficient distribution systems and delivery planning for perishable products can therefore help to avoid spoilage, save costs and positively affect the quality of a product. Periodic vehicle routing as well as inventory management at the depot play an important role in this context, by optimising the delivery schedule, the routes, the storage time and the quantity of products at the depot.

1.1 Literature Review

Problems concerned with the optimal routing of vehicles, to improve delivery operations, have been extensively studied for decades (Cordeau et al., 2007, Laporte, 2009). Over the course of time, several variants of the basic vehicle routing problem (VRP) have incorporated other aspects and more specific requirements related to decision making in the supply chain context (Schmid et al., 2013). A number of studies can be found on the issues related to the routing of perishable products. Thus, Tarantilis and Kiranoudis (2001) developed a metaheuristic for the vehicle routing related to the distribution of fresh milk with a fleet of heterogenous vehicles. In the context of fresh vegetable distribution, Osvald and Stirn (2008) included perishability into the vehicle routing problem with time windows and time-dependent travel times. Amorim and Almada-Lobo (2014) developed a multi-objective model for the vehicle routing problem with time windows to investigate different distribution scenarios and the trade-off between cost and product quality. The problem was solved by using the ϵ-method for small instances and by applying an evolutionary algorithms for larger instances. Rabbani et al. (2015) proposed a multi-objective VRP with time windows and customer selection, assuming a heterogenous fleet of vehicles and considering multiple deteriorating products. Wang et al. (2016) solved a multi-objective VRP with time windows and perishability considerations using a two-phase heuristic method based on a variable neighbourhood search and a genetic algorithm. Rabbani et al. (2016) considered the use of multiple middle depots and incorporated several aspects, such as product freshness and profit maximisation into the objective function. They developed a genetic algorithm for the solution of large instances. Considering perishability in a site-dependent vehicle routing problem with time windows and a heterogeneous fleet of vehicles, Amorim et al. (2014) developed a neighbourhood search algorithm and applied it to a real-life case study arising in a Portuguese food distribution company. Hsu et al. (2007) extended the vehicle routing problem with time windows by adding a stochastic cost component related to the perishability of products. Song and Ko (2016) proposed a non-linear model with the objective of maximising customer satisfaction related to the delivery of multi-commodity perishable products with refrigerated and non-refrigerated vehicles. The problem is solved using a priority-based heuristic approach.

A number of extensions exist on the integration of other aspects of the planning process into routing models, such as production, location and inventory decisions. While the focus in the following will be on the latter aspect, examples related to other features can be found in Farahani et al. (2012), Govindan et al. (2014) and the review of Amorim et al. (2013).

Nahmias (2011) and Karaesmen et al. (2011) provide reviews related to the management and modelling of perishable inventory systems. For a more general and extensive overview of the field of inventory-routing problems (IRP), its variants and associated solution approaches we refer to the reviews of Bertazzi et al. (2008), Andersson et al. (2010) and Coelho et al. (2013).

In the context of perishable products, Hiassat and Diabat (2011) proposed an integrated model for a location-inventory-routing problem considering products with a limited life-span. Le et al. (2013) developed an algorithm for an IRP based on column generation and cutting planes, the problem is extended in Hiassat et al. (2017), integrating location decisions into the model, and solved using a genetic algorithm. Coelho and Laporte (2014) applied branch-and-cut to optimally solve the perishable inventory-routing problem (PIRP) under general assumptions and consideration of two different selling policies. Jia et al. (2014) solved an IRP for perishable products with multiple time windows and loading costs, solving the problem using a two-phase solution algorithm. Mirzaei and Seifi (2015) considered the impact of lost sales in their inventory-routing problem. The resulting mixed integer non-linear programming model was solved using a metaheuristic based on simulated annealing and tabu search. Kande et al. (2015) proposed a tabu search metaheuristic for a routing problem with inventory and lot-sizing decisions as well as multiple source nodes. Dealing with uncertain demand in a multi-period IRP model, Soysal et al. (2015) further included environmental aspects in the form of greenhouse gas emissions and fuel consumption. Rahimi et al. (2017) developed a multi-objective model for the IRP of perishable products, allowing for a choice of different vehicles. They incorporated environmental aspects as well as customer satisfaction considerations on top of the traditional cost minimisation. The problem was solved by means of a genetic algorithm. Diabat et al. (2016) proposed a new arc-based formulation and a tabu search algorithm for the inventory-routing problem for perishable products. Azadeh et al. (2016) considered an inventory-routing problem with transshipments for a perishable product and apply a genetic algorithm to solve the problem. Li et al. (2016) developed a mixed integer linear programming model for perishable supply chains, incorporating production decisions in the inventory-routing problem and maximising profit. In addition, Zhao et al. (2008) proposed a similar structured two-echelon inventory routing problem without perishability considerations. The problem was solved using a variable large neighbourhood search. Table 1 provides a summary of the related scientific literature.

Table 1: Summary of the related literature

		Math	Perishable	Inventory		Two-	Multiple	Delivery	Time	Multiple	Multiple
Reference	Algorithm	Model	Products	at customer	at depot	Echelon	Depots	Patterns	Windows	Products	Objectives
Tarantilis and Kiranoudis (2001)	Heuristic		•								
Osvald and Stirn (2008)	Heuristic		•						•		
Amorim and Almada-Lobo (2014)	Heuristic	•	•						•		•
Rabbani et al. (2015)	Heuristic	•	•						•	•	•
Wang et al. (2016)	Heuristic	•	•						•		•
Rabbani et al. (2016)	Heuristic	•	•			•	•		•	•	•
Amorim et al. (2014)	Heuristic	•	•						•	•	
Hsu et al. (2007)	Heuristic	•	•						•		
Song and Ko (2016)	Heuristic	•	•								
Hiassat and Diabat (2011)		•	•	•		•	•				
Le et al. (2013)	Heuristic	•	•	•							
Hiassat et al. (2017)	Heuristic	•	•	•		•	•				
Coelho and Laporte (2014)	Exact	•	•	•							
Jia et al. (2014)	Heuristic	•	•	•					•		
Mirzaei and Seifi (2015)	Heuristic	•	•	•							
Kande et al. (2015)	Heuristic	•	•	•	•		•			•	
Soysal et al. (2015)		•	•	•							
Rahimi et al. (2017)	Heuristic	•	•	•						•	•
Diabat et al. (2016)	Heuristic	•	•	•							
Azadeh et al. (2016)	Heuristic	•	•	•							
Li et al. (2016)		•	•	•							
Zhao et al. (2008)	Heuristic	•		•	•	•					
This paper	Heuristic	•	•		•	•		•			

1.2 Contribution and organisation of this paper

This paper focuses on inventory-routing for perishable products in the context of last-mile city distribution. Given this focus, it is reasonable to assume a multi-level system, resulting in a two-echelon routing problem (Hemmelmayer et al., 2012). A survey of two-echelon routing problems can be found in Cuda et al. (2015). This study, like many others, is about the solution of a last-mile distribution system in a two-echelon setting. However, our problem differs from most existing two-echelon problems in two main ways. First, we consider inventory at the depot and not at the customer locations as is the case in many papers. Second, we are the first to handle multiple delivery patterns in the context of a two-echelon system. Our aim is to introduce the two-echelon perishable inventory-routing problem, model it and solve it heuristically through an adaptive large neighbourhood search (ALNS).

The paper is organised as follows. In Section 2 a formal description of the problem will be given. Section 3 introduces the mathematical formulation of the model, while Section 4 describes the heuristic. Computational results follow in Section 5, and conclusions are presented in Section 6.

2 Formal Problem Description

We consider the inventory-routing problem for perishable products within the context of urban last-mile delivery. We therefore assume a two-echelon system, with a supplier, an intermediary depot and several customer locations. Fresh products are delivered from the supplier to the depot and then stored until delivery occurs. Inventory levels are updated at the beginning of each day, representing a time period. The depot, which belongs to the supplier, receives flower or vegetable deliveries from producers. The customers, on the other hand, are independent and need to be served according to their availability and preferences. The availability of a customer is provided in the form of combinations of visit periods. These delivery day combinations are represented for each location in the form of a list of combinations of daily time windows, during which deliveries can be made, as it is the case in the periodic vehicle routing problem (PVRP) presented by Cordeau et al. (1997). An example of this would be a customer needing to be supplied with fresh produce every two days and delivery could take place either on days 1, 3, 5 and 7 or on days 2, 4 and 6. The rationale is that customers will receive a new order of products every couple of days to guarantee freshness of the product and fulfill the customer demand. This means that the departures of the vehicles need to be scheduled according to the delivery time windows at the customer location.

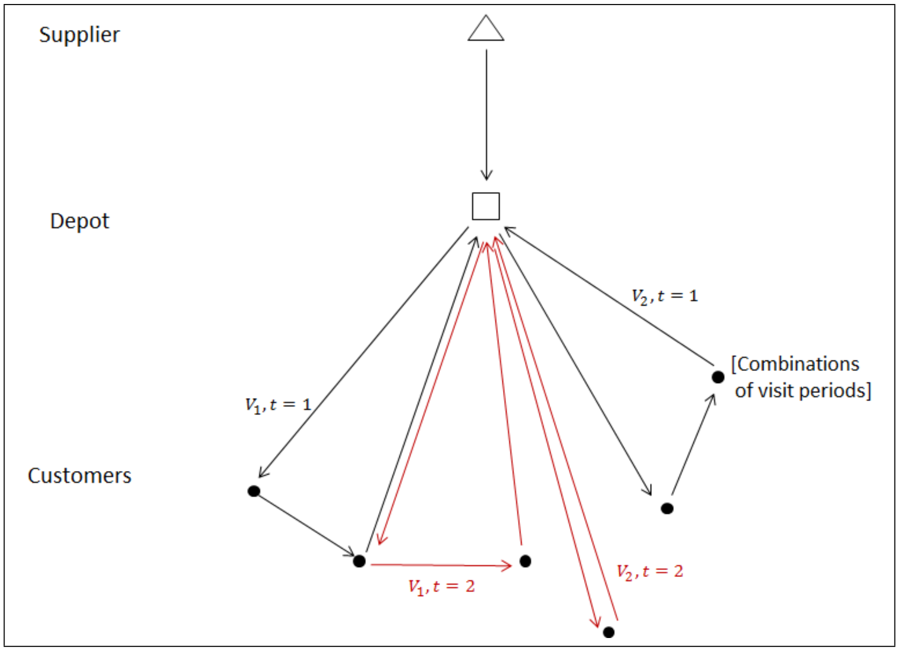

Figure 1: Two-Echelon Delivery System with Periodic Routing

This is because customers will commonly not accept delivery every day but rather follow a certain delivery pattern. An overview of the system is depicted in Figure 1.

The problem is defined over a limited time horizon of a week and the solution must satisfy the periodic customer demands. Customer demands are assumed to be deterministic and known for each period at the beginning of the weekly time horizon. The supplier has enough capacity to satisfy the demand and can deliver to the depot within a reasonably short-time frame. It is therefore possible to deliver goods to the depot and thence to different customer locations within the same day, i.e. a time period. Vehicle capacity, however, is limited and each delivery to the depot incurs a linehaul travel cost.

Products can be stored at the depot up to a certain capacity or until they are discarded as waste due to their perishable nature. Perishable products can be generally categorised into two types. The first type is associated with an expiry date, meaning that the products are suitable for consumption up until a certain point in time, after which they are discarded (Nahmias, 2011). This is often the case for dairy products for example. The second type relates to a gradual decrease in product quality and can be for example observed for salads, fruits and bread (Rong et al., 2011). The focus in this research will be on the latter type. Thus, the deterioration of a product occurs over time in relation to the age of the product. The cost of this deterioration is included in the inventory holding cost. From the depot to the customer, delivery is carried out by a homogeneous fleet of vehicles. Vehicles are readily available at the depot, though limited with respect to their capacity. The problem consists of the following decisions:

- When and how much to deliver from the supplier to the depot?
- When and how much to deliver from the depot to the customers?
- What is the optimal routing from the depot to the customer locations for the different time periods?

The aim is therefore to determine an optimal delivery schedule and routing to the customer locations during each time period, and optimise the storage time of the product at the depot, minimising both the routing and inventory costs, while accounting for the loss of freshness of the product over time.

3 Mathematical Formulation

This section introduces the notations and parameters used and provides a formal description of the mixed integer linear programming (MILP) formulation based on the assumptions presented in Section 2.

Table 2: **Summary of notation**

Sets	
N	set of nodes indexed by i, j, l {depot: 0; customer: 1,...,n}
A	set of arcs (i, j): $i, j \in N, i \neq j$
T	set periods indexed by t
K	set of vehicles indexed by $k : k \in \{1, ..., m\}$
G	set of product ages indexed by g
R_i	set of visit combinations of i
Parameters	
c_{ij}	routing costs on arc $(i, j) : i, j \in \{0, ..., n\}$
C	linehaul routing cost supplier-depot-supplier
d_i^t	demand of customer i in period t
Q^k	capacity of vehicle k ($k = 0$: supplier-depot; $k = 1, 2, 3$: depot-customer)
H	inventory holding capacity at depot
h^g	unit inventory holding cost at depot (including deterioration cost) for product age g
a^{rt}	1 if day t belongs to visit combination r
Variables	
x_{ij}^{kt}	1 if customer j is visited immediately after customer i by vehicle k in period t
y_i^{kt}	1 if vehicle k visits customer i in period t
z_i^r	1 if visit combination r of customer i chosen
u^t	number of vehicles supplier-depot in period t
v_i^{gkt}	quantity delivered of age g from depot to customer i in period t by vehicle k
w^t	quantity delivered from supplier to depot in period t
I^{gt}	quantity held of age g at depot in period t
s_i^{kt}	position of customer i in the routing of vehicle k in time period t

The problem is then:

$$\textbf{Minimise} \quad \sum_{t\in T} Cu^t + \sum_{g\in G}\sum_{t\in T} h^g I^{gt} + \sum_{i\in N}\sum_{j\in N}\sum_{k\in K}\sum_{t\in T} c_{ij}x_{ij}^{kt} \tag{1}$$

subject to

$$I^{gt} = I^{g-1,t-1} - \sum_{i\in N}\sum_{k\in K} v_i^{g-1,k,t-1} \qquad g \in G\backslash\{0\}, t \in T\backslash\{0\} \tag{2}$$

$$I^{0t} = w^t \qquad t \in T \tag{3}$$

$$I^{gt} \geq \sum_{i\in N\backslash\{0\}}\sum_{k\in K} v_i^{gkt} \qquad g \in G, t \in T \tag{4}$$

$$\sum_{g\in G} I^{g0} = w^0 \tag{5}$$

$$\sum_{g\in G} I^{gt} \leq H \qquad t \in T \tag{6}$$

$$\sum_{r\in R_i} a^{rt} d_i^t z_i^r = \sum_{g\in G}\sum_{k\in K} v_i^{gkt} \qquad i \in N\backslash\{0\}, t \in T \tag{7}$$

$$w^t \leq H - \sum_{g\in G} I^{g,t-1} \qquad t \in T \tag{8}$$

$$\sum_{g\in G}\sum_{i\in N\backslash\{0\}} v_i^{gkt} \leq Q^k y_0^{kt} \qquad k \in K, t \in T \tag{9}$$

$$w^t \leq Q^0 u^t \qquad t \in T \tag{10}$$

$$\sum_{k\in K} y_i^{kt} \leq 1 \qquad i \in N\backslash\{0\}, t \in T \tag{11}$$

$$y_i^{kt} \leq \sum_{j\in N} x_{ij}^{kt} \qquad i \in N, k \in K, t \in T \tag{12}$$

$$\sum_{r\in R_i} z_i^r = 1 \qquad i \in N\backslash\{0\} \tag{13}$$

$$\sum_{i\in N}\sum_{k\in K} x_{ij}^{kt} - \sum_{r\in R_j} a^{rt} z_j^r = 0 \qquad j \in N\backslash\{0\}, t \in T \tag{14}$$

$$\sum_{i\in N} x_{ij}^{kt} - \sum_{l\in N} x_{jl}^{kt} = 0 \qquad k \in K, t \in T, j \in N \tag{15}$$

$$s_i^{kt} - s_j^{kt} + n x_{ij}^{kt} \leq n - 1 \qquad i, j \in N\backslash\{0\}, t \in T, k \in K \tag{16}$$

$$v_i^{gkt},\ w^t,\ I^{gt},\ s_i^{kt} \geq 0 \tag{17}$$

$$x_{ij}^{kt},\ y_{kt},\ z_i^r \in \{0, 1\} \tag{18}$$

$$u^t \in \mathbb{Z}. \tag{19}$$

The objective function (1) minimises the sum of the delivery cost, consisting of linehaul travel and customer routing cost, and of the inventory cost. Constraints (2) and (3) are inventory constraints related to the age of the product. Constraints (4) requires a delivery to update the inventory in period zero. Constraints (5) and (6) ensure that inventory levels cover at least the delivery during the same period while also not exceeding the inventory capacity at the depot. Constraints (7) mean that the demand at each consumer is met for the chosen delivery pattern. Constraints (8) restrict the amount that can be delivered to the depot depending on depot capacity and existing inventory. Constraints (9) and (10) impose a vehicle capacity for delivery to the customer and the delivery to the depot. Constraints (11) state that delivery to a customer is made by only one vehicle, while constraints (12) mean that a delivery can only be made by an activated vehicle. Constraints (13) assign a delivery pattern to each customer and constraints (14) ensure that delivery can only occur on days belonging to the chosen delivery pattern. Constraints (15) state that each vehicle that visits a customer also leaves the customer. Constraints (16) are standard subtour elimination constraints. Constraints (17) to (19) enforce the non-negativity and integrality of the variables.

4 Heuristic

For very small instances, the problem can be solved to optimality by a standard integer linear programming solver, whereas this is not feasible for larger-size instances. We therefore propose a two-stage matheuristic, i.e. a "heuristic algorithm[ ] made by the interoperation of metaheuristics and mathematical programming techniques" (Boschetti et al., 2010), combining an adaptive large neighbourhood search (ALNS) with the solution of a MILP formulation, in order to solve the problem for more realistic instances. This two-stage approach allows the exploitation of the two-echelon structure of the problem by splitting it into routing and linehaul related decisions. The overall performance of the heuristic, however, depends on the cost structure of the instances, thus determining the order in which the different components of the heuristic are solved. As a result, three variants obtained by altering the structure of the heuristic, are proposed in this research.

4.1 Heuristic Variant 1

In the first variant, the MILP model is solved first, determining optimal customer delivery patterns, linehaul travel and the inventory of products at the depot. Based on this optimal solution, the ALNS then aims to find good solutions for the second-stage routing problem, delivering each customer according to the optimal delivery patterns determined in the first stage. The structure of the approach is described in pseudo-code in Algorithm 1.

Algorithm 1 General framework: Heuristic 1

1. Solve the MILP model (including delivery pattern selection)

2. Construct initial routing solution s

3. $s^* \leftarrow s$

4. Start search procedure:

while stopping criteria not met **do**

 4.1. Select destroy and repair operators from list Z based on weighting

 4.2. Apply chosen destroy and repair operators to s to obtain s'

 if acceptance criteria satisfied **then**

  $s \leftarrow s'$

  if s better than s^* **then**

   $s^* \leftarrow s$

5. Return best solution s^*

4.1.1 Solution of a MILP and construction of an initial routing solution

The MILP formulation used to solve the first stage of the problem is a reduced version of the mathematical formulation provided in Section 3. Note that while most parameters and variables remain the same, the variables v_i^{gkt} are replaced with the variables v^{gt}, and new variables d^t are added to the model in order to determine the aggregated demand for each period t. The detailed mathematical formulation is provided in the following:

$$\textbf{Minimise} \quad \sum_{t \in T} Cu^t + \sum_{g \in G} \sum_{t \in T} h^g I^{gt} \tag{20}$$

subject to

$$d^t = \sum_{g \in G} v^{gt} \qquad t \in T \tag{21}$$

$$I^{gt} = I^{g-1,t-1} - v^{g-1,t-1} \qquad g \in G \backslash \{0\}, t \in T \backslash \{0\} \tag{22}$$

$$I^{0t} = w^t \qquad t \in T \tag{23}$$

$$I^{gt} \geq v^{gt} \qquad g \in G, t \in T \tag{24}$$

$$\sum_{g \in G} I^{g0} = w^0 \tag{25}$$

$$\sum_{g \in G} I^{gt} \leq H \qquad t \in T \tag{26}$$

$$\sum_{i \in N \backslash \{0\}} \sum_{r \in R_i} a^{rt} d_i^t z_i^r = \sum_{g \in G} v^{gt} \qquad t \in T \tag{27}$$

$$w^t \leq H - \sum_{g \in G} I^{g,t-1} \qquad t \in T \tag{28}$$

$$w^t \leq Q^0 u^t \qquad t \in T \tag{29}$$

$$\sum_{r \in R_i} z_i^r = 1 \qquad i \in N \backslash \{0\} \tag{30}$$

$$v_{gt},\ w^t,\ I^{gt} \geq 0, \qquad z_i^r \in \{0,1\}, \qquad u^t \in \mathbb{Z}. \tag{31}$$

Once the MILP model is solved, an insertion heuristic (see Algorithm 2) is applied in order to determine the routing for each period t based on the previously selected customer delivery patterns. For each day, the customers allocated to the corresponding daily delivery list are chosen randomly and inserted in the best feasible position under consideration of all the daily routes. If no feasible insertion can be found due to the capacity restrictions, a new route is created and the customer is inserted in it.

Algorithm 2 Construction heuristic for the initial solution

1. $L \leftarrow \{1, ..., n\}$
for every customer $i \in L$ **do**
  1.1. Consider selected delivery pattern $r' \in R_i$
  for every day t in delivery pattern r' **do**
    1.2. Insert customer i in daily delivery list L^t
for every day $t \in T$ **do**
  while $L^t \neq \emptyset$ **do**
    2. Randomly select customer i from daily delivery list L^t
    for every customer i **do**
      2.1. Insert customer i in its best feasible insertion place
      if no feasible insertion place **then**
        2.1.1. Create new route and insert customer in new route
  3. $L^t \leftarrow L^t - \{i\}$

4.1.2 Destroy operators

We have developed a number of different destroy operators, operating at the customer and route level. The operators remove a percentage Nb of the customers from the current solution.

Random Customer Removal: This is a standard operator in which customers are selected randomly and removed from their current route.

Worst Customer Removal: For a random day, this operator identifies the worst customer according to its insertion cost in the current solution. This customer with the largest insertion cost or savings potential is then removed from the solution.

Related Customer Removal: This operator is based on the related customer removal operators used by Shaw (1998) and Azi et al. (2014). However, while Azi et al. (2014) build on Shaw (1998) by defining a proximity measure based on spatial and temporal distance, we apply two variants of the operator. The first uses a spatial distance measure, so that $z_{il} = c_{il}$, while the second applies a distance measure based on the difference in demands between customers, so that $z_{il} = |d_i^t - d_l^t|$. The structure of the operator is described in Algorithm 3.

Algorithm 3 Related Customer Removal

---

1. Select a customer j at random from the solution

2. Consider the delivery pattern assigned to customer j and select a random day t in the pattern

3. Remove customer j from day t

4. $L \leftarrow \{j\}$

while $|L| \leq \text{Nb}$ **do**

  5.1 Select a random customer i from L

  for each customer l in day d **do**

    5.1. Compute distance measure z_{il}

    5.2 Sort the resulting z_{il}'s in decreasing order, storing them in a list B

    5.3 Choose a random number x between 0 and 1

    5.4 $pos \leftarrow |B|x^b$

    5.5 Select customer l associated with the z_{il} value at position pos and remove it from day t

    5.6 $L \leftarrow L \cup \{l\}$

6. Return L

Parameter b in 5.4 regulates the intensity of the bias towards closer customers. As a result of the tuning of this parameter, b was set to 10 in our implementation.

Random Route Removal: For a random day, the operator selects a number of routes at random and removes them from the solution.

4.1.3 Repair operator

Customers that have been removed during the destruction procedure need to be reinserted using a repair operator. The operator used for this is based on cheapest insertion and follows the same insertion procedure as in the construction heuristic. Thus, for each day, customers are selected from the list of removed customers and reinserted into the solution in the cheapest feasible position. This process is repeated until, for each day, all of the daily customers are again part of the solution.

4.1.4 Acceptance criterion and adaptive mechanism

The acceptance criteria for candidates is based on a simulated annealing rule, as in Ropke and Pisinger (2006). The adaptive mechanism is only applied to the destroy operators in this setting, since the options to repair a solution are limited to the cheapest insertion operator. The heuristic terminates after a fixed number of iterations.

4.2 Heuristic Variant 2

The second variant integrates the selection of customer delivery patterns within the routing decision, constructing daily delivery routes before optimally solving the linehaul and inventory part of the problem. The MILP model is thus integrated into the ALNS framework and solved for each of the found solutions with alternative delivery patterns. The structure of the approach is described in pseudo-code in Algorithm 4.

Algorithm 4 General framework: Heuristic 2

1. Construct initial solution s (including delivery pattern selection and MILP formulation)

2. $s^* \leftarrow s$

3. Start search procedure:

while stopping criteria not met **do**

  3.1. Select destroy and repair operators from list Z^* based on weighting

  3.2. Apply chosen destroy and repair operators to s to obtain s'

  if change in delivery pattern selection **then**

    4. Solve the MILP model and update solution s'

  if acceptance criteria satisfied **then**

    $s \leftarrow s'$

    if s better than s^* **then**

      $s^* \leftarrow s$

5. Return best solution s^*

4.2.1 Construction of an initial solution

Whereas in the previous variant, the pattern selection was predetermined by the MILP model, here the construction heuristic starts by randomly selecting a delivery pattern for each customer. Once every customer has been assigned a delivery pattern, the procedure is identical to the previous construction heuristic, with the exception that a MILP model is solved at the end in order to determine the linehaul and storage component of the problem under consideration of the chosen delivery patterns. An overview of the construction heuristic can be found in Algorithm 5. The corresponding MILP model is a simplified version of the model presented in Section 4.1.1, for which the objective function and most of the constraints remain the same. However, the expressions d_t, representing the daily aggregated demand, are now predetermined by the heuristic and thus become parameters in the model, while constraints (27) and (30), linked to the pattern selection, are eliminated from the model, making the notations z_i^r and a^{rt} irrelevant.

Algorithm 5 Construction heuristic for the initial solution

1. $L \leftarrow \{1, ..., n\}$

for every customer $i \in L$ **do**

  1.1. Randomly select delivery pattern $r' \in R_i$

  for every day t in delivery pattern r' **do**

    1.2. Insert customer i in daily delivery list L^t

for every day $t \in T$ **do**

  while $L^t \neq \emptyset$ **do**

    2.1. Randomly select customer i from daily delivery list L^t

    for every customer i **do**

      2.2. Insert customer i at its best feasible insertion place

      if no feasible insertion place **then**

        2.2.1. Create new route and insert customer in new route

  3. Solve the MILP model for the linehaul travel and storage at the depot

  4. $L^t \leftarrow L^t - \{i\}$

4.2.2 Destroy Operators

In addition to the destroy operators of the first heuristic variant, the second variant also includes an operator at the delivery pattern level to allow for changes in the selection of customer delivery patterns.

Random Customer and Pattern Removal: The idea is the same as in the standard customer removal, but in addition to removing a random customer from all the routes, the operator also removes the record of the customer's delivery pattern from the solution, storing the customer in a list of unassigned customers without a selected delivery pattern.

4.2.3 Repair Operators

Similar to the destroy operators, the repair operators consist of those used in the previous variant and of a number of new operators related to the selection of customer delivery patterns. Note, that the use of repair operators depends on the preceding destroy operator, i.e. whether the operator affects solely the customer level or both the customer and the delivery pattern level. If both levels are affected, the heuristic first chooses an operator to select the delivery pattern and assign the customer to delivery days before choosing another operator to insert the customer in the routing solution.

Random Pattern Selection: The random pattern selection operates in the same manner as the pattern selection in the construction heuristic, where a delivery pattern is chosen at random from the list of customer specific delivery patterns.

Balanced Pattern Selection: This operator takes a more balanced approach for the pattern selection by applying a customer density measure ρ. Note that two different customer density scores are considered in this research. The first is based on the sum of the number of customers in each day of the pattern, while the second is based on the sum of the daily demands of each day in the pattern. The detailed structure of the operator is described in Algorithm 6.

Algorithm 6 Balanced Pattern Selection

while list of customers without patterns not empty **do**

  1. Randomly select customer j from list of customers without patterns

  for pattern $\in$ customer patterns R_j **do**

    2.1 Calculate the customer density measure ρ associated with the pattern

  3. Select best pattern for customer j based on smallest ρ

  4. Remove customer j from list of customers without patterns

4.3 Heuristic Variant 3

We have developed a third heuristic variant consisting of a hybrid of the first two variants. In this variant, the initial solution is generated in the same way as for variant 1 (i.e. the MILP model of variant 1 for the linehaul is solved first), while the general structure of the heuristic follows variant 2. This means that variant 3 starts with the optimal cost from the linehaul perspective but allows for more flexibility since the linehaul problem can be solved for different customer patterns using the MILP model of variant 2.

4.3.1 Acceptance criterion and adaptive mechanism

We use the same acceptance criterion as in the first variant, with the exception that the adaptive mechanism is applied to both the destroy operators and the repair operators used to restore customer patterns.

5 Computational Results

We have carried out a number of computational experiments in order to validate the MILP formulation and test the heuristic approaches proposed in this study. The experiments were designed to investigate the effect of the cost structure on the performance of the three heuristic variants. We have coded the heuristics in Python3.5 and CPLEX 12.6 running on a single thread. All computations were executed on a machine equipped with an Intel(R) Xeon(R) X5675 processor running at 3.07GHz.

5.1 Instance description

The heuristics were tested on two sets of instances, each consisting of 90 small-size instances with up to 150 customers. The two sets differ with respect to the size of the grid in which the customers are located. In the first set of instances the customers are located in a 2×2 square (in km), i.e. small grid, while in the second set the customers are located in a 25×25 square (in km), i.e. large grid. In both cases, the depot is located at (2.5,2.5). The distance for the linehaul travel is set at one of the following values for both sets: 20 km, 40 km or 80 km. Thus, the two different grid sizes impact the relations between the cost components of the objective function, as linehaul and inventory costs remain in the same range. This means, that in the case of the large grid instances, the routing cost contributes relatively more to the total cost. The calculation of the routing costs c_{ij} is based on the Euclidean distances between the locations in the plane and include fuel, wage and vehicle costs per km.

Table 3: Routing cost components per vehicle type

	Light Duty Vehicle	**Medium Duty Vehicle**
Average speed	30 km/h	70 km/h
Fuel consumption	30 l per 100 km	15 l per 100 km
Fuel cost	**0.42 €/km**	**0.21 €/km**
Driver's wage	9.5 €/h	12.5 €/h
Wage costs	**0.3 €/km**	**0.18 €/km**
Truck payment and insurance	0.3 €/km	0.3 €/km
Maintenance and repairs	0.15 €/km	0.15 €/km
Vehicle costs	**0.45 €/km**	**0.45 €/km**
Total routing cost per km	**1.17 €/km**	**0.84 €/km**

The inventory costs increase exponentially and are calculated based on the formula $h^g = price \times p \times b^t$, where the price of the product is randomly selected from the interval [10, 30], p is a percentage in [0.02, 0.04], b is a positive growth factor set at 2, and t is the time period. The planning horizon in this research is set for all instances to $T = 5$. Five delivery patterns consisting of different combinations of delivery days ({1,3}, {1,4}, {2,4}, {2,5}, {3,5}), are considered and each customer is assigned two delivery patterns chosen at random. Based on the instances of Song and Ko (2016), the customer demands are volume based and range between 0.3 m^3 and 1.8 m^3, the capacity of the vehicles used for the customer routing is set at 12 m^3. The vehicle capacity for the linehaul travel is 38 m^3, which corresponds to the standard size of a small shipping container in Europe. For the small grid instance structure the capacity at the depot is 50 m^3 for instances with up to 40 customers and 150m^3 for larger instances. The large grid instance structure features a larger inventory capacity at the depot of 100 m^3 for instances of up to 40 customers and 200 m^3 for larger instances.

5.2 Parameter settings

For the parameter tuning of the two heuristics, two sets of 18 test instances were selected at random, representing the two different instance structures considered in this research. The tuning for these two test sets was carried out separately. We executed 10 runs for each parameter setting of the heuristics, and the setting with the best average deviation from the best found solution was chosen. The results of the tuning were based on a search consisting of 25,000 iterations and a segment size of 200 iterations, as this resulted in a good trade-off between run time and solution quality. Three different intervals for the percentage of destruction (i.e. the percentage of customers to remove from the solution) were reviewed, namely $[10\%, 30\%]$, $[20\%, 40\%]$, $[30\%, 50\%]$. All other parameter values were initially set equal to those of Ropke and Pisinger (2006) and then sequentially altered in the tuning phase. The resulting best parameter setting for each heuristic variant and instance structure is shown in Table 4.

Table 4: Parameter settings

	Instance structure 1			Instance structure 2		
Parameters	**Variant 1**	**Variant 2**	**Variant 3**	**Variant 1**	**Variant 2**	**Variant 3**
Percentage of destruction:	30%–50%	30%–50%	30%–50%	30%–50%	30%–50%	30%–50%
Acceptance criterion:						
w	0.001	0.001	0.001	0.001	0.001	0.001
c	0.99976	0.99974	0.99983	0.99976	0.99983	0.99985
Weight adjustment:						
σ_1	33	22	44	22	44	33
σ_2	9	13.5	4.5	13.5	4.5	9
σ_3	13	19.5	6.5	19.5	6.5	13
r	0.1	0.1	0.1	0.5	0.5	0.5

5.3 Results

This section presents the results for the two instance structures and three heuristic variants based on the best parameter settings identified in Section 5.2. For very small instances, of up to ten customers, these results are compared with the optimal solution values found by the mathematical model, while for larger instances it is no longer possible to solve the problem to optimality. The run time for solving the model to optimality differs considerably between the two instance structures, as well as between individual instances. For the small grid instance structure, the model obtains an optimal solution for instances with 10 customers on average within 86 seconds, while for the large grid instance structure the run time is considerably longer, with an average of about 5,800 seconds. In addition, one of the large grid instances could not be solved to optimality, with a remaining optimality gap of 6.63% after running the model for several days. Increasing the number of customers to $n = 15$ for the small grid instances, leads to a considerable increase in the run time, resulting in an average run time of 7435.5 seconds.

The column headings of Tables 5 and 7 present the linehaul cost structure, the optimal solution value found by the mathematical model, as well as the performance of the heuristic variants for each of the small and large grid instances, respectively. The total cost term for the optimal solutions is broken down into the different cost components, namely the first echelon cost ($1^{st}E$), consisting of linehaul (LC) and inventory cost (IC), and the routing cost (RC). For the heuristic variants the tables provide the best solution values found, as well as the performance of the algorithms in terms of time and deviation. Best and average deviations are computed with respect to the optimal solution values found for each of the instances. The comparison of the optimal solution values with the three heuristic variants for instances with $n = 10$ in Table 5 shows that the variants manage on occasions to find optimal or close to optimal solutions for most small grid instances. On average, variant 3 performs best with an optimality gap of 1.79%, followed by variant 1 with a gap of 2.12%, and variant 2 with a gap of 2.48%. In terms of finding optimal or close to optimal solutions, variant 2 performs best, closely followed by variant 3, and considerably outperforming variant 1. For instances with 15 customers, the comparison of the optimal solution values with the three heuristic variants shows that each variant still finds reasonably good solutions. For this instance size variant 1 performs best on average with an optimality gap of 2.1%, closely followed by variant 3 with a gap of 2.4%, variant 2 under performs with an average deviation of 5.9%. A comparison with larger instances is not possible due to significantly longer run times.

Table 5: Optimal Solutions compared with the solutions found by the heuristic variants - small grid (for $n = 10$)

		Optimal Solutions						Variant 1				Variant 2				Variant 3			
#	Linehaul	Total	1st E	LC	IC	RC	Time (s)	Best	BestDev	AvDev	Time (s)	Best	BestDev	AvDev	Time (s)	Best	BestDev	AvDev	Time (s)
1	low	110.68	70.03	50.4	19.63	40.66	0	114.96	3.87	4.83	12.5	110.98	0.27	4.21	289.9	110.98	0.27	4.04	320.6
2	low	100.37	51.08	33.6	17.48	49.29	99	104.86	4.47	4.89	10.2	100.51	0.13	2.01	383.9	100.37	0	2.49	387.0
3	low	107.43	45.76	33.6	12.16	61.68	160	110.16	2.53	2.82	11.4	107.43	0	1.36	355.2	107.58	0.14	0.52	380.3
4	low	85.36	49.54	33.6	15.94	35.82	14	90.07	5.51	6.61	10.0	90.07	5.5	6.73	349.8	90.57	6.11	7.49	375.3
5	low	98.25	51.93	33.6	18.33	46.32	68	101.91	3.73	4.55	12.2	98.61	0.37	10.03	380.9	99.19	0.96	4.03	360.0
6	medium	118.36	75.91	67.2	8.71	42.45	234	119.23	0.73	1.10	11.5	118.75	0.33	2.69	281.9	119.58	1.03	2.20	317.1
7	medium	137.14	97.55	67.2	30.35	39.59	51	137.14	0	0.75	11.4	137.52	0.28	1.04	389.7	137.14	0	0.68	410.4
8	medium	159.19	113.03	67.2	45.83	46.16	72	160.48	0.81	0.98	12.5	159.22	0.02	0.98	396.2	160.48	0.81	0.97	422.0
9	medium	145.58	94.04	67.2	26.84	51.54	37	145.58	0	0.19	10.5	145.58	0	2.54	367.9	145.58	0	0.53	396.5
10	medium	136.99	89.81	67.2	22.61	47.18	48	138.18	0.87	1.59	11.3	138.00	0.73	1.57	371.8	136.99	0	0.81	406.6
11	high	203.19	152.89	134.4	18.49	50.30	86	203.32	0.06	0.54	12.3	203.43	0.11	0.91	346.5	203.41	0.11	0.45	400.0
12	high	156.22	107.74	67.2	40.54	48.48	58	156.22	0	0.21	12.2	156.22	0	0.50	393.6	156.22	0	0.34	450.2
13	high	201.95	160.86	134.4	26.46	41.09	199	206.30	0.21	2.20	12.4	201.95	0	1.06	354.1	203.44	0.74	1.50	399.0
14	high	209.87	161.80	134.4	27.40	48.07	103	210.01	0.07	0.37	12.2	210.38	0.24	1.27	378.6	210.38	0.24	0.59	442.4
15	high	211.77	172.07	134.4	37.67	39.70	56	211.77	0	0.25	12.4	211.77	0	0.30	307.2	211.77	0	0.24	371.3
Low linehaul							68.2		4.02	4.74	11.3		1.26	4.87	351.9		1.5	3.71	364.7
Medium linehaul							88.4		0.48	0.92	11.4		0.27	1.76	361.5		0.37	1.04	390.5
High linehaul							100.4		0.46	0.71	12.3		0.07	0.81	356.0		0.22	0.62	412.6
Overall							85.7		1.65	2.12	11.7		0.53	2.48	356.5		0.69	1.79	389.3

Table 6: Optimal Solutions compared with the solutions found by the heuristic variants - small grid (for $n = 15$)

				Optimal Solutions					Variant 1				Variant 2			Variant 3			
#	Linehaul	Total	1st E	LC	IC	RC	Time (s)	Best	BestDev	AvDev	Time (s)	Best	BestDev	AvDev	Time (s)	Best	BestDev	AvDev	Time (s)
1	low	154.67	93.58	50.40	43.18	61.09	11925.4	155.50	0.54	1.01	14.4	158.04	2.18	7.67	351.7	155.58	0.59	4.07	358.7
2	low	131.03	67.23	33.60	33.63	63.80	1004.9	134.48	2.64	3.92	14.3	133.98	2.25	6.96	308.3	135.21	3.19	4.05	316.8
3	low	119.71	62.36	33.60	28.76	57.35	5944.7	122.59	2.40	2.96	13.7	120.68	0.81	3.89	353.4	121.71	1.67	3.12	350.8
4	low	131.05	69.91	33.6	36.31	61.14	5023.6	134.67	2.77	3.33	13.1	135.68	3.53	6.11	307.9	135.214	3.18	4.20	338.9
5	low	121.43	97.18	67.20	29.98	24.25	6408.8	121.77	0.28	1.74	14.4	129.61	6.74	13.95	323.9	121.73	0.24	2.62	327.2
6	medium	144.18	89.22	67.20	22.02	54.96	6984.4	147.15	2.06	2.78	13.9	145.74	1.08	3.14	360.4	145.10	0.64	2.62	340.9
7	medium	145.18	82.58	67.20	15.38	62.60	53580.8	146.99	1.24	2.17	14.1	146.59	0.97	4.68	334.6	146.99	1.24	2.08	307.0
8	medium	185.93	126.93	67.20	59.73	59.00	1347.8	187.68	0.94	1.43	14.0	189.23	1.77	14.53	377.9	187.66	0.93	1.39	361.7
9	medium	149.38	90.12	67.20	22.92	59.26	2812.9	150.87	1.00	2.06	13.3	157.12	5.18	7.41	331.2	151.08	1.14	2.26	320.5
10	medium	159.22	91.78	67.20	24.58	67.44	3416.8	161.74	1.58	2.01	14.3	162.09	1.81	3.45	332.8	161.28	1.29	2.30	333.5
11	high	259.67	188.74	134.4	54.34	70.93	3492.2	262.76	1.19	1.53	12.6	263.60	1.51	2.63	319.4	262.877	1.24	1.95	331.3
12	high	226.51	173.04	134.4	38.64	53.47	5342.1	229.78	1.44	3.32	13.4	228.95	1.08	2.77	304.8	229.494	1.32	2.05	325.6
13	high	238.54	170.99	134.40	36.59	67.55	3123.3	241.22	1.12	1.72	14.1	243.11	1.91	2.85	355.5	241.45	1.22	1.89	314.8
14	high	257.44	194.44	134.4	60.04	0.00	679.86	258.66	0.47	0.79	13.4	263.27	2.27	3.80	308.4	259.176	0.67	1.55	329.9
15	high	222.05	167.66	134.40	33.26	54.39	444.9	222.05	0.00	0.50	14.4	226.02	1.79	5.20	334.5	222.05	0.00	0.40	328.4
Low linehaul							6061.5		1.7	2.6	14.0		3.1	7.7	329.1		1.8	3.6	338.5
Medium linehaul							13628.5		1.4	2.1	13.9		2.2	6.6	347.4		1.0	2.1	332.7
High linehaul							2616.5		0.8	1.6	13.6		1.7	3.5	324.5		0.9	1.6	326.0
Overall							7435.5		1.3	2.1	13.8		2.3	5.9	333.6		1.2	2.4	332.4

For the large grid instances, the comparison shows that it is harder for the heuristic variants to find optimal solutions. Variant 2 still performs best at finding good objective values, the best being on average only 1.45% worse than the optimal ones. Variant 3 closely follows with its best objective values being on average 1.64% worse than the optimal one, while the cost of the best solutions found by variant 1 deviate on average by 6.52% from those of the optimal solution. Overall, variant 3 performs best with an average optimality gap of 4.52%, followed by variant 2 with a gap of 4.63%, and variant 1 with a gap of 8.41%. This underperformance of variant 1 is caused by its structure which decomposes the problem into the first echelon and a routing problem. Starting by solving the first echelon problem to optimality, the variant fixes the customer delivery patterns and therefore restricts the solution space of the ALNS solutions to the conditional routing problem.

To better quantify this behaviour, Table 8 makes a comparison between three algorithmic strategies. The first one, in the left block, solves the problem optimally by CPLEX. The table reports the total cost and its decomposition into its various components. The second strategy, in the middle block, decomposes the problem into its two natural components: the first echelon problem and the routing problem conditioned by the first-echelon solution. It solves each of these two components optimally by CPLEX. The solution values obtained by means of this decomposition strategy deviate on average by 5.97% from the optimal solution values, even though each component is solved optimally. The third strategy, in the right block, solves the problem by our heuristic variant 1: the first echelon component is again solved optimally, but the routing component is solved heuristically by ALNS. The solution costs obtained under this strategy deviate on average by 7.96% from the optimal solution values, but only by 1.87% from the costs obtained under the second strategy. In other words, these results show that ALNS yields good solutions when compared with the optimal values yielded by the second strategy. The deviations observed between variant 1 and the optimal solutions are mostly a result of the decomposition of the problem into its two components, rather than a result of the behaviour of the ALNS per se. Our recommendation is to apply variant 3 and not variant 1 when the cost is relatively important with respect to the total cost.

Table 7: Optimal Solutions compared with the solutions found by the heuristic variants - large grid (for $n = 10$)

		Optimal Solutions						Variant 1				Variant 2				Variant 3			
#	Linehaul	Total	1st E	LC	IC	RC	Time (s)	Best	BestDev	AvDev	Time (s)	Best	BestDev	AvDev	Time (s)	Best	BestDev	AvDev	Time (s)
1	low	331.54	52.07	33.6	18.47	279.47	1445.0	333.69	0.65	1.61	12.1	334.13	0.78	1.43	452.7	333.69	0.65	1.21	467.7
2	low	318.89	44.07	33.6	10.47	274.82	4672.0	318.96	0.02	1.96	11.6	318.96	0.02	2.52	396.9	321.2	0.73	1.73	393.7
3	low	351.55	80.22	67.2	13.02	271.33	15332.0	355.15	1.02	3.65	10.3	354.38	0.81	2.65	436.9	357.94	1.82	3.50	424.9
4	low	373.03	82.75	67.2	15.55	290.28	194.0	413.40	10.82	11.44	10.6	373.57	0.14	4.14	423.5	377.91	1.31	5.20	431.3
5	low	383.38	49.77	33.6	16.17	333.61	45245.0	389.67	1.64	2.54	10.9	384.69	0.34	2.18	448.7	388.14	1.24	2.39	422.5
6	medium	295.13	95.69	67.2	28.49	199.44	496.0	331.98	12.49	15.18	12.5	308.28	4.46	7.24	305.9	302.19	2.39	5.95	301.7
7	medium	305.37	96.09	67.2	28.89	209.28	174.0	306.79	0.47	1.99	11.7	305.37	0	3.97	331.8	307.65	0.75	3.59	336.1
8	medium	310.89	93.19	67.2	25.99	217.70	59.0	396.61	27.57	29.24	9.4	328.80	5.76	8.74	403.2	326.44	5.00	12.52	446.2
9	medium	392.35	107.05	67.2	39.85	285.3	384.0	416.26	6.10	7.06	10.0	399.49	1.82	6.49	422.3	403.34	2.80	7.18	430.9
10	medium	358.46	75.20	33.6	41.60	283.2	837.0	360.87	0.67	1.55	12.6	363.65	1.45	7.66	348.2	360.51	0.57	2.66	321.4
11	high	416.47	172.44	134.4	38.04	244.0	735.0	426.67	2.45	3.11	9.7	419.11	0.63	1.31	351.4	416.47	0	2.16	366.9
12	high	358.03	109.30	67.2	42.10	248.7	126.0	363.28	1.47	3.11	11.1	358.03	0	2.24	459.8	358.03	0	1.37	473.0
13	high	371.52	112.95	67.2	45.75	258.5	57.0	466.08	25.45	27.40	10.5	377.86	1.71	5.48	359.6	382.2	2.88	7.08	377.1
14	high	365.12	120.678	67.2	53.47	244.44	11617	388.20	6.3	7.85	9.5	372.07	1.9	8.82	388.4	378.47	3.65	6.68	350.0
15	high	342.12*	-	-	-	-	239661.0	344.19	0.60	1.77	11.3	348.86	1.97	9.49	362.5	345.13	0.88	2.10	311.9
Low linehaul							13377.6		2.83	4.24	11.10		0.42	2.58	431.7		1.15	2.81	428.0
Medium linehaul							390		9.46	11.00	11.23		2.7	6.82	362.3		2.3	6.38	367.3
High linehaul							50439.2		7.26	8.65	10.39		1.24	5.47	384.3		1.48	3.88	375.8
Overall							5812.357		6.52	8.41	10.88		1.45	4.63	394.96		1.64	4.52	395.9

* Optimality gap of 6.63%

Table 8: Comparison between Optimal Solutions and decomposition approaches - large grid (with $n = 10$)

		Optimal Solutions CPLEX						Decomposed Problem							Variant 1				
	Linehaul	Total	1st E	LC	IC	RC	Time (s)	Total	1st E	LC	IC	RC	Dev	Time (s)	Best	AvDev[1]	BestDev[2]	AvDev[2]	Time (s)
1	low	331.54	52.074	33.6	18.47	279.46	1445	333.69	51.26	33.6	17.66	282.43	0.6	2.8	333.69	1.61	0	0.95	12.1
2	low	318.89	44.07	33.6	10.47	274.82	4672	318.96	43.74	33.6	10.14	275.22	0.0	3.2	318.96	1.96	0	1.94	11.6
3	low	351.55	80.22	67.2	13.02	271.32	15332	355.15	50.52	33.6	16.92	304.63	1.0	4.7	355.15	3.65	0	2.60	10.3
4	low	373.03	82.74	67.2	15.54	290.28	194	411.08	60.47	33.6	26.87	350.61	10.2	25.1	413.40	11.44	0.56	1.13	10.6
5	low	383.38	49.77	33.6	16.17	333.60	45245	389.67	46.18	33.6	12.58	343.49	1.6	5.2	389.67	2.54	0	0.88	10.9
6	medium	295.13	95.68	67.2	28.48	199.44	496	325.06	92.06	67.2	24.86	233	10.1	2.81	331.98	15.18	2.13	4.58	12.5
7	medium	305.37	96.08	67.2	28.88	209.28	174	305.37	96.09	67.2	28.89	209.28	0.0	3.9	306.79	1.99	0.46	1.99	11.7
8	medium	310.89	93.19	67.2	25.99	217.7	59	392.34	89.85	67.2	22.65	302.49	26.2	0.7	396.61	29.24	1.09	2.41	9.4
9	medium	392.35	107.04	67.2	39.84	285.3	384	416.26	95.21	67.2	28.01	321.05	6.1	4.4	416.26	7.06	0	0.91	10.0
10	medium	358.46	75.201	33.6	41.60	283.26	837	358.46	75.2	33.6	41.6	283.26	0.0	86.3	360.87	1.55	0.67	1.55	12.6
11	high	416.47	172.44	134.4	38.04	244.03	735	426.67	159.57	134.4	25.17	267.1	2.4	3.7	426.67	3.11	0	0.65	9.7
12	high	358.03	109.29	67.2	42.09	248.73	126	358.66	108.26	67.2	41.06	250.4	0.2	4.7	363.28	3.11	1.29	2.93	11.1
13	high	371.52	112.94	67.2	45.74	258.57	57	464.78	108.26	67.2	41.06	356.52	25.1	1.1	466.08	27.40	0.28	1.83	10.5
14	high	365.12	120.67	67.2	53.47	244.44	11617	386.28	117.55	67.2	50.35	268.73	5.8	489.8	388.21	7.85	0.50	1.94	9.5
15	high	342.12*	-	-	-	-	239661	342.12	153.73	134.4	19.33	188.39	0.0	523.9	344.19	1.77	0.60	1.77	11.3
Low linehaul							13377.6						2.71	8.2		4.24	0.11	1.50	11.1
Medium linehaul							390						8.49	19.6		11.0	0.87	2.29	11.2
High linehaul							50439.2						6.70	204.6		8.65	0.53	1.82	10.4
Overall							21402.3						5.97	77.5		7.96	0.51	1.87	10.9

* Optimality gap of 6.63%   [1] Deviation from the optimal solution value   [2] Deviation from the solution value of the decomposed problem

The results for larger instances are shown in Tables 9 and 10. The column headings present the different instance size and linehaul cost for the two instance structures and the performance in terms of time and deviation for each heuristic variant. The deviations in the table refer to average percentage deviations from the best solutions found for each of the instances. The percentage in brackets for variant 2 is the average percentage deviation of the final linehaul cost found by variant 2 from the optimal linehaul solution found by the MILP of variant 1. The best and worst values for the individual instances of both instance structures are presented in Tables 11 -14 in the appendix. Overall, heuristic variants 2 and 3 show a significantly longer run time than variant 1 due to the destruction of customer patterns during the process of the ALNS and the associated resolution of the MILP formulation. The detailed results in Table 9 show that for the instances with a small grid size the first variant performs the best in terms of finding good solutions, followed by variant 3. Both of these variants improve the initial solution by about 30%, with an overall average deviation from the best found solutions of about 1.7% (variant 1) and 2.93% (variant 3). Breaking down the results for the different linehaul costs indicates, that both variant 1 and 3 perform better for instances with higher linehaul cost. Despite improving its initial solution on average also by about 30%, heuristic variant 2 significantly underperforms in comparison to the other two variants with an average deviation from the best found solutions of about 6.95%. As the cost of the 1st echelon (i.e the linehaul part) of the problem accounts on average for about 44% of the total cost this underperformance seems closely related to the inability of variant 2 to reach better first echelon solutions, with the first echelon solutions found by variant 2 being 8.9% worse than the optimal solutions found by the MILP model of variant 1.

Table 9: Average results by heuristic variant for small grid instances

Instance Structure 1		Variant 1		Variant 2			Variant 3	
Size	Linehaul cost	Deviation (%)	Time (s)	Deviation (%)		Time (s)	Deviation (%)	Time (s)
30	low	2.01	20.15	9.18	(13.84)	256.08	2.89	235.71
	medium	1.50	19.76	7.13	(9.29)	273.00	2.81	265.72
	high	1.44	20.35	7.53	(7.09)	291.06	1.73	275.43
40	low	2.09	23.42	7.50	(10.28)	215.30	3.16	199.95
	medium	1.34	23.58	8.05	(11.15)	226.30	2.33	219.50
	high	1.03	23.75	7.67	(7.58)	230.44	1.81	209.62
50	low	3.27	29.55	6.27	(10.99)	208.00	5.56	202.66
	medium	1.81	29.48	8.04	(12.31)	219.54	2.52	208.40
	high	1.24	30.00	6.94	(8.52)	213.35	2.06	201.29
100	low	2.23	52.66	6.50	(10.59)	201.97	4.45	198.41
	medium	1.51	53.07	7.29	(9.07)	209.19	3.35	198.83
	high	1.02	53.42	5.69	(5.57)	216.60	2.58	204.91
150	low	1.73	86.90	5.04	(6.16)	196.52	3.14	198.48
	medium	1.72	85.14	5.54	(6.94)	200.91	2.87	204.00
	high	1.18	84.92	5.88	(6.69)	204.56	2.19	212.45
Low linehaul		2.27	42.54	6.90	(10.37)	215.57	3.84	207.04
Medium linehaul		1.58	42.21	7.21	(9.75)	225.79	2.78	219.29
High linehaul		1.18	42.49	6.74	(7.09)	231.20	2.07	220.74
Overall		1.67	42.41	6.95	(9.07)	224.19	2.90	215.69

The detailed results for the second instance structure with a larger grid size are presented in Table 10. It can be seen, that the difference in performance between the three heuristic variants isn't as large as for the first set of instances. Variant 1 performs best with respect to the quality of the solutions found, with an average deviation from the best solutions of 4.66%, followed by variant 2 with a deviation of 5.75%. Heuristic variant 2 performs only slightly worse than the other two variants with an average deviation of 6.43%. The average deviations are higher than for instance structure 1, but the improvement from the initial solutions is also larger, with an improvement from the initial solution by all three variants of about 41%. When distinguishing between different linehaul costs, the results show that all variants perform better for instances with high linehaul costs. While variant 2 still underperforms in terms of finding good solutions for the first echelon of the problem (with a deviation of 10.46%), 1st echelon costs only account for about 20% of the total costs. Thus, it partly compensates for the larger linehaul cost by allowing for more flexibility and finding better solutions for the routing part of the problem. This suggests, that the ratio between linehaul and routing costs impacts the performance of the heuristic variants.

Table 10: Average results by heuristic variant for large grid instances

Instance Structure 1		Variant 1		Variant 2			Variant 3	
Size	Linehaul cost	Deviation (%)	Time (s)	Deviation (%)		Time (s)	Deviation (%)	Time (s)
30	low	6.44	20.15	8.88	(13.53)	184.92	7.63	179.94
	medium	5.00	19.96	5.33	(14.67)	184.08	4.78	183.00
	high	4.35	19.89	6.96	(11.70)	193.27	5.68	184.82
40	low	6.40	25.18	6.47	(12.81)	166.42	7.53	174.02
	medium	6.85	25.48	7.36	(14.75)	167.12	8.45	174.52
	high	3.72	25.30	5.40	(8.27)	182.47	4.77	180.84
50	low	6.16	28.00	6.76	(9.28)	157.69	6.83	163.01
	medium	6.20	28.76	6.29	(12.71)	163.82	6.52	167.10
	high	4.51	29.41	7.32	(13.81)	166.85	5.42	165.15
100	low	3.41	51.41	6.03	(11.78)	177.38	4.78	175.16
	medium	3.50	50.11	6.14	(7.84)	180.20	5.86	176.66
	high	2.66	50.16	5.63	(7.42)	179.17	3.81	176.05
150	low	3.34	81.84	4.81	(5.22)	196.83	4.45	198.36
	medium	3.94	81.86	7.22	(7.19)	200.69	5.46	202.50
	high	3.39	80.50	5.87	(5.87)	203.12	4.31	201.64
Low linehaul		5.15	41.32	6.59	(10.52)	176.65	6.25	178.10
Medium linehaul		5.10	41.23	6.47	(11.43)	179.18	6.21	180.76
High linehaul		3.72	41.05	6.24	(9.41)	184.97	4.80	181.70
Overall		4.66	41.20	6.43	(10.46)	180.27	5.75	180.18

In addition, testing instances (of 30 to 50 customers), in which more delivery patterns are allowed per customer, has shown that for both instance structures the differences in performance between the three heuristic variants become more pronounced as the number of delivery patterns increases.

6 Conclusion

We have introduced the two-echelon inventory routing problem for perishable products. The problem was formulated mathematically and was solved by applying a two-stage matheuristic combining an ALNS with a MILP formulation. Three variants of the matheuristic were proposed and tested on different types of instance structures, varying in grid size and linehaul cost. The results demonstrate that instances of realistic sizes (involving up to 150 customers) can be solved by means of the proposed heuristic within reasonable computing times. The three variants of the heuristic differ greatly on small grid instances, but tend to become more similar on the larger grid instances. It is also easier to solve the problem optimally on the smaller grids. One limitation of this paper, which could possibly be overcome in future research, lies in the modelling of perishability. Indeed, we have assumed in our model that all products deteriorate linearly as a function of time. However, these phenomena are more complex in practice since not all products deteriorate linearly and at the same rate. Therefore, a more refined model could be exploited, particularly one that would take stochasticity into account.

Acknowledgements

This research was partially funded by The Netherlands Organisation of Scientific Research (NWO), as part of the Greendish project (ALWGroen.2014.017), by the Canadian Natural Sciences and Engineering Research Council under Grant 2015-06189 and by the WASS junior researcher grant. This support is gratefully acknowledged. Thanks are due to the referees for their valuable comments.

References

[1] Akkerman, R., Farahani, P., & Grunow, M. (2010). Quality, safety and sustainability in food distribution: a review of quantitative operations management approaches and challenges. OR Spectrum, 32(4), 863–904.

[2] Amorim, P., Meyr, H., Almeder, C., & Almada-Lobo, B. (2013). Managing perishability in production-distribution planning: a discussion and review. Flexible Services and Manufacturing Journal, 25(3), 389–413.

[3] Amorim, P., & Almada-Lobo, B. (2014). The impact of food perishability issues in the vehicle routing problem. Computers & Industrial Engineering, 67, 223–233.

[4] Amorim, P., Parragh, S. N., Sperandio, F., & Almada-Lobo, B. (2014). A rich vehicle routing problem dealing with perishable food: A case study. TOP, 22(2), 489–508.

[5] Andersson, H., Hoff, A., Christiansen, M., Hasle, G., & Løkketangen, A. (2010). Industrial aspects and literature survey: Combined inventory management and routing. Computers & Operations Research, 37(9), 1515–1536.

[6] Azadeh, A., Elahi, S., Farahani, M. H., & Nasirian, B. (2016). A Genetic Algorithm-Taguchi Based Approach to Inventory Routing Problem of a Single Perishable Product with Transshipment. Computers & Industrial Engineering, 104, 124–133.

[7] Azi, N., Gendreau, M., & Potvin, J. Y. (2014). An adaptive large neighborhood search for a vehicle routing problem with multiple routes. Computers & Operations Research, 41, 167–173.

[8] Bertazzi, L., Savelsbergh, M. W. P., & Speranza, M. G. (2008). Inventory routing. Golden B.L., Raghavan S, Wasil E.A., eds. The Vehicle Routing Problem: Latest Advances and New Challenges. Springer, New York, 49–72.

[9] Boschetti, M. A., Maniezzo, V., Roffilli, M., & Röhler, A. B. (2009). Matheuristics: Optimization, simulation and control. In International Workshop on Hybrid Metaheuristics, Springer, Berlin, Heidelberg, 171–177

[10] Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2013). Thirty years of inventory routing. Transportation Science, 48(1), 1–19.

[11] Coelho, L. C., & Laporte, G. (2014). Optimal joint replenishment, delivery and inventory management policies for perishable products. Computers & Operations Research, 47, 42–52.

[12] Cordeau, J.-F., Gendreau, M., & Laporte, G. (1997). A tabu search heuristic for periodic and multidepot vehicle routing problems. Networks, 30(2), 105–119.

[13] Cordeau, J.-F., Laporte, G., Savelsbergh, M. W. P., & Vigo, D. (2007). Vehicle Routing. C. Barnhart, G. Laporte, eds. Transportation, Handbooks in Operations Research and Management Science, 14 Elsevier, Amsterdam, 367–428.

[14] Cuda, R., Guastaroba, G., & Speranza, M. G. (2015). A survey on two-echelon routing problems. Computers & Operations Research, 55, 185–199.

[15] Diabat, A., Abdallah, T., & Le, T. (2016). A hybrid tabu search based heuristic for the periodic distribution inventory problem with perishable goods. Annals of Operations Research, 242(2), 373–398.

[16] Farahani, P., Grunow, M., & Günther, H. O. (2012). Integrated production and distribution planning for perishable food products. Flexible Services and Manufacturing Journal, 24(1), 28–51.

[17] Gevaers, R., Van de Voorde, E., & Vanelslander, T. (2009). Characteristics of innovations in last-mile logistics-using best practices, case studies and making the link with green and sustainable logistics. Proceedings of the European transport conference (2009), Association for European Transport (AET), London

[18] Govindan, K., Jafarian, A., Khodaverdi, R., & Devika, K. (2014). Two-echelon multiple-vehicle locationrouting problem with time windows for optimization of sustainable supply chain network of perishable food. International Journal of Production Economics, 152, 9–28.

[19] Hemmelmayr, V. C., Cordeau, J.-F., & Crainic, T. G. (2012). An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. Computers & Operations Research, 39(12), 3215–3228.

[20] Hiassat, A., & Diabat, A. (2011). A location-inventory-routing problem with perishable products. Proceedings of the 41st International Conference on Computers and Industrial Engineering, 386–391. Curran Associates, Inc., Los Angeles.

[21] Hiassat, A., Diabat, A., & Rahwan, I. (2017). A genetic algorithm approach for location-inventory-routing problem with perishable products. Journal of manufacturing systems, 42, 93–103.

[22] Hsu, C. I., Hung, S. F., & Li, H. C. (2007). Vehicle routing problem with time-windows for perishable food delivery. Journal of Food Engineering, 80(2), 465–475.

[23] Jia, T., Li, X., Wang, N., & Li, R. (2014). Integrated inventory routing problem with quality time windows and loading cost for deteriorating items under discrete time. Mathematical Problems in Engineering, vol. 2014, 1–14.

[24] Kande, S., Prins, C., Belgacem, L., & Redon, B. (2015). Local Search Based Metaheuristics for Two-Echelon Distribution Network with Perishable Products. In International Conference on Operations Research and Enterprise Systems, 212–231. Springer, Cham.

[25] Laporte, G. (2009). Fifty years of vehicle routing. Transportation Science, 43(4), 408–416.

[26] Le, T., Diabat, A., Richard, J. P., & Yih, Y. (2013). A column generation-based heuristic algorithm for an inventory routing problem with perishable goods. Optimization Letters, 7(7), 1481–1502.

[27] Li, Y., Chu, F., Yang, Z., & Calvo R.W. (2016). A Production Inventory Routing Planning for Perishable Food with Quality Consideration. IFAC-PapersOnLine, 49(3), 407–412.

[28] Mirzaei, S., & Seifi, A. (2015). Considering lost sale in inventory routing problems for perishable goods. Computers & Industrial Engineering, 87, 213–227.

[29] Nahmias, S. (2011). Perishable Inventory Systems (Vol. 160). Springer Science & Business Media, New York.

[30] Osvald, A., & Stirn, L. Z. (2008). A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food. Journal of Food Engineering, 85(2), 285–295.

[31] Rabbani, M., Ramezankhani, M. J., Farrokhi-Asl, H., & Farshbaf-Geranmayeh, A. (2015). Vehicle Routing with Time Windows and Customer Selection for Perishable Goods. International Journal of Supply and Operations Management, 2(2), 700–719.

[32] Rabbani, M., Farshbaf-Geranmayeh, A., & Haghjoo, N. (2016). Vehicle routing problem with considering multi-middle depots for perishable food delivery. Uncertain Supply Chain Management, 4(3), 171–182.

[33] Rahimi, M., Baboli, A., & Rekik, Y. (2017). Inventory Routing Problem for Perishable Products by Considering Customer Satisfaction and Green Criteria. In: Freitag M., Kotzab H., Pannek J. (eds) Dynamics in Logistics, 445–455. Lecture Notes in Logistics. Springer, Cham

[34] Rong, A., Akkerman, R., & Grunow, M. (2011). An optimization approach for managing fresh food quality throughout the supply chain. International Journal of Production Economics, 131(1), 421–429.

[35] Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation science, 40(4), 455–472

[36] Schmid, V., Doerner, K. F., & Laporte, G. (2013). Rich routing problems arising in supply chain management. European Journal of Operational Research, 224(3), 435–448.

[37] Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. M. Maher, J.-F. Puget (Eds.), Principles and Practice of Constraint Programming, Springer-Verlag, New York, 417–431

[38] Song, B. D., & Ko, Y. D. (2016). A vehicle routing problem of both refrigerated-and general-type vehicles for perishable food products delivery. Journal of Food Engineering, 169, 61–71.

[39] Soysal, M., Bloemhof-Ruwaard, J. M., Haijema, R., & van der Vorst, J. G. (2015). Modeling an Inventory Routing Problem for perishable products with environmental considerations and demand uncertainty. International Journal of Production Economics, 164, 118–133.

[40] Tarantilis, C. D., & Kiranoudis, C. T. (2001). A meta-heuristic algorithm for the efficient distribution of perishable foods. Journal of Food Engineering, 50(1), 1-9.

[41] Wang, X., Wang, M., Ruan, J., & Zhan, H. (2016). The multi-objective optimization for perishable food distribution route considering temporal-spatial distance. Procedia Computer Science, 96, 1211–1220.

[42] Yu, M., & Nagurney, A. (2013). Competitive food supply chain networks with application to fresh produce. European Journal of Operational Research, 224(2), 273–282.

[43] Zhao, Q. H., Chen, S., & Zang, C. X. (2008). Model and algorithm for inventory/routing decision in a three-echelon logistics system. European Journal of Operational Research, 191(3), 623–635.

Appendix 1: Best and Worse Values - Small grid instances

This appendix presents the best and worst values for the instance structure with a small grid size in Tables 11 and 12 respectively. The column headings present the instance size and linehaul cost structure for each of the instances, the best known value and the best (Table 11) or worst (Table 12) solutions found by the three heuristic variants. The solutions are provided in terms of the total cost (TC) as well as broken down into first echelon costs (1^{st}E) and routing costs (RC).

Table 11: Best Solutions Found - small grid instances

Instances	**Best known**	**Heuristic 1**	**Heuristic 2**	**Heuristic 3**

Table 11: Best Solutions - small grid instances (continued)

Instances			Best known			Heuristic 1			Heuristic 2			Heuristic 3		
#	Size	Linehaul	TC	1^{st}E	RC	TC	1^{st}E*	RC	TC	1^{st}E	RC	TC	1^{st}E	RC
#	Size	Linehaul	TC	1^{st}E	RC	TC	1^{st}E*	RC	TC	1^{st}E	RC	TC	1^{st}E	RC
16	30	low	**223.7**	133.5	90.2	**223.7**	133.5	90.2	230.4	138.9	91.5	226.4	*	92.9
17	30	low	**188.1**	87.1	101.0	**188.1**	87.1	101.0	206.6	100.1	106.5	190.1	*	103.0
18	30	low	**211.0**	101.4	109.5	**211.0**	101.4	109.5	224.3	116.1	108.3	211.8	*	110.4
19	30	low	**196.3**	88.0	108.2	197.0	88.0	108.9	206.4	100.4	106.0	**196.3**	*	108.2
20	30	low	**203.1**	102.9	100.2	203.8	102.9	100.9	206.7	103.1	103.6	**203.1**	*	100.2
21	30	medium	**267.7**	176.4	91.3	**267.7**	176.4	91.3	283.0	182.6	100.5	269.8	*	93.4
22	30	medium	**240.0**	139.1	101.0	**240.0**	139.1	101.0	247.3	145.3	102.0	242.0	141.6	100.4
23	30	medium	**223.7**	124.4	99.3	225.3	118.9	106.4	**223.7**	124.4	99.3	227.1	122.2	104.8
24	30	medium	**272.0**	173.6	98.4	**272.0**	173.6	98.4	297.3	195.2	102.2	274.1	*	100.5
25	30	medium	**237.4**	138.2	99.2	237.7	136.8	100.9	238.5	147.0	91.6	**237.4**	138.2	99.2
26	30	high	**374.9**	279.8	95.1	**374.9**	279.8	95.1	391.9	292.9	99.0	378.2	*	98.4
27	30	high	**350.0**	262.3	87.7	**350.0**	262.3	87.7	370.9	274.8	96.1	351.4	*	89.1
28	30	high	**357.3**	256.1	101.2	**357.3**	256.1	101.2	371.5	268.3	103.2	361.2	*	105.1
29	30	high	**347.0**	245.8	101.2	**347.5**	245.8	101.7	365.8	261.3	104.5	347.0	*	101.2
30	30	high	**344.0**	250.8	93.2	**344.0**	250.8	93.2	359.6	262.1	97.5	344.7	*	93.8
31	40	low	**238.0**	119.2	118.8	**238.0**	119.2	118.8	249.1	130.5	118.6	238.5	*	119.4
32	40	low	**238.7**	131.2	107.5	239.1	131.2	107.9	258.5	141.3	117.2	**238.7**	*	107.5
33	40	low	**214.3**	89.0	125.3	**214.3**	89.0	125.3	223.3	92.9	130.4	218.4	*	129.4
34	40	low	**301.1**	167.8	133.3	**301.1**	167.8	133.3	304.8	176.7	128.2	304.8	*	137.0
35	40	low	**246.9**	107.8	139.1	248.1	107.8	140.3	248.1	119.3	128.8	**246.9**	*	139.1
36	40	medium	**311.9**	187.8	124.2	**311.9**	187.8	124.2	332.7	205.5	127.2	312.8	*	125.0
37	40	medium	**295.9**	166.7	129.2	296.6	166.7	130.0	302.4	174.7	127.7	**295.9**	*	129.2
38	40	medium	**307.5**	183.7	123.8	308.6	183.7	124.9	331.6	196.9	134.7	**307.5**	*	123.8
39	40	medium	**285.8**	163.5	122.4	**285.8**	163.5	122.4	292.2	172.1	120.1	285.9	*	122.4
40	40	medium	**336.2**	203.8	132.4	**336.2**	203.8	132.4	346.1	213.8	132.3	341.0	*	137.1
41	40	high	**428.8**	312.1	116.7	**428.8**	312.1	116.7	445.3	321.4	123.9	432.0	*	119.9
42	40	high	**364.7**	249.5	115.2	365.8	249.5	116.3	368.2	253.8	114.5	**364.7**	*	115.2
43	40	high	**446.0**	328.7	117.3	**446.0**	328.7	117.3	457.1	333.5	123.5	446.5	*	117.8
44	40	high	**428.3**	302.1	126.2	**428.3**	302.1	126.2	446.1	308.3	137.8	429.0	*	127.0
45	40	high	**413.4**	298.1	115.4	**413.4**	298.1	115.4	429.3	306.0	123.3	413.9	*	115.9
46	50	low	**303.0**	158.9	144.2	315.6	148.7	166.9	**303.0**	158.9	144.2	319.7	*	171.0
47	50	low	**331.3**	181.7	149.6	335.3	169.4	165.8	**331.3**	181.7	149.6	340.5	181.7	158.9
48	50	low	**256.6**	110.1	146.5	**256.6**	110.1	146.5	271.0	122.2	148.8	263.3	*	153.2
49	50	low	**236.6**	81.1	155.5	**236.6**	81.1	155.5	243.5	90.8	152.6	244.1	91.5	152.6
50	50	low	**277.6**	117.5	160.1	**277.6**	117.5	160.1	283.5	124.7	158.8	285.8	*	168.3

Table 11: **Best Solutions - small grid instances (continued)**

Instances			**Best known**			**Heuristic 1**			**Heuristic 2**			**Heuristic 3**		
#	**Size**	**Linehaul**	**TC**	1^{st}**E**	**RC**	**TC**	1^{st}**E***	**RC**	**TC**	1^{st}**E**	**RC**	**TC**	1^{st}**E**	**RC**
51	50	medium	**364.6**	204.7	160.0	366.7	204.7	162.0	379.5	215.5	164.1	**364.6**	*	160.0
52	50	medium	**326.1**	161.1	165.0	327.8	161.1	166.7	334.1	162.7	171.4	**326.1**	*	165.0
53	50	medium	**303.8**	151.1	152.7	**303.8**	151.1	152.7	314.0	158.4	155.6	304.6	157.0	147.7
54	50	medium	**351.9**	203.2	148.7	**351.9**	203.2	148.7	355.5	212.3	143.2	355.6	*	152.4
55	50	medium	**359.3**	204.0	155.3	**359.3**	204.0	155.3	371.1	220.8	150.3	361.1	*	157.1
56	50	high	**410.6**	256.4	154.2	**410.6**	256.4	154.2	422.9	261.0	161.9	413.1	*	156.7
57	50	high	**428.8**	270.7	158.1	**428.8**	270.7	158.1	447.6	286.9	160.7	429.4	*	158.7
58	50	high	**437.2**	287.0	150.2	**437.2**	287.0	150.2	457.2	300.3	156.9	439.6	*	152.6
59	50	high	**433.2**	267.6	165.6	434.2	267.6	166.6	438.0	280.3	157.7	**433.2**	*	165.6
60	50	high	**411.9**	268.6	143.3	**411.9**	268.6	143.3	443.6	301.4	142.2	415.0	*	146.4
61	100	low	**482.9**	171.7	311.2	**482.9**	171.7	311.2	498.1	185.5	312.7	488.5	*	316.9
62	100	low	**514.6**	200.9	313.7	**514.6**	200.9	313.7	521.1	219.2	301.9	533.6	217.8	315.8
63	100	low	**496.1**	189.5	306.6	**496.1**	189.5	306.6	514.3	201.5	312.8	502.5	*	313.0
64	100	low	**457.3**	165.5	291.8	**457.3**	165.5	291.8	480.8	185.0	295.8	471.6	*	306.1
65	100	low	**522.6**	226.5	296.2	**522.6**	226.5	296.2	542.0	243.3	298.8	532.0	*	305.5
66	100	medium	**597.4**	294.2	303.2	**597.4**	294.2	303.2	645.7	321.3	324.4	600.8	*	306.6
67	100	medium	**581.9**	281.2	300.7	**581.9**	281.2	300.7	604.2	293.4	310.8	587.3	*	306.2
68	100	medium	**541.3**	256.9	284.4	**541.3**	256.9	284.4	561.6	261.8	299.7	543.8	*	286.9
69	100	medium	**634.9**	334.9	300.0	**634.9**	334.9	300.0	668.7	356.7	312.0	637.0	*	302.1
70	100	medium	**569.5**	281.3	288.2	575.9	281.3	294.7	596.9	298.5	298.5	**569.5**	*	288.2
71	100	high	**928.2**	641.9	286.3	**928.2**	641.9	286.3	975.6	670.4	305.2	933.7	*	291.8
72	100	high	**783.1**	491.4	291.7	**783.1**	491.4	291.7	796.2	507.8	288.5	785.2	*	293.8
73	100	high	**811.7**	502.7	309.1	**811.7**	502.7	309.1	835.7	514.4	321.4	812.2	*	309.5
74	100	high	**887.5**	589.7	297.8	**887.5**	589.7	297.8	916.7	615.4	301.3	899.3	*	309.5
75	100	high	**834.6**	543.1	291.5	**834.6**	543.1	291.5	864.0	560.0	304.1	844.9	*	301.8
76	150	low	**829.5**	357.7	471.7	**829.5**	357.7	471.7	853.8	374.5	479.3	841.0	*	483.3
77	150	low	**748.3**	282.0	466.3	749.5	282.0	467.5	776.7	315.6	461.1	**748.3**	*	466.3
78	150	low	**885.7**	424.7	461.0	**885.7**	424.7	461.0	922.9	441.5	481.4	894.7	*	470.0
79	150	low	**867.9**	424.9	443.0	870.0	424.9	445.1	895.4	441.7	453.8	**867.9**	*	443.0
80	150	low	**981.1**	525.3	455.9	**981.1**	525.3	455.9	989.7	542.1	447.6	983.1	*	457.9
81	150	medium	**965.4**	481.0	484.4	**965.4**	481.0	484.4	1012.6	514.6	498.0	979.2	*	498.2
82	150	medium	**922.5**	460.5	462.0	**922.5**	460.5	462.0	936.0	480.9	455.1	929.7	*	469.2
83	150	medium	**932.2**	483.2	449.0	**932.2**	483.2	449.0	951.2	488.6	462.6	937.8	*	454.7
84	150	medium	**947.6**	496.7	450.9	**947.6**	496.7	450.9	1003.8	530.3	473.5	962.6	*	465.9
85	150	medium	**882.3**	426.4	456.0	**882.3**	426.4	456.0	926.3	460.0	466.4	890.6	*	464.3
86	150	high	**1180.2**	731.9	448.2	**1180.2**	731.9	448.2	1255.3	799.1	456.2	1188.7	*	456.8

Table 11: **Best Solutions - small grid instances (continued)**

Instances			**Best known**			**Heuristic 1**			**Heuristic 2**			**Heuristic 3**		
#	**Size**	**Linehaul**	**TC**	**1^{st}E**	**RC**	**TC**	**1^{st}E***	**RC**	**TC**	**1^{st}E**	**RC**	**TC**	**1^{st}E**	**RC**
87	150	high	**1158.3**	680.3	478.0	1163.9	680.3	483.6	1239.6	747.5	492.1	**1158.3**	*	478.0
88	150	high	**1242.9**	783.1	459.8	**1242.9**	783.1	459.8	1261.8	794.4	467.3	1251.3	*	468.2
89	150	high	**1363.3**	917.4	445.9	**1363.3**	917.4	445.9	1415.9	952.7	463.2	1367.7	*	450.3
90	150	high	**1193.21**	745.7	447.4	**1193.2**	745.7	447.4	1206.2	761.8	444.3	1195.7	*	449.9
Total # of best known solutions found:						56			3			16		

* optimal solution MILP model if linehaul is solved first

Table 12: Worst Solutions Obtained - small grid instances

Instances			**Best known**			**Heuristic 1**			**Heuristic 2**			**Heuristic 3**		
#	**Size**	**Linehaul**	**TC**	**1^{st}E**	**RC**	**TC**	**1^{st}E***	**RC**	**TC**	**1^{st}E**	**RC**	**TC**	**1^{st}E**	**RC**
16	30	low	**223.7**	133.5	90.2	231.9	133.5	98.4	257.6	150.3	107.3	245.6	148.1	97.4
17	30	low	**188.1**	87.1	101.0	196.8	87.1	109.7	216.1	101.5	114.6	199.3	*	112.2
18	30	low	**211.0**	101.4	109.5	219.7	101.4	118.3	235.1	122.6	112.5	218.8	106.7	112.1
19	30	low	**196.3**	88.0	108.2	204.6	88.0	116.6	215.6	101.4	114.2	203.1	*	115.0
20	30	low	**203.1**	102.9	100.2	212.1	102.9	109.3	233.4	122.2	111.2	215.5	*	112.6
21	30	medium	**267.7**	176.4	91.3	274.6	176.4	98.2	305.9	207.8	98.1	284.5	185.2	99.3
22	30	medium	**240.0**	139.1	101.0	247.6	139.1	108.5	262.5	148.5	114.0	248.5	140.3	108.2
23	30	medium	**223.7**	124.4	99.3	229.7	118.9	110.8	242.6	130.0	112.6	240.0	122.4	117.6
24	30	medium	**272.0**	173.6	98.4	279.9	173.6	106.3	309.1	209.0	100.1	280.9	173.6	107.3
25	30	medium	**237.4**	138.2	99.2	245.8	136.8	109.0	260.8	153.4	107.4	250.4	*	113.6
26	30	high	**374.9**	279.8	95.1	383.2	279.8	103.4	414.9	312.6	102.4	396.6	299.8	96.8
27	30	high	**350.0**	262.3	87.7	360.8	262.3	98.6	397.3	288.5	108.8	363.3	*	101.0
28	30	high	**357.3**	256.1	101.2	368.3	256.1	112.1	398.8	275.1	123.7	373.5	263.4	110.2
29	30	high	**347.0**	245.8	101.2	365.6	245.8	119.8	386.1	269.5	116.6	354.3	*	108.5
30	30	high	**344.0**	250.8	93.2	350.2	250.8	99.4	370.9	267.0	104.0	351.6	*	100.8
31	40	low	**238.0**	119.2	118.8	246.2	119.2	127.1	259.7	130.5	129.2	266.7	134.1	132.7
32	40	low	**238.7**	131.2	107.5	252.9	131.2	121.7	268.0	147.0	121.0	252.5	*	121.3
33	40	low	**214.3**	89.0	125.3	225.6	89.0	136.6	249.0	103.5	145.5	225.4	*	136.4
34	40	low	**301.1**	167.8	133.3	311.0	167.8	143.2	323.1	176.7	146.4	315.3	*	147.5
35	40	low	**246.9**	107.8	139.1	254.6	107.8	146.8	265.4	119.3	146.1	259.7	*	151.9
36	40	medium	**311.9**	187.8	124.2	324.1	187.8	136.4	352.0	214.9	137.2	334.4	214.9	119.6
37	40	medium	**295.9**	166.7	129.2	301.5	166.7	134.8	325.9	195.8	130.1	320.0	182.4	137.6
38	40	medium	**307.5**	183.7	123.8	315.1	183.7	131.4	350.9	211.8	139.1	318.2	*	134.5
39	40	medium	**285.8**	163.5	122.4	293.2	163.5	129.7	316.9	185.4	131.6	294.8	*	131.3
40	40	medium	**336.2**	203.8	132.4	348.7	203.8	144.9	367.3	231.8	135.5	352.3	*	148.5
41	40	high	**428.8**	312.1	116.7	440.8	312.1	128.7	476.3	341.7	134.5	449.5	319.1	130.4

Table 12: Worst Solutions - small grid instances (continued)

Instances			Best known			Heuristic 1			Heuristic 2			Heuristic 3		
#	Size	Linehaul	TC	1^{st}E	RC	TC	1^{st}E*	RC	TC	1^{st}E	RC	TC	1^{st}E	RC
42	40	high	**364.7**	249.5	115.2	369.5	249.5	120.0	402.4	273.1	129.3	374.6	251.4	123.2
43	40	high	**446.0**	328.7	117.3	454.2	328.7	125.5	488.8	360.2	128.6	462.8	*	134.1
44	40	high	**428.3**	302.1	126.2	439.4	302.1	137.3	504.9	363.9	141.0	444.4	315.5	128.9
45	40	high	**413.4**	298.1	115.4	423.6	298.1	125.6	453.7	321.7	131.9	427.6	*	129.6
46	50	low	**303.0**	158.9	144.2	333.4	148.7	184.7	331.9	156.6	175.4	328.3	158.9	169.4
47	50	low	**331.3**	181.7	149.6	343.3	169.4	173.9	354.2	181.7	172.5	361.9	181.7	180.2
48	50	low	**256.6**	110.1	146.5	269.7	110.1	159.6	289.2	128.9	160.3	286.6	121.4	165.2
49	50	low	**236.6**	81.1	155.5	249.2	81.1	168.2	258.0	96.8	161.2	257.6	95.5	162.1
50	50	low	**277.6**	117.5	160.1	290.4	117.5	172.9	302.9	132.8	170.1	294.6	130.1	164.5
51	50	medium	**364.6**	204.7	160.0	375.2	204.7	170.6	406.6	239.2	167.4	380.7	*	176.0
52	50	medium	**326.1**	161.1	165.0	336.8	161.1	175.7	362.1	193.8	168.3	337.7	*	176.6
53	50	medium	**303.8**	151.1	152.7	319.6	151.1	168.5	329.1	168.6	160.5	323.2	157.4	165.9
54	50	medium	**351.9**	203.2	148.7	362.7	203.2	159.5	404.6	238.8	165.8	365.0	*	161.8
55	50	medium	**359.3**	204.0	155.3	372.4	204.0	168.4	401.9	246.8	155.1	383.2	224.1	159.1
56	50	high	**410.6**	256.4	154.2	421.2	256.4	164.8	446.0	278.5	167.5	424.7	*	168.3
57	50	high	**428.8**	270.7	158.1	439.9	270.7	169.2	485.0	318.6	166.5	463.8	297.0	166.8
58	50	high	**437.2**	287.0	150.2	445.8	287.0	158.8	496.8	339.6	157.2	450.2	*	163.3
59	50	high	**433.2**	267.6	165.6	439.5	267.6	171.9	460.8	306.0	154.7	474.5	288.3	186.2
60	50	high	**411.9**	268.6	143.3	420.0	268.6	151.5	469.3	306.0	163.3	425.1	*	156.5
61	100	low	**482.9**	171.7	311.2	500.3	171.7	328.6	523.2	201.2	322.0	511.5	190.2	321.3
62	100	low	**514.6**	200.9	313.7	528.5	200.9	327.6	553.3	225.0	328.3	573.2	222.8	350.4
63	100	low	**496.1**	189.5	306.6	522.1	189.5	332.6	538.8	211.3	327.4	523.6	205.2	318.4
64	100	low	**457.3**	165.5	291.8	479.9	165.5	314.3	508.4	184.5	323.9	500.0	186.9	313.1
65	100	low	**522.6**	226.5	296.2	544.6	226.5	318.1	580.1	263.7	316.3	557.2	248.2	309.1
66	100	medium	**597.4**	294.2	303.2	620.1	294.2	325.9	662.1	340.0	322.1	664.6	334.8	329.8
67	100	medium	**581.9**	281.2	300.7	602.0	281.2	320.8	634.1	316.5	317.5	633.4	311.0	322.4
68	100	medium	**541.3**	256.9	284.4	556.6	256.9	299.7	593.9	278.6	315.4	578.5	263.5	315.0
69	100	medium	**634.9**	334.9	300.0	650.8	334.9	315.9	696.5	377.2	319.3	678.0	354.9	323.1
70	100	medium	**569.5**	281.3	288.2	587.3	281.3	306.0	624.1	317.4	306.7	602.8	*	321.5
71	100	high	**928.2**	641.9	286.3	946.7	641.9	304.8	1066.8	721.7	345.1	994.8	*	352.9
72	100	high	**783.1**	491.4	291.7	802.0	491.4	310.6	821.5	506.2	315.3	830.7	*	339.3
73	100	high	**811.7**	502.7	309.1	825.2	502.7	322.5	861.4	528.8	332.7	869.3	538.2	331.1
74	100	high	**887.5**	589.7	297.8	907.0	589.7	317.3	976.4	656.8	319.6	932.3	614.0	318.4
75	100	high	**834.6**	543.1	291.5	850.7	543.1	307.6	933.3	578.7	354.6	896.7	581.6	315.0
76	150	low	**829.5**	357.7	471.7	860.4	357.7	502.7	889.9	393.7	496.2	899.2	*	541.5
77	150	low	**748.3**	282.0	466.3	770.2	282.0	488.2	814.2	315.1	499.1	810.5	315.6	494.9

Table 12: **Worst Solutions - small grid instances (continued)**

Instances			**Best known**			**Heuristic 1**			**Heuristic 2**			**Heuristic 3**		
#	**Size**	**Linehaul**	**TC**	1^{st}**E**	**RC**	**TC**	1^{st}**E***	**RC**	**TC**	1^{st}**E**	**RC**	**TC**	1^{st}**E**	**RC**
78	150	low	**885.7**	424.7	461.0	922.3	424.7	497.5	948.1	441.5	506.5	934.3	441.5	492.8
79	150	low	**867.9**	424.9	443.0	905.8	424.9	480.9	920.9	441.7	479.2	927.5	441.7	485.9
80	150	low	**981.1**	525.3	455.9	1018.8	525.3	493.6	1031.6	544.4	487.2	1028.6	552.4	476.2
81	150	medium	**965.4**	481.0	484.4	1000.2	481.0	519.2	1035.3	521.7	513.6	1016.4	*	535.3
82	150	medium	**922.5**	460.5	462.0	948.8	460.5	488.3	965.4	494.1	471.3	975.7	494.1	481.6
83	150	medium	**932.2**	483.2	449.0	964.2	483.2	481.1	997.8	516.8	481.0	977.3	509.3	468.1
84	150	medium	**947.6**	496.7	450.9	976.1	496.7	479.4	1029.3	530.3	499.0	1018.3	532.4	485.8
85	150	medium	**882.3**	426.4	456.0	911.7	426.4	485.3	946.5	467.2	479.3	941.1	459.9	481.2
86	150	high	**1180.2**	731.9	448.2	1207.4	731.9	475.4	1284.4	799.1	485.2	1216.7	747.6	469.1
87	150	high	**1158.3**	680.3	478.0	1191.5	680.3	511.2	1274.2	747.5	526.7	1250.4	747.5	502.9
88	150	high	**1242.9**	783.1	459.8	1277.0	783.1	493.9	1306.1	814.2	491.9	1323.4	799.8	523.6
89	150	high	**1363.3**	917.4	445.9	1381.7	917.4	464.3	1501.5	1028.3	473.2	1416.9	959.3	457.5
90	150	high	**1193.2**	745.7	447.5	1219.3	745.7	473.6	1253.8	779.1	474.6	1229.2	*	483.5

* optimal solution MILP model if linehaul is solved first

Appendix 2: Best and Worse Values - large grid instances

This appendix presents the best and worst values for the instance structure with a large grid size in Tables 13 and 14 respectively. The column headings present the instance size and linehaul cost structure for each of the instances, the best known value and the best (Table 13) or worst (Table 14) solutions found by the three heuristic variants. The solutions are provided in terms of the total cost (TC) as well as broken down into first echelon costs (1^{st}E) and routing costs (RC).

Table 13: Best Solutions Found - large grid instances

Instances			**Best known**			**Heuristic 1**			**Heuristic 2**			**Heuristic 3**		
#	**Size**	**Linehaul**	**TC**	1^{st}**E**	**RC**	**TC**	1^{st}**E***	**RC**	**TC**	1^{st}**E**	**RC**	**TC**	1^{st}**E**	**RC**
16	30	low	**707.8**	97.9	609.9	727.0	93.4	633.5	**707.8**	97.9	609.9	719.8	*	626.4
17	30	low	**660.3**	131.7	528.5	705.6	119.1	586.5	**660.3**	131.7	528.5	693.6	128.2	565.4
18	30	low	**626.0**	83.6	542.4	**626.0**	83.6	542.4	655.4	95.6	559.8	635.7	*	552.1
19	30	low	**689.6**	110.6	578.9	696.1	106.0	590.1	706.2	115.8	590.4	**689.6**	110.6	578.9
20	30	low	**631.4**	88.1	543.3	**631.4**	88.1	543.3	673.7	106.8	566.9	636.7	91.8	544.9
21	30	medium	**823.6**	204.9	618.7	**823.6**	204.9	618.7	852.0	241.6	610.4	831.6	*	626.8
22	30	medium	**777.1**	222.1	555.0	795.3	193.3	602.0	**777.1**	222.1	555.0	788.8	*	595.4
23	30	medium	**736.3**	160.3	576.1	763.5	160.3	603.2	773.3	183.9	589.4	**736.3**	*	576.1
24	30	medium	**690.5**	131.7	558.9	699.1	111.9	587.2	**690.5**	131.7	558.9	696.0	130.6	565.5
25	30	medium	**766.0**	171.5	594.5	789.7	145.5	644.1	782.3	167.5	614.8	**766.0**	171.5	594.5
26	30	high	**738.3**	189.1	549.3	**738.3**	189.1	549.3	760.0	218.0	542.0	740.6	*	551.6

Table 13: Best Solutions - large grid instances (continued)

Instances			Best known			Heuristic 1			Heuristic 2			Heuristic 3		
#	Size	Linehaul	TC	1^{st}E	RC	TC	1^{st}E*	RC	TC	1^{st}E	RC	TC	1^{st}E	RC
27	30	high	**715.6**	178.6	537.1	**715.6**	178.6	537.1	754.4	202.2	552.2	763.2	*	584.6
28	30	high	**827.2**	233.0	594.2	831.2	233.0	598.2	851.3	256.0	595.3	**827.2**	*	594.2
29	30	high	**753.2**	189.4	563.8	782.0	175.0	607.0	**753.2**	189.4	563.8	757.2	177.4	579.8
30	30	high	**799.5**	266.9	532.6	837.7	251.0	586.8	804.0	264.1	539.9	**799.5**	266.9	532.6
31	40	low	**854.0**	177.4	676.6	913.1	162.8	750.3	**854.0**	177.4	676.6	881.6	167.2	714.3
32	40	low	**868.6**	138.4	730.2	**868.6**	138.4	730.2	879.9	154.9	725.0	872.0	146.2	725.8
33	40	low	**825.3**	134.9	690.4	**825.3**	134.9	690.4	839.4	154.2	685.2	842.2	*	707.3
34	40	low	**832.5**	120.4	712.0	**832.5**	120.4	712.0	847.9	141.7	706.2	849.4	141.7	707.8
35	40	low	**881.8**	112.3	769.6	**881.8**	112.3	769.6	882.2	128.5	753.7	920.2	*	807.9
36	40	medium	**822.3**	187.8	634.5	873.9	168.1	705.8	**822.3**	187.8	634.5	856.3	*	688.2
37	40	medium	**853.6**	186.5	667.1	**853.6**	186.5	667.1	883.0	219.8	663.2	867.6	*	681.1
38	40	medium	**872.0**	167.9	704.1	912.5	155.1	757.5	**872.0**	167.9	704.1	925.7	*	770.6
39	40	medium	**863.7**	169.3	694.3	**863.7**	169.3	694.3	914.5	191.0	723.5	925.5	*	756.2
40	40	medium	**836.1**	170.4	665.7	**836.1**	170.4	665.7	867.8	195.5	672.2	864.2	*	693.8
41	40	high	**995.0**	277.3	717.6	998.7	274.1	724.6	1005.9	295.4	710.4	**995.0**	277.3	717.6
42	40	high	**976.1**	234.6	741.5	**976.1**	234.6	741.5	992.6	246.4	746.3	978.2	240.5	737.7
43	40	high	**875.9**	187.0	689.0	**875.9**	187.0	689.0	901.6	223.1	678.5	878.2	*	691.2
44	40	high	**962.2**	249.3	712.9	983.9	245.6	738.3	977.4	259.4	718.0	**962.2**	249.3	712.9
45	40	high	**1012.8**	223.7	789.1	**1012.8**	223.7	789.1	1026.9	233.2	793.7	1034.4	223.8	810.6
46	50	low	**1003.6**	185.1	818.5	1052.8	174.0	878.8	**1003.6**	185.1	818.5	1050.0	*	876.0
47	50	low	**982.0**	134.0	847.9	991.4	134.0	857.4	989.8	144.5	845.3	**982.0**	*	847.9
48	50	low	**1030.6**	168.2	862.4	1109.4	145.9	963.5	1068.7	157.9	910.8	**1030.6**	168.2	862.4
49	50	low	**1049.0**	139.1	909.9	**1049.0**	139.1	909.9	1101.6	154.7	947.0	1081.0	*	942.0
50	50	low	**971.6**	127.4	844.2	**971.6**	127.4	844.2	997.4	144.2	853.2	986.9	*	859.5
51	50	medium	**1146.6**	242.9	903.7	**1146.6**	242.9	903.7	1159.3	246.8	912.5	1168.4	*	925.5
52	50	medium	**1058.6**	204.1	854.5	1076.6	180.4	896.2	**1058.6**	204.1	854.5	1083.3	*	903.0
53	50	medium	**979.1**	144.1	835.0	**979.1**	144.1	835.0	998.8	165.0	833.7	993.0	*	848.9
54	50	medium	**1039.8**	160.2	879.6	1083.7	148.4	935.3	**1039.8**	160.2	879.6	1115.6	*	967.2
55	50	medium	**1009.9**	143.7	866.1	**1009.9**	143.7	866.1	1030.4	159.2	871.2	1041.5	155.2	886.3
56	50	high	**1204.4**	341.2	863.2	**1204.4**	341.2	863.2	1328.4	419.2	909.2	1260.8	*	919.6
57	50	high	**1165.1**	283.8	881.4	1198.5	283.8	914.8	1203.0	330.0	873.1	**1165.1**	*	881.4
58	50	high	**1124.4**	285.8	838.7	1127.1	260.5	866.6	**1124.4**	285.8	838.7	1151.7	*	891.3
59	50	high	**1074.4**	240.8	833.6	**1074.4**	240.8	833.6	1116.5	265.5	851.0	1106.2	247.5	858.7
60	50	high	**1163.7**	283.2	880.5	1167.8	283.2	884.6	1200.3	309.9	890.4	**1163.7**	*	880.5
61	100	low	**1988.6**	147.9	1840.7	**1988.6**	147.9	1840.7	2086.2	168.5	1917.7	1989.0	*	1841.0
62	100	low	**1881.5**	167.7	1713.9	**1881.5**	167.7	1713.9	1933.1	177.0	1756.1	1925.9	*	1758.2

Table 13: Best Solutions - large grid instances (continued)

Instances			Best known			Heuristic 1			Heuristic 2			Heuristic 3		
#	Size	Linehaul	TC	1^{st}E	RC	TC	1^{st}E*	RC	TC	1^{st}E	RC	TC	1^{st}E	RC
63	100	low	**2012.8**	204.3	1808.5	**2012.8**	204.3	1808.5	2057.5	237.1	1820.4	2029.6	233.4	1796.2
64	100	low	**2212.6**	359.3	1853.3	**2212.6**	359.3	1853.3	2284.7	411.5	1873.2	2266.6	*	1907.3
65	100	low	**2059.1**	286.8	1772.3	2087.8	257.2	1830.6	**2059.1**	286.8	1772.3	2074.4	*	1817.3
66	100	medium	**2066.8**	284.5	1782.3	**2066.8**	284.5	1782.3	2074.4	305.2	1769.2	2159.8	*	1875.3
67	100	medium	**2002.3**	305.1	1697.2	**2002.3**	305.1	1697.2	2044.2	337.6	1706.7	2038.6	*	1733.5
68	100	medium	**1950.1**	257.8	1692.3	**1950.1**	257.8	1692.3	2026.4	272.6	1753.8	1996.4	*	1738.6
69	100	medium	**2071.3**	351.4	1719.8	**2071.3**	351.4	1719.8	2163.5	359.6	1803.9	2141.5	*	1790.1
70	100	medium	**2088.0**	285.9	1802.1	**2088.0**	285.9	1802.1	2149.1	299.3	1849.8	2134.0	*	1848.1
71	100	high	**2232.6**	532.0	1700.7	**2232.6**	532.0	1700.7	2244.9	542.9	1701.9	2240.9	*	1708.9
72	100	high	**2310.4**	567.4	1743.1	2312.0	567.4	1744.7	2354.1	618.5	1735.6	**2310.4**	*	1743.1
73	100	high	**2036.1**	439.6	1596.4	**2036.1**	439.6	1596.4	2076.7	486.4	1590.4	2039.7	*	1600.1
74	100	high	**2311.4**	468.4	1843.0	**2311.4**	468.4	1843.0	2372.8	501.2	1871.6	2328.9	*	1860.5
75	100	high	**2280.5**	508.8	1771.7	2306.3	466.3	1839.9	**2280.5**	508.8	1771.7	2284.2	*	1817.9
76	150	low	**3032.6**	306.0	2726.6	**3032.6**	306.0	2726.6	3115.3	322.8	2792.4	3071.8	*	2765.8
77	150	low	**3118.1**	397.3	2720.8	3125.0	380.5	2744.5	**3118.1**	397.3	2720.8	3149.1	*	2768.5
78	150	low	**2930.5**	286.4	2644.2	**2930.5**	286.4	2644.2	2966.8	303.2	2663.6	2966.5	*	2680.1
79	150	low	**3246.2**	440.6	2805.5	**3246.2**	440.6	2805.5	3266.2	457.4	2808.7	3338.6	*	2897.9
80	150	low	**3178.5**	422.3	2756.2	**3178.5**	422.3	2756.2	3233.6	439.1	2794.6	3211.9	*	2789.6
81	150	medium	**3006.7**	429.3	2577.5	**3006.7**	429.3	2577.5	3161.4	476.0	2685.4	3052.1	*	2622.8
82	150	medium	**3167.9**	574.7	2593.2	**3167.9**	574.7	2593.2	3306.7	608.3	2698.4	3201.9	*	2627.1
83	150	medium	**3177.1**	418.7	2758.5	**3177.1**	418.7	2758.5	3323.0	439.1	2883.9	3199.9	*	2781.2
84	150	medium	**3118.9**	505.8	2613.0	**3118.9**	505.8	2613.0	3214.1	532.3	2681.7	3205.2	*	2699.4
85	150	medium	**3195.6**	531.9	2663.7	**3195.6**	531.9	2663.7	3316.1	565.5	2750.6	3289.6	565.5	2724.1
86	150	high	**3457.4**	783.2	2674.3	3493.6	713.6	2780.0	3531.6	747.4	2784.2	**3457.4**	783.2	2674.3
87	150	high	**3623.2**	802.9	2820.3	**3623.2**	802.9	2820.3	3657.7	862.5	2795.2	3638.2	*	2835.3
88	150	high	**3417.9**	803.5	2614.4	**3417.9**	803.5	2614.4	3492.6	841.1	2651.5	3465.2	*	2661.7
89	150	high	**3643.0**	848.4	2794.6	3699.4	848.4	2850.9	3787.5	917.8	2869.7	**3643.0**	*	2794.6
90	150	high	**3374.4**	784.9	2589.5	**3374.4**	784.9	2589.5	3536.5	821.0	2715.5	3443.0	*	2658.1
Total # of best known solutions found:						46			15			14		

* optimal solution MILP model if linehaul is solved first

Table 14: Worst Solutions Obtained - large grid instances

Instances			Best known			Heuristic 1			Heuristic 2			Heuristic 3		
#	**Size**	**Linehaul**	**TC**	**1^{st}E**	**RC**	**TC**	**1^{st}E***	**RC**	**TC**	**1^{st}E**	**RC**	**TC**	**1^{st}E**	**RC**
16	30	low	**707.8**	97.9	609.9	758.4	93.4	665.0	782.1	109.5	672.6	766.2	*	672.8
17	30	low	**660.3**	131.7	528.5	762.1	119.1	642.9	757.1	139.8	617.3	743.4	*	624.2
18	30	low	**626.0**	83.6	542.4	670.6	83.6	587.0	721.2	93.0	628.2	697.1	*	613.5
19	30	low	**689.6**	110.6	578.9	740.2	106.0	634.2	815.0	118.7	696.4	785.3	*	679.2
20	30	low	**631.4**	88.1	543.3	723.5	88.1	635.4	734.2	101.5	632.7	743.1	91.7	651.5
21	30	medium	**823.6**	204.9	618.7	854.2	204.9	649.3	893.3	242.1	651.2	865.9	*	661.1
22	30	medium	**777.1**	222.1	555.0	854.1	193.3	660.8	843.5	226.6	617.0	835.5	*	642.2
23	30	medium	**736.3**	160.3	576.1	819.8	160.3	659.6	830.0	178.0	652.0	793.8	*	633.6
24	30	medium	**690.5**	131.7	558.9	768.5	111.9	656.7	721.2	140.5	580.7	767.7	*	655.8
25	30	medium	**766.0**	171.5	594.5	831.3	145.5	685.8	830.1	167.4	662.7	832.8	*	687.3
26	30	high	**738.3**	189.1	549.3	805.6	189.1	616.5	830.0	217.2	612.9	812.4	192.8	619.6
27	30	high	**715.6**	178.6	537.1	758.5	178.6	579.9	837.2	191.6	645.6	817.5	187.9	629.7
28	30	high	**827.2**	233.0	594.2	859.5	233.0	626.6	956.0	273.6	682.5	864.3	*	631.3
29	30	high	**753.2**	189.4	563.8	825.6	175.0	650.6	797.6	198.3	599.3	835.5	176.5	659.0
30	30	high	**799.5**	266.9	532.6	877.3	251.0	626.3	861.2	256.4	604.8	891.3	*	640.4
31	40	low	**854.0**	177.4	676.6	982.3	162.8	819.5	947.5	177.4	770.1	1001.3	*	838.5
32	40	low	**868.6**	138.4	730.2	953.4	138.4	815.0	968.3	154.9	813.4	968.3	*	829.8
33	40	low	**825.3**	134.9	690.4	904.3	134.9	769.4	917.5	152.2	765.3	931.2	148.2	783.1
34	40	low	**832.5**	120.4	712.0	906.7	120.4	786.3	926.1	141.7	784.4	917.0	*	796.6
35	40	low	**881.8**	112.3	769.6	960.7	112.3	848.4	987.4	121.0	866.5	978.0	*	865.8
36	40	medium	**822.3**	187.8	634.5	924.0	168.1	755.9	940.6	179.0	761.6	932.9	*	764.8
37	40	medium	**853.6**	186.5	667.1	911.2	186.5	724.7	924.3	219.5	704.8	927.9	*	741.4
38	40	medium	**872.0**	167.9	704.1	1015.1	155.1	860.0	992.9	169.7	823.2	1016.8	*	861.7
39	40	medium	**863.7**	169.3	694.3	958.5	169.3	789.1	976.1	208.6	767.5	989.9	202.7	787.1
40	40	medium	**836.1**	170.4	665.7	914.1	170.4	743.7	931.4	194.1	737.3	934.0	186.0	748.0
41	40	high	**995.0**	277.3	717.6	1046.9	274.1	772.7	1114.4	311.8	802.6	1074.1	285.4	788.7
42	40	high	**976.1**	234.6	741.5	1034.7	234.6	800.1	1046.9	243.2	803.7	1037.8	*	803.2
43	40	high	**875.9**	187.0	689.0	928.9	187.0	741.9	953.5	205.1	748.4	936.7	*	749.8
44	40	high	**962.2**	249.3	712.9	1047.9	245.6	802.3	1059.8	256.2	803.6	1052.2	*	806.6
45	40	high	**1012.8**	223.7	789.1	1099.7	223.7	876.0	1097.7	231.0	866.7	1123.1	225.5	897.7
46	50	low	**1003.6**	185.1	818.5	1097.0	174.0	923.1	1117.5	185.1	932.3	1120.7	*	946.8
47	50	low	**982.0**	134.0	847.9	1068.0	134.0	934.0	1067.5	148.7	918.8	1054.5	*	920.5
48	50	low	**1030.6**	168.2	862.4	1160.6	145.9	1014.7	1126.5	157.9	968.5	1194.3	*	1048.4
49	50	low	**1049.0**	139.1	909.9	1146.5	139.1	1007.5	1189.2	154.7	1034.5	1191.8	*	1052.7
50	50	low	**971.6**	127.4	844.2	1067.9	127.4	940.6	1128.3	144.2	984.1	1070.5	143.1	927.4
51	50	medium	**1146.6**	242.9	903.7	1225.3	242.9	982.4	1239.8	269.5	970.3	1238.7	*	995.9

Table 14: **Worst Solutions - small grid instances (continued)**

Instances			Best known			Heuristic 1			Heuristic 2			Heuristic 3		
#	Size	Linehaul	TC	1^{st}E	RC	TC	1^{st}E*	RC	TC	1^{st}E	RC	TC	1^{st}E	RC
52	50	medium	**1058.6**	204.1	854.5	1144.8	180.4	964.4	1189.1	212.8	976.3	1146.1	*	965.8
53	50	medium	**979.1**	144.1	835.0	1071.9	144.1	927.8	1086.7	157.2	929.5	1072.0	156.7	915.3
54	50	medium	**1039.8**	160.2	879.6	1198.9	148.4	1050.4	1192.9	178.6	1014.3	1172.4	*	1023.9
55	50	medium	**1009.9**	143.7	866.1	1095.7	143.7	951.9	1123.0	161.1	961.9	1104.2	*	960.4
56	50	high	**1204.4**	341.2	863.2	1290.4	341.2	949.2	1420.2	420.6	999.6	1308.6	*	967.4
57	50	high	**1165.1**	283.8	881.4	1240.6	283.8	956.8	1337.8	324.8	1013.0	1259.3	*	975.6
58	50	high	**1124.4**	285.8	838.7	1190.1	260.5	929.6	1228.1	295.5	932.6	1226.6	286.1	940.4
59	50	high	**1074.4**	240.8	833.6	1248.0	240.8	1007.1	1144.7	247.7	897.0	1166.6	249.3	917.3
60	50	high	**1163.7**	283.2	880.5	1251.7	283.2	968.4	1283.9	338.6	945.3	1273.7	292.1	981.6
61	100	low	**1988.6**	147.9	1840.7	2154.6	147.9	2006.7	2280.1	161.3	2118.7	2151.9	169.8	1982.1
62	100	low	**1881.5**	167.7	1713.9	2082.6	167.7	1914.9	2037.2	179.1	1858.1	2058.4	*	1890.8
63	100	low	**2012.8**	204.3	1808.5	2162.0	204.3	1957.7	2155.1	239.7	1915.4	2181.1	233.1	1947.9
64	100	low	**2212.6**	359.3	1853.3	2340.9	359.3	1981.6	2457.1	416.5	2040.5	2416.0	400.8	2015.2
65	100	low	**2059.1**	286.8	1772.3	2206.9	257.2	1949.8	2263.8	284.5	1979.3	2255.3	288.1	1967.2
66	100	medium	**2066.8**	284.5	1782.3	2166.0	284.5	1881.5	2201.0	305.5	1895.5	2256.3	291.4	1964.9
67	100	medium	**2002.3**	305.1	1697.2	2160.1	305.1	1855.0	2274.3	348.4	1925.9	2224.7	322.9	1901.8
68	100	medium	**1950.1**	257.8	1692.3	2088.9	257.8	1831.1	2148.7	285.4	1863.3	2146.2	*	1888.3
69	100	medium	**2071.3**	351.4	1719.8	2200.0	351.4	1848.5	2251.5	373.6	1877.9	2238.9	373.2	1865.7
70	100	medium	**2088.0**	285.9	1802.1	2215.9	285.9	1930.0	2299.5	320.2	1979.3	2260.2	334.5	1925.6
71	100	high	**2232.6**	532.0	1700.7	2361.0	532.0	1829.0	2449.7	548.0	1901.7	2398.3	548.1	1850.3
72	100	high	**2310.4**	567.4	1743.1	2414.0	567.4	1846.6	2590.7	607.8	1982.9	2483.5	620.7	1862.8
73	100	high	**2036.1**	439.6	1596.4	2202.6	439.6	1763.0	2362.5	517.0	1845.4	2276.2	479.9	1796.4
74	100	high	**2311.4**	468.4	1843.0	2444.9	468.4	1976.6	2490.7	486.4	2004.4	2517.3	505.4	2011.8
75	100	high	**2280.5**	508.8	1771.7	2455.1	466.3	1988.8	2492.8	506.3	1986.5	2454.5	475.2	1979.4
76	150	low	**3032.6**	306.0	2726.6	3265.4	306.0	2959.4	3409.5	322.8	3086.7	3303.4	*	2997.4
77	150	low	**3118.1**	397.3	2720.8	3300.0	380.5	2919.5	3316.1	405.6	2910.5	3264.2	403.5	2860.6
78	150	low	**2930.5**	286.4	2644.2	3146.5	286.4	2860.1	3173.9	312.5	2861.4	3192.2	*	2905.8
79	150	low	**3246.2**	440.6	2805.5	3542.6	440.6	3101.9	3515.2	457.4	3057.7	3483.1	457.4	3025.6
80	150	low	**3178.5**	422.3	2756.2	3330.4	422.3	2908.1	3464.3	439.1	3025.2	3424.9	455.9	2969.0
81	150	medium	**3006.7**	429.3	2577.5	3202.0	429.3	2772.7	3337.0	484.7	2852.2	3308.4	*	2879.1
82	150	medium	**3167.9**	574.7	2593.2	3358.0	574.7	2783.2	3429.9	609.5	2820.4	3411.9	*	2837.2
83	150	medium	**3177.1**	418.7	2758.5	3399.3	418.7	2980.6	3439.2	434.5	3004.7	3435.8	*	3017.1
84	150	medium	**3118.9**	505.8	2613.0	3386.6	505.8	2880.8	3582.6	539.4	3043.2	3440.3	*	2934.5
85	150	medium	**3195.6**	531.9	2663.7	3461.9	531.9	2930.0	3563.1	565.5	2997.5	3514.2	548.9	2965.2
86	150	high	**3457.4**	783.2	2674.3	3669.6	713.6	2956.0	3715.1	780.8	2934.3	3756.2	790.7	2965.5
87	150	high	**3623.2**	802.9	2820.3	3811.2	802.9	3008.2	3878.8	855.4	3023.4	3862.7	872.0	2990.7

Table 14: **Worst Solutions - small grid instances (continued)**

Instances			**Best known**			**Heuristic 1**			**Heuristic 2**			**Heuristic 3**		
#	**Size**	**Linehaul**	**TC**	1^{st}**E**	**RC**	**TC**	1^{st}**E***	**RC**	**TC**	1^{st}**E**	**RC**	**TC**	1^{st}**E**	**RC**
88	150	high	**3417.9**	803.5	2614.4	3609.8	803.5	2806.3	3748.0	840.6	2907.4	3693.6	*	2890.1
89	150	high	**3643.0**	848.4	2794.6	3849.9	848.4	3001.5	4149.7	956.8	3192.8	3962.9	972.8	2990.1
90	150	high	**3374.4**	784.9	2589.5	3613.1	784.9	2828.2	3744.0	822.1	2921.9	3755.4	855.3	2900.0

* optimal solution MILP model if linehaul is solved first