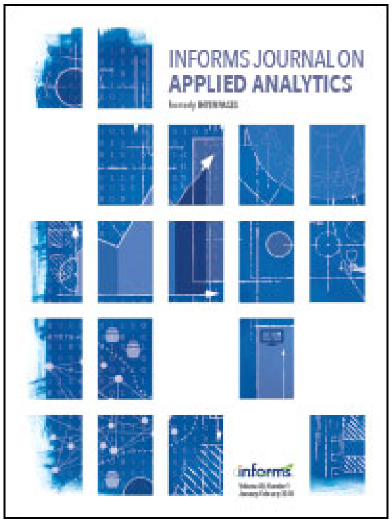
Optimizing Mobility for Elderly and Disabled Dutch Citizens Using Taxis

INFORMS Journal on Applied Analytics

de Ruiter, Frans J.C.T.; van Rooij, Johan M.M.; Hulsen, Peter; Post, Bart; Goes, Jeroen et al

https://doi.org/10.1287/inte.2024.0180



INFORMS Journal on Applied Analytics

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Optimizing Mobility for Elderly and Disabled Dutch Citizens Using Taxis

Frans J. C. T. de Ruiter; Johan M. M. van Rooij; , Peter Hulsen; , Bart Post; , Jeroen Goes; , Geert Teeuwen; , Matthijs Tijink; , Bart Verberne; , Niels Bourgonjen; , Roelf Nienhuis, Tjeerd van der Poel, Laurens van Remortele

informs.
https://pubsonline.informs.org/journal/ijaa

INFORMS JOURNAL ON APPLIED ANALYTICS
Vol. 55, No. 1, January–February 2025, pp. 66–82
ISSN 2644-0865 (print), ISSN 2644-0873 (online)

THE FRANZ EDELMAN AWARD
Achievement in Operations Research

Optimizing Mobility for Elderly and Disabled Dutch Citizens Using Taxis

Frans J. C. T. de Ruiter,[a,b,*] Johan M. M. van Rooij,[c] Peter Hulsen,[a] Bart Post,[a] Jeroen Goes,[a] Geert Teeuwen,[a] Matthijs Tijink,[a] Bart Verberne,[a] Niels Bourgonjen,[d] Roelf Nienhuis,[e] Tjeerd van der Poel,[e] Laurens van Remortele[e]

[a] Consultants in Quantitative Methods (CQM), 5616 RM Eindhoven, Netherlands; [b] Department of Operations Research and Logistics, Wageningen University, 6706 KN Wageningen, Netherlands; [c] Department of Information and Computing Sciences, Utrecht University, 3584 CC Utrecht, Netherlands; [d] Geodan (part of Sogelink), 1079 MB Amsterdam, Netherlands; [e] Transvision B.V., 2909 LC Capelle aan den IJssel, Netherlands

*Corresponding author

Contact: frans.deruiter@wur.nl, https://orcid.org/0000-0002-4311-0651 (FJCTdR); j.m.m.vanrooij@uu.nl, https://orcid.org/0000-0001-9149-4162 (JMMvR); bart.post@cqm.nl, https://orcid.org/0000-0002-4950-3464 (BP); matthijs.tijink@cqm.nl (MT)

Accepted: October 25, 2024

https://doi.org/10.1287/inte.2024.0180

Abstract. In the Netherlands, 200,000 elderly and disabled citizens annually use subsidized taxi rides executed by Transvision. The day-to-day planning of up to 15,000 long-distance rides was previously a complex and daunting task split over dozens of subcontractors. Transvision, CQM, and Geodan developed an optimization solution that combines the rides into efficient taxi routes. Starting in January 2020, this solution significantly improved the mobility challenge for elderly and disabled citizens, including (1) increased punctuality and a 50% improvement in passenger satisfaction, (2) savings of 15 million driving kilometers per year, and (3) combined financial savings for all stakeholders of 60 million euros over the years 2019 to 2023 and another total of 30 million euros projected for 2024 and 2025, according to conservative estimates. Daily planning in a single batch can range from 1,000 to 15,000 rides. To construct high-quality ride plans in reasonable time for this massive-scale operations research problem, we applied classical operations research techniques viewed through a modern lens. In this paper, we explain how practical large-scale dial-a-ride problems can be solved using high-quality heuristics that exploit the power of parallel processing. Furthermore, we present new and efficient techniques to perform the required millions to billions of calculations to determine distances and driving times on the Dutch road network. We overcome several practical challenges such as (1) aligning the interests of a vulnerable passenger group and over 60 different taxi operators, (2) aligning the software that interfaces with the various companies, and (3) adapting to changing regulations and ad hoc COVID-19 measures.

Keywords: dial-a-ride problem • disability transit • sustainable development goals • large-scale optimization • simulated annealing • shortest-path calculation • paratransit • distance matrix • Edelman award

Introduction and History

Public transport by bus or train can pose significant barriers for elderly and disabled people. For example, a route may contain wheelchair-inaccessible areas, and people with cognitive impairments may face obstacles that are impossible for them to overcome. To give elderly and disabled citizens the same transit opportunities as other citizens, the Netherlands offers subsidized government programs using taxis. Similar programs also exist in many forms in other countries worldwide; for example, in the United States, they are called "paratransit." Accessible transportation for elderly and disabled people is deemed essential to enable them to participate actively in society and is part of the United Nations Sustainable Development Goal 11 ("sustainable cities and communities"), in particular, measurable indicator 11.2.1 (United Nations 2015), which was adopted in 2015: "Proportion of population that has convenient access to public transport, by sex, age and disability status, also distinguishing older people."

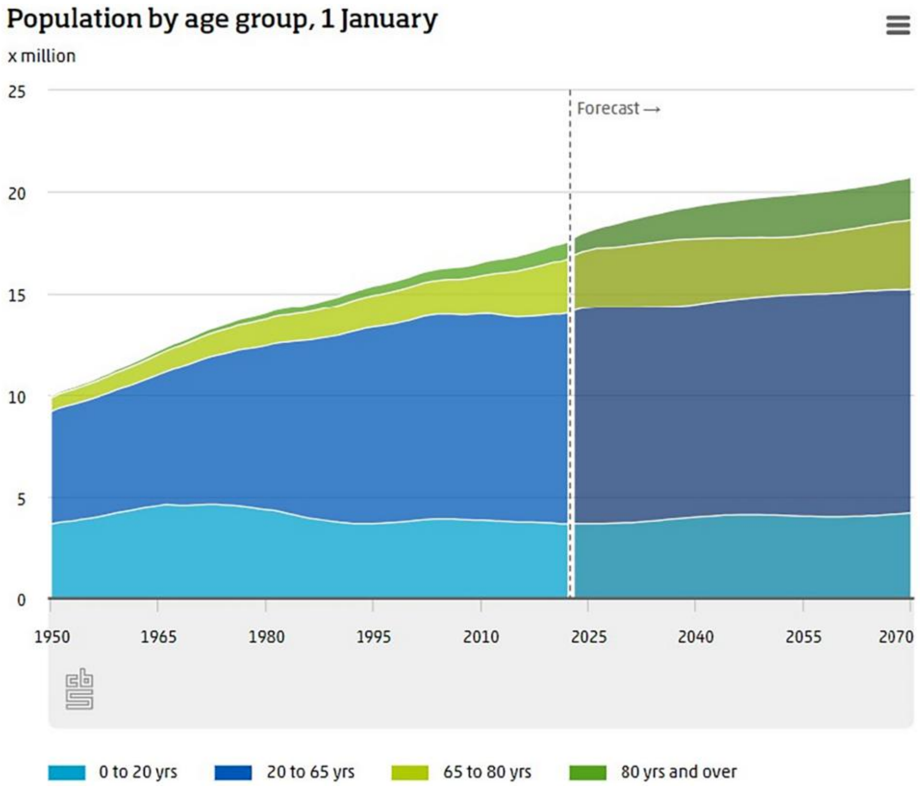
In 2020, the United Nations passed a resolution endorsing the plan Decade of Healthy Ageing by the World Health Organization, which strives to improve transport for this group of people (United Nations 2020).

Starting in 1995, local Dutch municipalities were obliged to provide "interregional" long- distance transportation for elderly and disabled people in addition to short-distance rides within a region. These long-distance rides could be used, for example, to visit friends and family, attend funerals, or travel to job interviews. Since 1998, interregional rides have been organized on a national level, and in 2004, the interregional transportation program was rebranded as "Valys." The Valys program is by far the largest subsidized taxi program for elderly and disabled citizens in the Netherlands. Under this contract, which is granted every four years, over one million long-distance taxi rides are used to transport 200,000 elderly and disabled citizens annually. As the population ages, the size of the program will only increase in the future (see Figure 1). Over a quarter of the population is forecasted to be 65+ years old in 2040, and the number of people 80 years or older is projected to double by 2043, from 0.9 million at the end of 2022 to 1.7 million in 2043.

In 2014, the mobility company Transvision won the Valys contract. Transvision focuses on organizing taxi transport for people who require assistance during their rides, and it subcontracts taxi companies for the execution of the transport. Transvision handles the entire process, including reservations, scheduling, dispatching, and communicating estimated times of arrival. Initially, the Valys rides were assigned to subcontracted taxi companies based on the area in which the taxi companies operated. Planners at these companies were responsible for incorporating these rides into their operational schedules. Consequently, because of this decentralized approach and the typical long-distance nature of these rides, taxi drivers often had to drive far outside their usual operating areas. This often forced them to drive a return trip without passengers and thus not receive any payment for a costly trip. This made these rides very unattractive for the taxi companies. From the start, Transvision aimed to address this inefficiency and the inefficiency of taxis having excess capacity when transporting only a few passengers at a time. To do so, Transvision tried to incorporate operations research (OR) into a business that is known for its conservative stance toward technology. Its first attempts failed, not only because entering anything more than a list of single rides into the mandatory onboard computers in taxis proved to be difficult but, more importantly, because the solutions presented at that time failed to meet the efficiency for which Transvision aimed.

Switching to an OR-driven plan required a change of mentality for both Transvision and its subcontractors. A central planning system using OR techniques had to be extremely efficient and achieve positive benefits for all parties if it were to succeed. Hence, in the years following Transvision's winning of the Valys contract in 2014, the scheduling remained decentralized and

Figure 1. (Color online) An Aging Population in the Netherlands Will Increase Pressure to Provide Efficient Operations and Budgeting of Transport for Elderly Citizens



Source. Statistics Netherlands (2022). Permission conveyed under Creative Commons Attribution CC BY 4.0 (https://www.cbs.nl/en-gb/about-us/website/copyright).

distributed among all of Transvision's 60 subcontractors. Each subcontractor was responsible for planning its small, isolated share of the daily rides that originate in its area of operation (yet their associated drop-off locations may be scattered across the Netherlands!).

Despite the failure of its first attempt, Transvision persisted and won the next Valys contract in 2018 by promising to execute it at a cost level that was only achievable by developing very efficient scheduling plans. Therefore, Transvision organized a competition using its previous learnings, and several OR companies and data science companies participated. One of the participators was a joint collaboration between Geodan and CQM. Geodan provides location intelligence, that is, high-quality maps with historical speeds and accurate geocoding of addresses, to a large number of clients in the Netherlands. CQM is a data science consultancy company that was spun off from Philips more than 40 years ago. CQM specializes in developing OR solutions and ensuring that they work in practice. The joint efforts of Geodan and CQM won the new competition organized by Transvision by a large margin, realizing a level of efficiency that was previously thought impossible: Geodan and CQM proved that savings of approximately 100,000 kilometers on a typical day could be achieved with OR solutions. The outcome of this competition fueled the Transvision team's initial confidence to fully exploit OR solutions.

This led to the development of our OR solution for Valys taxi planning. Participants in the Valys program are financially encouraged to place their bookings of interregional rides before a specified cutoff time; an additional fee applies to late bookings. The cutoff time is scheduled at the end of the day prior to the day of the ride to facilitate maximal flexibility in booking while also allowing for sufficient computation time to construct an efficient schedule. After this cutoff time, the set of booked rides is presented to our algorithm. The time window to perform computations is limited; the subcontractors must be given sufficient time to incorporate the resulting plan into their own operational planning for the next day because they typically have many other rides to execute outside the scope of Valys. Therefore, we allow the optimization process to take a maximum of only one hour of computation time. Our solution, which requires all possible point-to-point distances and travel times with realistic speed estimates, minimizes the time taxis need to be operated while satisfying a large number of operational constraints. To calculate all these options is a huge feat given the size of the problem and the limited computation time. At the completion of the computation (i.e., after one hour), the resulting schedule is communicated to the subcontracting taxi companies. This communication is done by directly interfacing the OR solution with the administrative planning software of the various subcontracting taxi companies.

Challenges and Need for Operations Research

Planning Valys routes is a nationwide integrated problem involving between 2,000 and 5,000 rides on normal days, with spikes as high as 15,000 rides on holidays such as Christmas Day; for execution, it involves 60 subcontractors and more than 1,000 taxis and drivers. This scale is unheard of in most other taxi-planning applications, which are restricted to either a municipality or region or have a more direct on-demand (and thus local) nature and involve only one or a few subcontractors. This scale presented several major challenges for planning and aligning stakeholders, as we describe below.

1. Different standards for punctuality. To provide elderly and disabled citizens transport that is comparable to public transport, it is important that taxis arrive on time and that a passenger's total time from pickup to drop-off is limited in that the maximum allowable travel time is reasonable. Scheduling rides to achieve high punctuality is difficult because of the uncertainty in travel and boarding times. In 2019, each subcontractor had its own method of estimating travel times, with or without digital aids for calculations. Predicting whether passengers would be picked up on time was often difficult. Even more difficult was determining whether scheduling a detour to pick up an additional passenger would violate the maximum allowable travel time for passengers already scheduled in the car (i.e., 150% of the minimum needed travel time between pickup and drop-off). Hence, in a substantial number of cases, without intent, the maximum allowable time would be exceeded.

2. Daunting and complex planning. Typically, each subcontractor has one or more planners tasked with scheduling its daily operations, which typically include many rides outside the scope of Valys. To efficiently include the Valys rides into their schedules, planners from each of the 60 subcontractors had to plan a small share of the Valys rides manually. However, given the exceptionally large number of rides scheduled through Valys (as we mention above), this "small share" could still be hundreds of rides daily per subcontractor.

Before the implementation of our centralized optimization, all rides assigned to a subcontractor had pickups in that subcontractor's operating area, and drop-off locations were typically spread across the Netherlands and could be far outside a subcontractor's normal operating area. In an attempt to minimize empty driving, subcontractors were allowed to exchange rides with other subcontractors. However, the process consisted of calling, informing, and negotiating an exchange of rides with other subcontractors. This required significant effort by a subcontractor, with little guarantee that an exchange proposal would be accepted. Another option that planners had was that they could try to look for rides that were within their own portfolio and had geographically close pickup and drop-off locations, but with slightly mismatching time windows. In such cases, a subcontractor could inquire whether a passenger was willing to change the pickup time. Again, this required laborious effort, with little guarantee that a passenger was able and willing to accept the change.

We foresee other challenging planning questions in the near future. New tenders written by the Dutch government focus on, or already mandate, the use of electric vehicles for taxi transport. Manual planning will then become even more complex because it will involve incorporating suitable charging times into driver schedules, which is further complicated by the fact that charging can only be performed at very specific locations (i.e., charging stations).

3. Driver shortages. The available taxi-driver workforce is extremely small; thus, efficient use of the drivers' time is paramount to success. In recent years, taxi drivers became so scarce that other important subsidized mobility programs, such as school transport for disabled children, became unreliable or even unavailable at times (Venneman and de Gruijter 2022). This scarcity is inherent in any subsidized program using taxis, although the severity of the shortage differs among programs. The importance of technology for efficiency of this workforce is therefore endorsed by the Dutch Social Economic Council (De Sociaal-Economische Raad 2023). To sustain the programs now and in the future, drivers must have agreeable working conditions (e.g., sufficient breaks, tolerable stress, predictable schedules), and their time must be used efficiently.

4. Large volatile demand. Valys is mainly used for (social) participation of elderly and disabled citizens, for example, to visit friends, family, or amusement parks or to travel to job interviews and funerals. These rides are ad hoc in nature and do not show any recurring pattern. The total number of bookings in the Netherlands varies per day. Typically, more rides are booked on weekends than on weekdays, and they peak on holidays. Although, on a national level, the daily number of bookings is fairly consistent, the regional demand is much more volatile. Therefore, up until 2019, with a decentralized planning system, one subcontractor could be extremely busy, whereas a neighboring subcontractor had idle taxis. Looking ahead, overall demand and volatility are likely to increase further because of the aging population. In addition, in December 2023, the Dutch parliament approved a resolution to remove the upper bound on the number of kilometers passengers can use for Valys (Helder 2023), which will likely lead to an even larger number of rides.

5. Increased environmental standards. The amount of CO_2 emissions that result from a mobility contract the size of the Valys contract is significant. In a typical year, the total sum of booked kilometers between pickup and drop-off is around 50 million. Using our optimistic estimate of efficient manual planning (see Table 1 for details), this yields a staggering 85 million kilometers driven annually. This, combined with CO_2 emissions of 156 grams per kilometer, as estimated by van Gijlswijk et al. (2022), yields more than 13,000 tons of CO_2 emission per year.

In addition to CO_2 emissions, nitrogen emissions present another environmental challenge. The Netherlands is in the middle of a "nitrogen crisis." This crisis is the consequence of the country being burdened with extremely high levels of reactive nitrogen compounds from intensive agriculture farming and combustion engines. This crisis came to light in 2019 because the country far exceeded European norms and harsh measures had to be taken; examples include reducing the speed limit on highways to 100 kilometers per hour, paying farmers large sums of money to stop farming, halting housing development projects, and reducing air traffic around Amsterdam. All national road transport (via nitrogen emissions) consists of only 6.1% of the problematic nitrogen deposition, according to the crisis commission set up by the government (Remkes et al. 2020). Nevertheless, the government argues that any single sector cannot solve the crisis on its own. Hence, nitrogen reductions are to be sought by different means across sectors, including transport.

Table 1. Comparison of an Optimized Plan vs. a Fully Individualized (Singular) Ride and a Simulated Best Plan That a Subcontractor Could Use During an Average (i.e., Non-COVID-19) Month

	Planning by:		
	Optimization	Singular	Subcontractor simulation
Percentage of rides in a multiple-ride route	85%	0%	65%
Combination ratio	0.89	0.5	0.59
Hourly paid kilometers	62	34	39

Notes. A fully individualized (singular) ride is a ride for which a taxi operator picks up a single passenger, drops off that passenger at his/her destination, and drives back empty (i.e., without passengers) to the operator's home base. The combination ratio is the total paid kilometers divided by the total driving kilometers. Hourly paid kilometers is total paid kilometers divided by total driving time in hours.

6. Software interfacing. In the Netherlands, every taxi is legally required to have an approved onboard computer to register all rides and communicate with central planning systems. Historically, multiple taxi registration systems have been designed primarily for assigning individual rides to drivers. These systems must now integrate with our new OR solution, which does not communicate single rides but, rather, an ordered sequence of pickups and drop-offs to execute. For this solution to succeed in practical scenarios, it must interface effectively with these diverse systems, employing various techniques to ensure each system operates efficiently in the new context.

Impact

The OR solution that has been in production since 2020 will have a profound impact on the viability of the Valys program in the future, keeping accessible transport available and affordable for elderly and disabled Dutch citizens in a rapidly aging population. It has provided direct benefits from the start of its implementation. Most notably, these benefits include improved punctuality of scheduled rides for the 200,000 active passengers, reduction of CO_2 emissions, financial gains, and reduction of manual planning efforts. These benefits are described in more detail in the next sections.

Service Impact: On-Time Performance and Reduced Maximum Ride Time

When the OR solution was introduced in January 2020, we standardized driving times using realistic historical driving speeds on each road segment. This directly resulted in improved on-time performance and better adherence to the restriction specifying the maximum time a passenger can spend in a car. As a result, we saw a 50% decrease in the percentage of rides that resulted in registered complaints in the first two months of 2020. This is the most comparable period for evaluating the OR solution versus manual planning. Beginning in March 2020, bookings were not allowed for two months because of the COVID-19 pandemic; this was followed by a period in which additional restrictions on taxi occupation and additional cleaning rules were applied. Despite these new rules, passenger satisfaction remained high; however, given the COVID-19 reality, distinguishing the effect of the OR solution on passenger ride evaluations is difficult.

The optimization algorithm attempts to combine multiple consecutive pickups and drop-offs, which results in significant pooling and therefore detours for passengers. Pooling is allowed and was also applied prior to 2020 for rides departing at approximately the same time and from the same geographic location in a subcontractor's area. One might think that an OR solution that minimizes driving kilometers would result in an increase in the average time spent passengers spend in a taxi because of extensive pooling. However, we see from our data that the average time increased only marginally. The difference in average time that a passenger spent in the taxi before 2020 and after the new solution became operational in 2020 is less than one minute because of our strict adherence to our limitation on the maximum time a passenger can spend in a taxi. This limits the pooling possibilities because our driving time estimates do not allow (unintentionally) optimistic estimates when pooling appears convenient.

Environmental Impact: Reducing CO_2 and Nitrogen Emissions

Our OR solution reduces the number of kilometers that taxis are driven on a typical day by approximately 100,000 kilometers. The CO_2 emissions saved are more than 15 tons of CO_2 on a typical day (computed using 156 grams of CO_2 per kilometer, as we discuss in the section titled Challenges and Need for Operations Research). Using a very conservative estimate of at least *50,000* kilometers per day on average, as we discuss in "A Novel Pricing Mechanism to Benefit Subcontractors" in the section titled Reflections on the Implementation Journey, to compare the OR solution with manual planning and manual ride exchanges, we project to save 90 million kilometers over the six-year contract period, the equivalent of more than 14,000 tons of CO_2. In addition to CO_2 emissions in this period, this reduction also contributed to decreasing nitrogen emissions, thus offering a modest but valuable contribution to addressing the nitrogen crisis. Furthermore, our OR solution can be adapted to facilitate the anticipated transition to electric taxis, which produce zero nitrogen emissions. To achieve this, the algorithm has to take into account both the time for charging and driving to the charging locations. Geodan has accurate data on the (fast-) charging locations across the Netherlands. The algorithm can be adapted to plan these charging times for long-distance routes based on the charging locations and, if possible, let them coincide with regulatory break periods for drivers.

Financial Impact: Transvision and Other Stakeholders

The execution of the Valys program is the responsibility of commercial parties such as Transvision and its subcontractors, although the government subsidizes the program. Therefore, any real sustainable solution should be underpinned by financial benefits for all stakeholders: the government, Transvision, and the subcontractors. This was achieved in the implementation trajectory through a novel pricing mechanism in which prices were determined by the efficiency of the plan. We explain the benchmarking of saved kilometers and the integrated working of the pricing mechanism in the section titled Reflections on the Implementation Journey.

Total direct financial benefits would have been higher had we not had to deal with the COVID-19 pandemic, which led to a temporary reduction in rides for the vulnerable group of elderly and disabled citizens. Taking this into account, the combined financial benefits were on average 15 million euros per year, which is 20% of the annual 75 million euros that the program would have cost the government without the OR solution. This is based on an average variable cost of one euro per driven kilometer (for fuel, taxi driver, and variable maintenance). Details on kilometer savings are given in the section titled A Novel Pricing Mechanism to Benefit Subcontractors. Using the same conservative estimate of 50,000 driving kilometers per day as above, this yields over 15 million kilometers per year. Over almost six years, because the contract has recently been extended to mid-2025, the cumulative savings are therefore projected to be 90 million euros. The financial benefits consist of several components:

1. The initial belief in OR efficiency led to Transvision's offering a more competitive tender price, which exceeds 10% of the price for which the company could run a sustainable business without OR. This directly benefits society because, for the same budget, more passenger kilometers can be subsidized by the program.

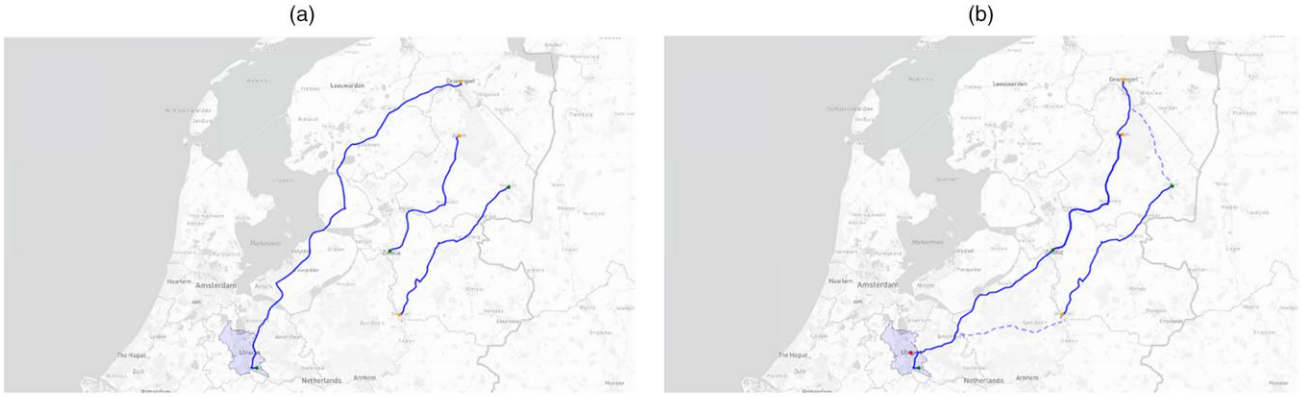
2. The remaining (conservative) approximate benefits of 10% are allocated to Transvision and its subcontractors by exceeding the initial anticipated efficiency gains. By driving fewer kilometers than anticipated, the operational costs decreased further than we initially thought possible. The division of these benefits between Transvision and subcontractors is detailed in the section titled A Novel Pricing Mechanism to Benefit Subcontractors. These gains enabled investments in electric vehicles and the modernization of the wheelchair taxi fleets of the various subcontractors.

Planning Impact: A Fair and Evenly Distributed Load

The reduction in demand volatility is a significant benefit for planners. Prior to the optimization solution, subcontractors faced uncertainty because of the unpredictability of the total number of rides nationwide, amplified by regional demand variations. With our novel fairness criteria, as we explain in the section titled Applying Fairness below, we guide the solution to distribute the workload among subcontractors based on their average historical share. This approach minimizes the impact of regional demand variations and enhances the predictability of the number of rides each subcontractor is assigned. Previously, during busy summer months, the share of rides among subcontractors varied significantly. Our fairness mechanism allocates a more equal share of the total rides from day to day for each subcontractor. This comparison yields, on average, a reduction in standard deviation of the day-to-day share of rides by 36% compared with the initial allocation method based on zip code areas.

Planners at subcontractors can now focus on selecting the right drivers and assigning routes that match driver preferences. They no longer need to spend time coordinating with passengers to adjust time windows or manually exchange rides. The analytical algorithm effectively searches for rides outside a subcontractor's area once a route departs from that area. This enables combinations, such as the one we show in Figure 2. Combinations such as these would be difficult to identify manually in the previously applied distributed scheduling approach; this is in addition to challenges encountered in obtaining approval from all involved subcontractors and passengers for such exchanges.

Figure 2. (Color online) Maps Showing an Example of Three Individual Routes and an Optimized Combined Route



Notes. (a) We see three individual routes, from which only one is visible from the viewpoint of the subcontractor operating in the Utrecht area. (b) We see a route that combines all three individual routes into one more efficient optimized route. Determining this optimized route manually would require finding the nontrivial route as well as negotiating ride exchanges among many subcontractors.

Our Operations Research Approach to Large-Scale Mobility

Our OR solution consists of the steps depicted in the pipeline in Figure 3. After the cutoff time, the set of known booked rides enters this pipeline resulting in an optimized plan computed within a time limit of one hour. This time limit is based on (1) being able to set the cutoff as late as possible, thus avoiding manual changes to the plan; and (2) giving the taxi companies enough time to coordinate their operations for the next day.

We take the following steps in our OR solution pipeline:

1. Geocoding. In the geocoding process, Geodan assigns precise latitudes and longitudes to each pickup and drop-off address. This ensures that we have accurate locations on the map as a starting point, which is fundamental for calculating routes. It allows for more accurate distance and travel-time calculations, directly impacting the effectiveness of our solution.

2. Calculating distances and travel times. Each location is mapped to the road network. With a new state-of-the-art distance-matrix computation engine developed by CQM for this project, we calculate a few hundred million to even billions of point-to-point distances and travel times within the limited computing time available. The distance matrix is essential because it enables us to store all point-to-point distances and travel times and make them easily available to our optimization algorithm.

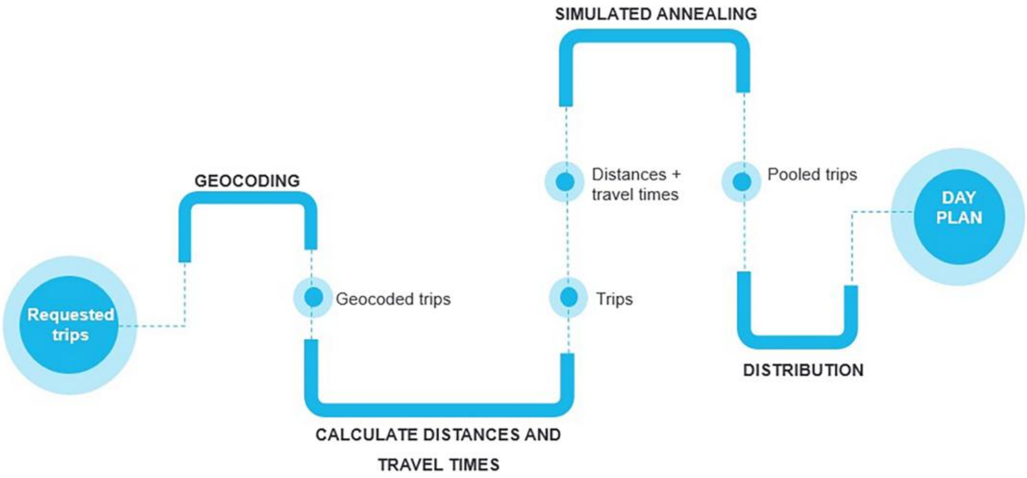
3. Simulated annealing. To address the specific large-scale dial-a-ride problem (DARP), we employ a classic simulated annealing algorithm, viewed through a modern lens. Our strategy involves constructing a simplified solution space and using a highly efficient parallel implementation. We also tailored the neighborhood operations to prioritize feasible and effective solutions. This strategy allows us to generate high-quality solutions within a mere 15 minutes, achieving up to four million iterations per second using just eight CPU threads.

The output of this pipeline is a final optimized plan. This plan is then distributed to the considerable number of software modules used by the various subcontractors and finally to the onboard computers in the taxis. Below, we consider the second and third steps of the pipeline, which form the heart of the OR model. That is, we first describe our solution to solve a large-scale dial-a-ride problem in the section titled Solving the Dial-a-Ride Problem Using Simulated Annealing and then our large-scale distance matrix computation in the section titled Very Large-Scale Distance Matrix Computation.

Solving the Dial-a-Ride Problem Using Simulated Annealing

The problem of planning the Valys paratransit service is closely related to the DARP in the OR literature; see Cordeau and Laporte (2007), Baldacci et al. (2012), and Molenbruch et al. (2017) for surveys. In this problem, passengers specify a desired pickup (or drop-off) time, and vehicles must be efficiently routed to transport all passengers from their starting points to their destinations. The objective in solving a DARP is to minimize the cost for operating the vehicles where pooling of rides is allowed. That is, multiple unrelated passengers are allowed to be transported simultaneously in the same vehicle on a route as long as for each passenger, a maximum time the delay may cause is respected. In our case, we minimize the operating time required by the vehicles to transport the passengers. Operational and service constraints apply in a DARP; in our case, these include time window constraints around the specified times, a maximum shift duration, mandatory breaks for the drivers, different vehicle types, a constraint that vehicles should return to their origin areas,

Figure 3. (Color online) Pipeline Showing the Consecutive Computing Steps Needed to Transform Next-Day Requests into an Optimized Ride Plan



and additional constraints for wheelchair passengers and guide dogs (e.g., passengers with guide dogs must sit in the passenger front seat).

We consider the *static* DARP variant because all rides are known at least one day in advance. Exact methods for this problem can only solve instances up to a few hundred rides at most; see Cordeau (2006) and Schulz and Pfeiffer (2024). For larger instances such as ours, a common approach is to use heuristics such as an insertion heuristic; see Diana and Dessouky (2004) and Markovic et al. (2015). The tabu search in Cordeau and Laporte (2003) and Jain and Van Hentenryck (2011) uses a large neighborhood search. Other approaches include variable neighborhood search (Parragh et al. 2010) and the combination of clustering with exact methods (Bertsimas and Yan 2020). Cummings et al. (2023) consider a two-stage dial-a-ride variant without pooling, which they plan to be robust against a limited number of cancellations. In addition, Kuijpers (2023) considers quick reoptimization after new bookings and cancellations using GRASP, yet also concludes that GRASP does not provide the same high-quality first-stage solutions as our approach. Note that for our Valys problem, the number of last-minute cancellations is low, and most of the efficiency is gained through optimal planning the previous day.

We emphasize that, with the exception of Kuijpers (2023), which considers our Valys problem, none of the approaches we discuss above was built for the scale applied here because they consider a few hundred to at most 1,000 to 2,000 rides. These numbers are dwarfed by Valys, which has instances of 5,000 rides on a weekend day while peaking at 15,000 on holidays such as Christmas Day. The only paper in the literature we know of that comes close to these numbers is Bertsimas and Yan (2020), who consider 3,000 to 7,000 rides, but they do not consider a road network–based distance matrix, and they divide the problem into multiple subproblems, which seems very restrictive to our case because of the nature of multiple-city interregional transportation. We solve the DARP without division in subproblems using the classic simulated annealing heuristic viewed through a modern lens.

The Simulated Annealing Heuristic. Simulated annealing is a local search method that maintains a current solution to the problem. In each iteration, the simulated annealing algorithm considers a random neighboring solution to the current solution defined through a set of neighborhood operations that slightly perturb the current solution. If the neighboring solution has a better objective value, then it is accepted as the new current solution. If not, then a random number r from the interval [0,1] is drawn uniformly, and the neighboring solution is accepted when $r < e^{-\Delta E/T}$, where ΔE is the difference in cost between the current solution and the neighboring solution, and T is the temperature parameter. If it is not accepted, the neighboring solution is ignored. The temperature parameter T is initially set to a high value so that most neighboring solutions are accepted. T is then reduced by a multiplicative factor slightly below one during each iteration, decreasing the probability that nonimproving neighboring solutions are accepted over time. The algorithm terminates after a large number of iterations, where the temperature is typically low enough so that the algorithm only considers improving neighbors. This classical search method is recognized to result in approximately globally optimal solutions.

In our case, the current solution consists of a set of routes over which all passengers are transported. Initially, this set of routes corresponds to a route with only a single ride for each route. Neighboring solutions are created by looking at one route (using the *Create* neighbor operation) or by considering two routes and performing one of the other neighborhood operations displayed in Table 2. These operations are classical neighborhood transitions applied to vehicle routing problems. The only operation that requires some explanation is *Tailswap*. This operation looks for two rides on different routes that are similar in the sense that they have pickup or drop-off locations that are close both location-wise and time-wise. Then, the operation selects from each route the set of rides that have pickup times after the chosen similar ride and exchanges the two sets of rides between the routes. The operations have varying success rates in yielding routes that adhere to all restrictions (i.e., result in feasible routes). The *Create* and *Move* operations are the most successful because the route from which the ride was extracted always remains feasible. In the first iterations, over 80% of the operations used are of the *Create* or *Move* type, and the remainder are of the *Swap* or *Tailswap* type. In the later iterations of the algorithm, this shifts toward an equal share of *Create/Move* and *Swap/Tailswap* operations.

Our approach views this relatively simple simulated annealing approach through a modern lens: its success relies critically on selecting the right neighborhoods and, most importantly, on various techniques to improve speed, including parallelization. We describe these considerations in the sections below, after which we

Table 2. Neighborhood Operations for Ride Combinations for the Simulated Annealing Heuristic

Operation	Description
Create	Extract a ride from a route to create a new route.
Move	Extract a ride from a route and move it to another existing route.
Swap	Exchange two rides between two routes.
Tailswap	Exchange two sets of multiple rides between routes.

conclude this section in the section titled Applying Fairness with a description of a fairness criteria to balance the work for subcontractors.

Simple Solution Space. We model each route as a subset of rides without storing the order of pickups and drop-offs. This yields a much simpler solution space for the simulated annealing approach than one in which the ordering is included in the modeling. However, it requires that we calculate the order each time we do a move in the simulated annealing algorithm. We experimented with both approaches and found that using subsets of routes requires far less computation time to obtain solutions of similar quality. This is because fewer simulated annealing iterations are required and computing the order of pickups and drop-offs can be done very efficiently in each iteration using a mini branch-and-prune algorithm for the few possible orderings based on pickup and drop-off windows. The number of possible orderings of pickups and drop-offs is significantly limited for three reasons. First, the vehicle capacity is limited, with the largest taxis only allowing space for up to six passengers. In conjunction with the restrictive rule of allowing only a 50% increase in transit time per passenger, this limits the number of riders who can simultaneously be in the car. Second, there are few stops on a route, and drivers can drive only a few hours without a long break. Third, pickup time windows are rather restrictive, with lengths of less than 30 minutes. Therefore, for any given set of rides within a route, most of the ordering of the rides is already determined. Note that these characteristics differ from those found in broader pickup-and-delivery problems, such as last-mile deliveries. These cases typically have wider time windows, more stops, or higher concurrent loads.

We search for the best ordering using an efficient mini branch-and-prune algorithm. This algorithm first sorts all pickups and drop-offs based on their time windows. Then, we build a search tree by considering exchanging pickups or drop-offs that are next to each other in the current order. The above rules allow for very efficient pruning of this search tree, often requiring no branching at all.

Neighborhood Selection: Bias and Feasibility. We precompute information about compatibility and similarity of pairs of pickup and drop-off locations. Compatibility implies that two rides might appear together in a feasible route, considering the time windows and driving times. This excludes about half the possible combinations of two rides because these would violate time windows or maximum shift duration. We use this compatibility information in the simulated annealing algorithm in which we select neighbors only from the compatible possibilities. Similarity is defined at either pickup, drop-off, or both. It is a combination of the difference in time window and travel duration between two stops. The neighborhood operations use this precomputed similarity information to create a bias in the neighborhood selection.

Note that these procedures for both bias and feasibility would be hard to compute without a full travel-time matrix, which we explain in the section titled Very Large-Scale Distance Matrix Computation.

Fast, Parallel Algorithm Implementation. We used a significant amount of development time to maximize the number of iterations per second that our algorithm executed to produce high-quality solutions in a short time. In addition to various optimizations done in low-level programming, important speed improvements resulted from the parallel exploration of multiple neighborhoods and caching.

Parallel execution of calculations has become a decisive factor in artificial intelligence algorithms, especially in deep learning. In classical OR, the use of parallelism is mostly done by separating the search space into multiple subspaces to be searched in parallel, sometimes with some small interaction between the parallel searches, for example, a branch-and-bound algorithm communicating a common bound. More recent approaches involve low-level parallelization; see Schryen (2020) for an extended overview. Our approach differs from most of these approaches because we apply parallelization *within* the same solution, which consists of a set of candidate routes. This approach is in contrast to, for example, the parallel simulated annealing for vehicle routing discussed in Wang et al. (2015), which includes multiple independent simulated annealing runs synchronizing their best solutions when ready and then reusing this best solution as the initial starting solution in the next series of parallel simulated annealing runs.

Observe that the neighborhood operations specified in Table 2 work only on two routes. Hence, in a sequential simulated annealing, most parts of the current solution are not changed while evaluating a new neighbor. We consider parallel moves by employing a locking mechanism based on parallel programming principles. That is, we lock the routes on which a neighborhood operation operates, preventing other neighborhood operations from using the routes simultaneously. At the same time, we enforce that the same sequence of neighborhood operations is executed as in the sequential (nonparallel) algorithm, and we synchronize the local effects on the objective function. This combination of locking and synchronizing while using the same neighborhood operation sequence guarantees that we obtain the same final solution as the sequential algorithm while gaining the runtime benefits of parallelism.

In this way, we can, in principle, scale the number of potential parallel neighborhood operations to nearly half the number of routes expected in an optimal solution. In practice, we run our algorithm on a 32-core machine on which we can fully exploit the parallelism. We had to take special care in implementing the necessary synchronization and investigating the impact of parallel neighborhood operations on route-overarching scores such as fairness.

A final improvement lies in smart caching of historic moves. Although the branch-and-prune evaluator for feasibility is fast, it might still take several microseconds, and because of the nature of a simulated annealing algorithm, it is likely to generate the same set of rides in a route several times during execution. This would unnecessarily repeat the evaluation, and caching many moves reduces the required computation time further, especially in the final stages of the algorithm.

Applying Fairness. There are no *hard* constraints on the number of vehicles that a subcontractor uses. In a region, many rides could be booked on one day and only a few the next day because bookings cannot be refused and there is no limit on the number of bookings on any given day. Nevertheless, during discussions, subcontractors indicated that the volatility in their day-to-day workload was undesirable (see also the section titled Adjustments in the Subcontractor Planning Processes). Therefore, we changed the objective from one that only minimizes cost (or travel time) to one that weighs minimal time and *fairness* in the amount of paid kilometers per subcontractor.

To give each subcontractor a fair share of the rides, we calculated the number of paid kilometers each would receive using the 2019 operation-area allocation. That is, per subcontractor, we calculated the paid kilometers of rides with pickups in that subcontractor's designated operating area over a given rolling horizon (e.g., three months). Dividing this number by the total paid kilometers in the Netherlands during that time gives a share $\sigma_i \in [0,1]$ per transporter i. If, on a specific day, we have P paid kilometers over the entire Netherlands, then subcontractor i is given a target paid kilometers of $\sigma_i P$. Whenever the solution allocates routes such that the subcontractor's paid kilometers deviate by more than 5% from the target, a (large) penalty is incurred in the objective. The result is that on almost all days, the subcontractors are within 5% of their targets, with negligible increases in overall driving time.

Very Large-Scale Distance Matrix Computation

The second ingredient of our OR solution is our state-of-the-art distance matrix computation engine, which can efficiently create a matrix with hundreds of millions of distance and travel-time entries. This was key to attempting to enable any OR solution in the first place.

The Need for State-of-the-Art Travel-Time and Distance Calculations. An often overlooked necessity for transportation algorithms is the need for accurate calculation of distances and travel times. Many examples in the academic literature use Euclidean distances (i.e., as-the-crow-flies distances). Especially when instance sizes become larger, theoretical experiments usually do not use a distance matrix but consider distances that can be directly computed from the location of the origin and destination nodes. Reinelt (1991) includes examples of the over 3,000 papers that use the classical traveling salesman problem (TSP) instance library (TSPLIB).

In practice, however, one does need real road-network-based distances and travel times, often based on roadmaps with millions of roads. Euclidean distances simply do not suffice for accurate planning. At the same time, straightforward shortest-path implementations of Dijkstra's algorithm to compute a distance matrix become intractable because of the large number of locations involved and the size of the road network. The following quote from Cook et al. (2018), in which one of the largest TSPs was solved on a roadmap (i.e., a TSP visiting all British pubs), illustrates the challenge:

> In working with road data, we were faced with the challenge of finding the correct TSP solution even though we could not possibly ask Google for all 1,232,159,688 pairs of pub-to-pub distances. In our earlier work on the 24,727-pubs tour, we used an ad hoc, trial-and-error process to gather enough Google pub-to-pub distances to permit the computation to go through […] For this new, much more difficult, problem, we developed algorithms to automate this portion of the computation, requesting pub-to-pub distances for 2,214,453 pairs, only 1/500th of the total number.

In the Valys problem, we faced additional challenges beyond those described above. First, Cook et al. (2018) did not include an operational time constraint. Their algorithm to find a solution began processing on January 11, 2018, and the distance-gathering component completed more than a month later on February 15, 2018. Three months later, on May 16, 2018, they delivered the optimal solution (although they found a good solution much more quickly). At Transvision, we have just one hour of total computation time each day during which we need to calculate the required distances and also solve the large-scale dial-a-ride problem.

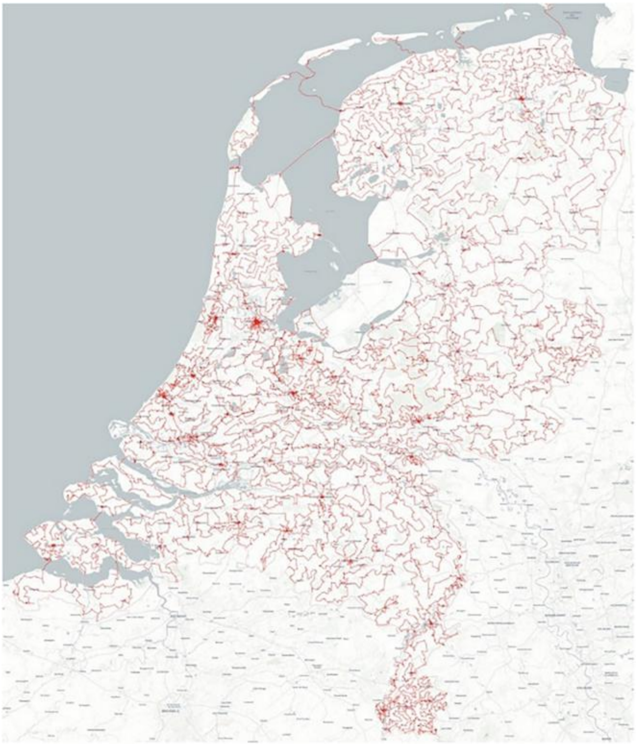
Second, the TSP problem described in the quote above uses walking distances. For these distances, as-the-crow-flies distances are a good approximation (or lower bound), which can be used to select the necessary pairs for full-distance computations. For our problem, we need travel times by car, for which as-the-crow-flies times do not reflect reality. In addition, our problem involves many more rides that can be deemed likely to end up in the same route, making many more point-to-point travel times of interest to the algorithm. Furthermore, we wish to precompute the rides that can never be together on the same route (see the section titled Neighborhood Selection: Bias and Feasibility), a procedure that relies on a full and accurate matrix with distances and travel times.

Nevertheless, let us for now assume that we would only need 5% of all distance pairs and we are optimizing a day with only 2,250 rides that have both a pickup and drop-off location, therefore 4,500 locations. Therefore, we need to compute $4{,}500 \times 4{,}500 \times 0.05 \approx 1.01$ million distances and travel times. The roadmap in the Netherlands consists of a few million nodes and edges. Even with proper preprocessing of the map, calculating a single point-to-point route takes at least five to 10 milliseconds. Any overhead to call an (external) service can quickly make this time much greater. Using a very conservative estimate of a total of 20 milliseconds per distance pair, the time needed for the 1.01 million pairs generated by only 2,250 rides is over five hours. This greatly exceeds our available computation time, making an efficient solution that considers all potential combinations useless.

Finally, from the perspectives of reliability and maintenance, a simple yet adaptable setup with well-defined interfaces offers significant benefits. This approach includes having a separately testable matrix calculation tailored to project needs, independence from real-time external services for point-to-point calculations (thus avoiding connectivity risks), and the advantage of not having to address these aspects in the algorithm's code.

The state-of-the art, extremely efficient distance-matrix calculation developed by CQM fills this need. With CQM's calculation, a matrix with $4{,}500 \times 4{,}500 = 20{,}250{,}000$ point-wise distance-travel time pairs took 25 seconds on an Intel Core i9-10855H 2.40-GHz Windows computer with 32 GB of RAM using four cores, that is, only 0.0012 milliseconds per pair of locations. To show the potential of further scaling, we also calculated a (symmetric) matrix with $57{,}912 \times 57{,}912/2 = 1{,}676{,}899{,}872$ entries of distances and travel times between Dutch National monuments, using cycling distances. The required computation time was less than two hours, with a significant amount of time also spent on writing the matrix of almost 100 GB to disk. These results were subsequently used to calculate the largest TSP ever solved on a roadmap (this record still stands as of January 2024). This was achieved by CQM in conjunction with the same team that solved the TSP quoted above; see Cook et al. (2021). However, in this case, the algorithm used a matrix with *all* distances, and no selection of required distance pairs nor the large calculation time previously associated with such a feat. Figure 4 visualizes the record-breaking TSP solution.

Figure 4. (Color online) Map Illustrating the Record-Breaking Optimal Traveling Salesman Tour Visiting All 57,912 Dutch Monuments



Note. Within two hours, the symmetric distance matrix with 1,676,870,916 point-to-point distances was calculated.

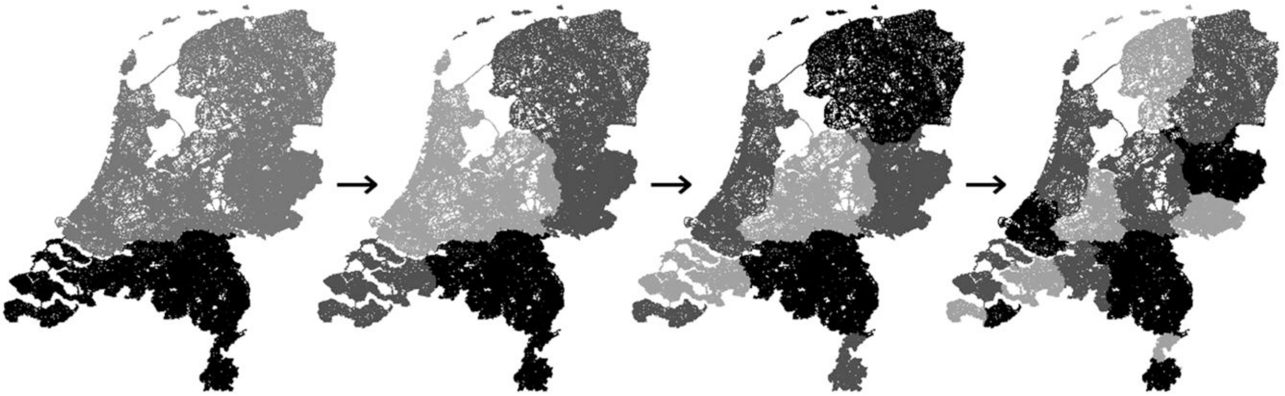
Map Preprocessing for Fast Calculations. Dijkstra's algorithms for point-to-point distances, as well as other shortest-path algorithms, are known as "efficient" because polynomial algorithms exist for them. However, as we discussed in the previous section, computation time quickly increases when considering graphs of millions of nodes and edges and also requires very large-distance matrices containing tens or hundreds of millions of point-to-point pairs.

We use two properties of road networks to enable significant improvements over the classical Dijkstra's algorithm by preprocessing:

- Road networks have small min-cuts (i.e., the minimum number of edges in a graph that need to be deleted to split the graph into multiple components) on all scales: typically, there are only a few roads between neighborhoods, between cities, crossing rivers, and between countries.
- Most routes tend to use the same few common roads (e.g., highways and other major thoroughfares).

Our preprocessing algorithm is similar to that of Delling et al. (2011). We split the road network's nodes into a hierarchical partition. On each level, we partition a connected component into two parts, where we try to balance the partition sizes and minimize the number of roads crossing between partitions. In this case, the first property of road networks applies: small cuts with a small number of roads on them typically exist. Figure 5 shows how the Netherlands can be

Figure 5. Hierarchical Partition for the Netherlands



Note. Assuming knowledge of Dutch geography, one can recognize the partition along the many rivers and natural boundaries in the Netherlands.

partitioned hierarchically. We use the inertial flow algorithm (Schild and Sommer 2015) to create our partitions; the result is a hierarchy 18 levels deep, with leaf partitions of at most 250 nodes.

A shortest path, which starts and ends outside a partition, must enter and exit that partition using the partition-crossing roads. Thus, we can simplify the graph corresponding to a partition by only retaining roads that lie on the shortest paths from partition-entering to partition-exiting roads. The second property of road networks also applies here: these shortest paths usually take only a few of the possible roads within a partition. This allows us to combine sequential individual roads into a single-shortcut road.

By precomputing these partitions and their simplified graphs, we can quickly calculate shortest paths even on gigantic road networks. Furthermore, it allows us to update the preprocessed map in near real time. Changes affect only one or few partitions, so only those partitions have to be updated. This particular feature is used heavily by Transvision to ensure that travel times are up to date.

A Fast Implementation of the Dijkstra Algorithm. To calculate shortest paths, we use the multilevel Dijkstra algorithm (Jung and Pramanik 2002). We compute the shortest paths from a source node; however, whenever such a path reaches a partition-exiting road, we switch to the corresponding parent partition in the hierarchy and its simplified graph. We do the same but traverse arcs in reverse from the target node. These shortest paths meet at some nodes, typically higher up in the hierarchy. One of these meeting paths must be the shortest path. Because the simplified graphs for partitions are limited in size, the runtime of the shortest-path computation roughly scales with a factor of log$N$, where N is the number of nodes in the road network.

We improve on independently calculating all pairwise shortest paths using the following observations. Notice that the forward and backward searches for a single path depend on each other only through the points where they meet. Also note that we only need to check for meeting paths on roads entering or exiting a partition and, as we just discussed, the structure of road networks, and the hierarchical partition ensures that there are relatively few such roads (typically some constant times log$N$).

Our implementation conducts an exhaustive backward search for each location in the distance matrix, storing the distance and travel time for each partition-entering or partition-exiting road. Then, for each location, we execute an exhaustive forward search. Finally, we check where any of the forward and backward searches meet to determine all location-to-location distances. Our implementation combines the forward search with the meeting check for further efficiency. For the matrix sizes in which we are interested, the majority of the computation takes place in the forward and backward searches. Thus, in practice, our algorithm's running time scales nearly linearly to the number of locations.

Reflections on the Implementation Journey

After CQM and Geodan won the competition organized by Transvision in 2018, which we described in the introduction, we made additional practical improvements in 2019.

We tuned the OR algorithm, and in addition, dozens of taxi registration software systems and onboard computers had to be updated and linked to the central planning system. Successfully interfacing with these aged software systems was an undertaking that required months of extensive alignments and testing. Several months were required for an intensive pilot phase with subcontractors. In this phase, the subcontractors were assigned one or two computed routes per day, whereas the rest of the rides remained regionally allocated. This provided valuable feedback about the routes that subcontractors preferred and gave substance to discussions between planners who were using the OR solution and taxi operators. In addition, we introduced a new pricing mechanism to streamline incentives for efficient planning and to provide a fairer distribution of the resulting financial gains (see the section titled A Novel Pricing Mechanism to Benefit Subcontractors below). Months were needed to achieve sufficient buy-in from all stakeholders. On January 1, 2020, the OR solution went live.

From 2020 onward, we introduced several upgrades to the OR solution to align with the preferences of drivers and taxi operators. These include algorithmic efficiency upgrades, parallelization of the heuristic, consideration of on-the-fly road closures, and incorporation of the fairness distribution for subcontractors as an optimization objective (see the section titled Adjustments in the Subcontractor Planning Processes below). Additionally, several ad hoc changes were applied to the algorithm to address sudden challenges such as the COVID-19 pandemic and the Dutch nitrogen crisis. Figure 6 shows a concise implementation timeline.

Agile Optimization: Dealing with Sudden Crises

Our OR solution was designed with agility in mind: both the simulated annealing algorithm and the specific way in which the distance matrices are computed are relatively easy to adapt in case of rapidly changing requirements because of external circumstances.

This agility was tested three months after we went live as the COVID-19 pandemic struck. Overnight, the government imposed new restrictions to counter infections among the Valys passengers, who belong to the most vulnerable groups in society. The next day, pooling was temporarily disallowed, and our plans had to incorporate extra cleaning time of the cars between picking up passengers. These changes were minor from an algorithmic point of view because we only needed to change the procedure that tests the feasibility of a taxi route. However, this adaptation went significantly smoother with the new OR solution. Using manual planning, subcontractors would have had to modify their plans manually, thus leading to potential planning errors, because they lacked experience with the challenges brought on by the COVID-19 pandemic; the likely result would have been reduced on-time performance.

A second test of this agility came when the government reduced the speed limit on highways to 100 kilometers per hour (BBC 2019) in response to the nitrogen crisis in the Netherlands (see also the section above titled Environmental Impact: Reducing CO_2 and Nitrogen Emissions). Previously, our approach relied on historical driving speeds on each road segment, which were updated every three months. The speed limit change required Geodan and CQM to develop new, realistic driving times for highways. To address this, we developed a method that provides conservative driving time estimates for the initial days without requiring historic driving speed data. This helped us to deal with these changes while maintaining high-quality solutions and high quality of service.

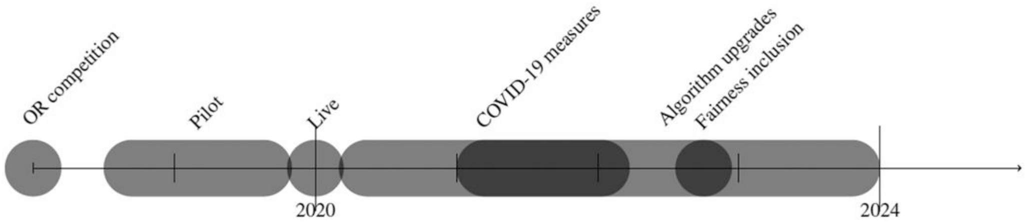
A Novel Pricing Mechanism To Benefit Subcontractors

One of the key factors to our success was aligning all stakeholders. Apart from extensive involvement of all parties and broad discussions on the functional requirements, we also had to devise a mutually beneficial pricing mechanism ensuring that all parties would benefit from the increased planning efficiency.

Understanding the financial conditions imposed by almost all taxi tenders in the Netherlands is helpful in understanding our novel pricing mechanism. For each booking, Transvision receives a fee from the government that is proportional to the shortest direct-travel distance between the pickup and drop-off locations of a passenger. This shortest direct-travel distance is referred to as the number of *paid kilometers*. These paid kilometers are independent of any planning choices (in particular, pooling) and are therefore fixed at the time of booking. The same pricing mechanism based on a fixed fee per paid kilometer had been the standard in contracts between taxi companies like Transvision and its subcontractors.

Keeping the old pricing mechanism toward subcontractors in place while using optimized routes would have had some negative consequences, especially in periods in which a relatively low number of rides were booked. To understand this, first observe that

Figure 6. Implementation Timeline from Transvision’s Operations Research Competition in 2018 Until 2024 with Savings of Approximately 100,000 Kilometers on a Typical Day



Transvision would have had to reduce the fixed fee per kilometer it paid to its subcontractors in order to realize any financial gains itself and to cover the risk and cost associated with developing the OR solution. Because the OR plans are much more efficient, this would often not be a problem for the subcontractors. That is, when the number of empty driving kilometers is reduced, the number of paid kilometers for the subcontractor can be much higher than the *actual* driving distance of the route. The same applies when, through systematic pooling, efficient routes are provided, which often involve passengers from different pickup and/or drop-off locations riding together in the vehicle. However, in periods with a relatively low number of ride requests, planning efficiently is harder. In those periods, the reduced fee may not cover the costs for the subcontractor. Efficiency decreases with fewer rides because they are more sparsely spread over time and geography, thus making pooling and reducing empty kilometers harder. Therefore, the old pricing mechanism would place the full risks of the unknown, possibly a volatile number of rides, on the subcontractor. The COVID-19 crisis, a time during which our new pricing mechanism had already been applied, proved that this was not a hypothetical risk.

To better distribute both the risk and the financial gains in these scenarios, we developed a new pricing mechanism. The goals of this pricing mechanism were (1) to financially cover Transvision's risks and operating expenses, and (2) to benefit subcontractors by ensuring consistently higher revenues per kilometer and hour spent for Valys. To do so, the new pricing mechanism is based on two key performance indicators (KPIs) of the plan: the *combination ratio* and *hourly paid kilometers*. We define these KPIs as follows:

$$\text{Combination ratio} = \frac{\text{Total paid kilometers}}{\text{Total driving kilometers;}}$$

$$\text{Hourly paid kilometers} = \frac{\text{Total paid kilometers}}{\text{Total driving time (hours)}}.$$

In both KPIs, only the denominator can be influenced by efficient planning because the paid kilometers are fixed beforehand. Indeed, if one plans more efficiently, then the total driven kilometers and the total time needed to execute all rides decrease. The new pricing mechanism for a ride works as follows:

1. If the route is singular (i.e., consists of a single ride), Transvision pays the full basic fee.

2. Otherwise, Transvision pays a fee that is negatively proportional to the combination ratio.

Under the new pricing mechanism, each route that involves multiple rides must exceed a minimum threshold for hourly paid kilometers. If the route's hourly paid kilometers are below this threshold, then the rides are presented to subcontractors as individual rides at the full basic fee.

The hourly paid kilometers is the most important KPI for subcontractors because subcontractor costs correlate highly to driver time. However, the total time required to execute a route is not fixed before executing the route: the actual time spent on boarding, driving, and driver breaks is variable and nontrivial to track precisely using the available data. Therefore, we chose the combination ratio KPI, for which the data are easier to collect and communicate, to determine the fee that Transvision pays to the subcontractors.

To calibrate the pricing mechanism, we initially compared the new optimized plans to the scenario in which all rides would be singular routes, assuming that the number of paid kilometers and empty kilometers in a singular route are equal because of the interregional nature of Valys. The latter would result in a combination ratio of 0.5 because empty driving kilometers then equal the paid kilometers. However, as the subcontractors rightfully pointed out, this grossly underestimates their own manual planning capabilities. Therefore, we calculated the KPIs for three alternatives: (1) with the new optimized plan, (2) when all rides are executed as singular routes, and (3) using an optimistic estimate in which each subcontractor manually plans its small share of rides in isolation. Table 1 shows the KPI values in each scenario for a typical month in 2023.

Manual planning by each of the 60 subcontractors, as done in 2019, means that small subcontractors would have to schedule approximately 20 rides per day, and larger subcontractors would schedule up to 500 rides. With a handful of rides (e.g., 20), the optimal schedule can easily be found manually, especially because most of these rides are hard to combine and are therefore left as single rides. With several dozens to hundreds of rides, as larger subcontractors schedule, the subcontractors have some options, but manually optimizing becomes more difficult. For the numbers in Table 1, we simulated an optimistic estimate of manual planning by creating a schedule using our OR solution for each isolated set of rides separately per subcontractor. This resulted in an average combination ratio of 0.59 (i.e., average of the combination ratio of the sets of routes assigned to each subcontractor). This is better than executing individual rides; however, it is not close to the average combination ratio of 0.89 that our OR solution found when applied to all rides nationwide. Note that on a typical day with 4,000 rides and 200,000 paid kilometers, the difference between a combination ratio of 0.89 and 0.59 is more than 100,000 driving kilometers. Still, in the section titled Financial Impact: Transvision and Other Stakeholders, we calculated overall financial savings based on a more conservative estimate of 50,000 kilometers saved on average per day.

Adjustments in the Subcontractor Planning Processes

In 2020, planning efforts were reduced as a result of the OR solution, which freed up time, thus enabling the subcontractors to plan other services. Were all taxi companies enthusiastic from the start? No, and we must admit that some subcontractors would rather go back to the old approach. A challenge with the OR solution is that all routes are presented one day in advance because the system waits until the vast majority of bookings have been made to initiate its processing. Before 2020, the subcontractors knew the requested rides at the time of booking. Therefore, a small minority of the rides would be known two days in advance and a few even sooner. This advance information gave subcontractors an indication of the required capacity, which helped them schedule vehicles and drivers. As a result, we included the fairness criteria in the objective function, giving them a more predictable number of rides (see the section titled Applying Fairness).

Another reason for hesitation among some subcontractors was that the OR plan removed the option to "negotiate" the pickup times with passengers, allowing subcontractors to slightly adjust pickup or drop-off times to better fit their schedules. In addition, subcontractors can no longer exchange rides among themselves because most rides are now part of multiple-ride routes. Although both methods of creating a plan in this way require an intensive manual process when done correctly (as a few transporters proved), cherry-picking the best rides to exchange in order to optimize routes is possible. Unfortunately, we have no data to measure the extent of the efficiency gain by doing so on an individual basis. It is certainly imaginable that a few experienced planners could cherry-pick with other subcontractors and myopically find better routes. However, the frequency of these myopic combination opportunities occurring is low to nonexistent, and complex route combinations (e.g., the route shown in Figure 2) are nearly impossible to construct manually.

Portability

Valys is by far the largest taxi contract in the Netherlands. In addition to Valys, many smaller contracts, such as regional rides and nonemergency medical transport, exist where the solution can be applied directly. Combining these contracts into one big plan has thus far not been possible because current regulations forbid such combinations in most cases. Inspired by the efficiency gains of the proven optimization solution, policymakers can steer toward more inclusive contracts that allow combining routes from different contracts.

The OR solution can also be rolled out to other countries that have taxi transport for elderly and disabled citizens with constraints similar to those of Valys. Most notably, in the United States, each city or state has to provide transportation means for the elderly and disabled, called paratransit, under the Americans with Disabilities Act (ADA). Each of these contracts are of similar or smaller size than the Dutch Valys tender. For example, the MTA New York City Transit holds the largest paratransit contract, accounting for five million rides and 73 million paid kilometers annually; see table 7 in American Public Transportation Association (2021). This scale is comparable to that of Valys, which facilitates one million rides per year, amounting to 50 million paid kilometers, largely because of the long-distance nature of its services. Notably, in New York City, several subcontractors also contribute by independently managing and planning a small portion of rides. This fragmentation is similar to Valys's situation prior to 2020 and is known to lead to less efficient planning because of the isolated handling of scheduling and services.

Finally, the technical advancements in the distance matrix calculation and simulated annealing have been effectively utilized in diverse applications, such as multiple-depot pickup and delivery of freight using a heterogeneous fleet of trucks. Specifically, the distance calculation is used to provide input for algorithms that manage the allocation of tens of thousands of tank containers across Europe's intermodal transport network. Additionally, in the warehouses of the Netherlands's largest food retailer, the advancements in simulated annealing described in this paper are used to reduce the required number of load carriers by 10%. This indicates a broad scope for the application of the advances associated with this OR solution.

Acknowledgments

The authors thank Meinolf Sellmann for helpful feedback during the semifinalist stage and the coaches Xinhui Zhang, Luciana Buriol, and Erick Wikum for helping them throughout the preparation, as well as a careful reading of an earlier version of this paper. The authors also appreciate the careful reading by Dick den Hertog and helpful guidance of Hein Fleuren. All OR and statistics experts at CQM helped the authors along the way by answering questions related to their expertise. The authors are also grateful to the entire project team and financial business developers at Transvision, most of whom have been at Transvision from the start of the project until this writing: Erwin Vergeer, Anne Rozeboom, Helma Schouwenaar, and Mireille Monshouwer. Finally, the authors would like to thank the Ministry of Health, Welfare and Sport, the entire planning and scheduling team at Transvision, the dozens of subcontractors and thousands of taxi drivers who transport and care for elderly and disabled citizens every day.

References

American Public Transportation Association (2021) 2021 Public transport fact book. Accessed August 20, 2024, https://www.apta.com/wp-content/uploads/APTA-2021-Fact-Book.pdf.

Baldacci R, Mingozzi A, Roberti R (2012) Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *Eur. J. Oper. Res.* 218(1):1–6.

BBC (2019) Netherlands forced to slash speed limit to reduce emissions. Accessed June 10, 2024, https://www.bbc.co.uk/news/world-europe-50396037.

Bertsimas D, Yan J (2020) The edge of optimization in large-scale vehicle routing for paratransit. Accessed June 10, 2024, https://www.dropbox.com/scl/fi/w8lo7m8oepz2wu93u38wj/Theedgeofoptimizationinlarge-scalevehicleroutingforparatransit.pdf?rlkey=kxgfhqvrnjfbb0av6eyjizv26&e=1&dl=0.

Cook W, Espinoza D, Goycoolea M, Helsgaun K (2018) UK49687 Shortest possible tour to nearly every pub in the United Kingdom. Accessed June 10, 2024, https://www.math.uwaterloo.ca/tsp/uk/index.html.

Cook W, Espinoza D, Goycoolea M, Helsgaun K, de Ruiter F, Leushuis C, Tijink M, Verberne B (2021) Cycling tour to 57,912 national monuments in the Netherlands. Accessed June 10, 2024, https://www.math.uwaterloo.ca/tsp/nl/index.html.

Cordeau J-F (2006) A branch-and-cut algorithm for the dial-a-ride problem. *Oper. Res.* 54(3):573–586.

Cordeau J-F, Laporte G (2003) A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Res. Part B: Methodological* 37(6):579–594.

Cordeau J-F, Laporte G (2007) The dial-a-ride problem: Models and algorithms. *Ann. Oper. Res.* 153:29–46.

Cummings K, Jacquillat A, Vaze V (2023) Activated Benders decomposition for day-ahead paratransit itinerary planning. Preprint, submitted December 20, http://dx.doi.org/10.2139/ssrn.4665007.

De Sociaal-Economische Raad (2023) Aanhoudende arbeidsmarktkrapte in publieke sectoren vraagt om ferme keuzes van kabinet (SER-advice February 2023 to the ministry of VWS). Accessed August 29, 2024, https://www.ser.nl/nl/adviezen/arbeidsmarktkrapte [In Dutch].

Delling D, Goldberg AV, Pajor T, Werneck RF (2011) Customizable route planning. Pardalos PM, Rebennack S, eds. *Proc. 10th Internat. Sympos. Experiment. Algorithms (SEA'11)*, Lecture Notes in Computer Science, vol. 6630 (Springer, Berlin, Heidelberg), 376–387.

Diana M, Dessouky MM (2004) A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. *Transportation Res. Part B: Methodological* 38(6):539–557.

Helder C (2023) Motie Agema en Dijk inzake Valys kilometerregistratie. Accessed June 10, 2024, https://open.overheid.nl/documenten/f24c28a4-0524-4e30-8664-e7b5bcf68ba1/file [In Dutch].

Jain S, Van Hentenryck P (2011) Large neighborhood search for dial-a-ride problems. Lee J, ed. *Principles Practice Constraint Programming – CP 2011* (Springer, Berlin, Heidelberg), 400–413.

Jung S, Pramanik S (2002) An efficient path computation model for hierarchically structured topographical road maps. *IEEE Trans. Knowledge Data Engrg.* 14(5):1029–1046.

Kuijpers N (2023) Dynamic reoptimization of transport for elderly and disabled. Unpublished master's thesis, Utrecht University, Utrecht, Netherlands.

Markovic N, Nair R, Schonfeld P, Miller-Hooks E, Mohebbi M (2015) Optimizing dial-a-ride services in Maryland: Benefits of computerized routing and scheduling. *Transportation Res. Part C Emerging Tech.* 55:156–165.

Molenbruch Y, Braekers K, Caris A (2017) Typology and literature review for dial-a-ride problems. *Ann. Oper. Res.* 259:295–325.

Parragh SN, Doerner KF, Hartl RF (2010) Variable neighborhood search for the dial-a-ride problem. *Comput. Oper. Res.* 37(6):1129–1138.

Reinelt G (1991) TSPLIB—A traveling salesman problem library. *ORSA J. Comput.* 3(4):376–384.

Remkes JW, Van Dijk JJ, Dijkgraaf E, Freriks A, Gerbrandy GJ, Maij WH, Nijhof AG, et al (2020) Niet alles kan overal: Eindadvies over structurele aanpak op lange termijn. Accessed August 29, 2024, https://edepot.wur.nl/523657 [In Dutch].

Schild A, Sommer C (2015) On balanced separators in road networks. Bampis E, ed. *Experiment. Algorithms. SEA 2015*, Lecture Notes in Computer Science, vol. 9125 (Springer, Cham, Switzerland), 286–297.

Schryen G (2020) Parallel computational optimization in operations research: A new integrative framework, literature review and research directions. *Eur. J. Oper. Res.* 287(1):1–18.

Schulz A, Pfeiffer C (2024) A branch-and-cut algorithm for the dial-a-ride problem with incompatible customer types. *Transportation Res. Part E Logist. Transportation Rev.* 181:103394.

Statistics Netherlands (2022) Forecast: Larger population due to migration. *Statist. Netherlands* (December 16), https://www.cbs.nl/en-gb/news/2022/50/forecast-larger-population-due-to-migration.

United Nations (2015) United Nations Decade of Healthy Ageing (2021–2030): Resolution adopted by the General Assembly on September 25, 2015. Accessed August 30, 2024, https://documents.un.org/doc/undoc/gen/n15/291/89/pdf/n1529189.pdf.

United Nations (2020) United Nations decade of healthy ageing (2021–2030): Resolution Adopted by the General Assembly on December 14, 2020. Accessed August 30, 2024, https://documents.un.org/doc/undoc/gen/n20/363/87/pdf/n2036387.pdf.

van Gijlswijk R, Ligterink N, Bhorasar A, Smokers R (2022) Real-world fuel consumption and electricity consumption of passenger cars and light commercial vehicles-2021. Accessed June 10, 2024, https://publications.tno.nl/publication/34639300/erZOUs/TNO-2022R10409.pdf.

Venneman I, de Gruijter W (2022) Chaos in leerlingenvervoer speciaal onderwijs treft duizenden kinderen die juist structuur nodig hebben. De Volkskrant (October 20), https://www.volkskrant.nl/nieuws-achtergrond/chaos-in-leerlingenvervoer-speciaalonderwijs-treft-duizenden-kinderen-die-juist-structuur-nodighebbeñb9e16dc0/ [In Dutch].

Wang C, Mu D, Zhao F, Sutherland JW (2015) A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows. *Comput. Indust. Engrg.* 83:111–122.

Frans J. C. T. de Ruiter is a senior consultant at CQM and a senior researcher at Wageningen University. His consultancy work covers decision making in many industries, including transportation, energy, retail, and semiconductors. His research interests are optimization under uncertainty and transportation, supply chains, and their application. He was awarded the INFORMS Optimization Society Best Student Paper Prize during his PhD. He holds a PhD in operations research from Tilburg University and was a visiting researcher at the Massachusetts Institute of Technology (Cambridge, Massachusetts) and The Technion in Haifa.

Johan M. M. van Rooij is director of data science at Valcon and assistant professor at Utrecht University. He worked at CQM when involved in this project. His industry work involves operations research and data science applied in manufacturing, supply chains, infrastructure, and transportation. His research is on parameterized algorithms for which he received the EATCS-IPEC Nerode prize for advances in that field. He holds a PhD in computing science from Utrecht University.

Peter Hulsen is a partner at CQM, where he applies his deep belief in operations research to create innovative solutions across various sectors. His expertise in machine learning, algorithm optimization, and software development has significantly impacted high-tech manufacturing, infrastructure, transportation, and warehousing. He holds an MSc in computer science from Eindhoven University of Technology.

Bart Post is a consultant at CQM, where he is an operations research expert, consultant, as well as researcher and developer of the distance matrix calculation. He holds a PhD in mathematics from Eindhoven University of Technology.

Jeroen Goes is a senior consultant at CQM, where he leads many analytics development and implementation projects. For many years, he was the project lead for the Transvision project. Prior to joining CQM, he worked at several high-tech companies and developed routing software. He holds an MSc in mathematics from Eindhoven University of Technology.

Geert Teeuwen is a principal consultant at CQM, where he is an algorithm expert. From 2000, he developed the initial point-to-point distance and travel calculations and was an architect of the new distance matrix calculation. He holds an MSc in computer science from Eindhoven University of Technology.

Matthijs Tijink is a consultant at CQM. Most of his work involves applying statistics and machine learning within challenging problems at R&D companies, with an occasional project relating to operations research. He was one of the main software developers of the distance matrix calculation. He holds an MSc in mathematics from the University of Twente.

Bart Verberne is a consultant at CQM, specializing in logistics optimization and high-performance software implementations. His primary role in the project centers around significant contributions to the simulated annealing algorithm. He holds an MSc in computer science from the Eindhoven University of Technology.

Niels Bourgonjen is a business manager at Geodan (now part of Sogelink) and was involved with the project from the start. He is an expert on location intelligence, and he holds an MSc in geology from Utrecht University.

Roelf Nienhuis is product owner at Transvision and has founded several software companies that he is leading, including Stranded Flight Solutions and several e-commerce companies. He holds a BSc from Breda University of Applied Sciences.

Tjeerd van der Poel is the chief financial officer (CFO) at Transvision. Prior to joining Transvision, he was Director of Business Control at KPN and manager at energy company E.ON, as well as a management consultant at Deloitte. He holds an MSc in economics from Erasmus University Rotterdam.

Laurens van Remortele is the chief executive officer (CEO) at Transvision, as well as CEO of the taxi app Dextr. He is a member of the board of the Royal Dutch Taxi Transport Association KNV. Prior to joining Transvision, he was a board member at ZCN, a company focusing on nonemergency medical transport.