Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 95 (2026) 353–360

Transportation Research Procedia

www.elsevier.com/locate/procedia

Euro Working Group on Transportation Annual Meeting 2025 - EWGT2025

Multi-objective optimisation of parking capacities in urban areas

Tygo Nijsten[a,b,*], Jan-Pieter Dorsman[a], Michel Mandjes[c,a], Erwin Walraven[b], Maaike Snelder[d,b]

[a]*Korteweg-de Vries Institute for Mathematics, University of Amsterdam, Science Park 105-107, 1098 XG, Amsterdam, The Netherlands*
[b]*Sustainable Urban Mobility and Safety, TNO, Anna van Buerenplein 1, 2595 DA, The Hague, The Netherlands*
[c]*Mathematical Institute, Leiden University, Einsteinweg 55, 2333 CC Leiden, The Netherlands*
[d]*Department of Transport & Planning, Delft University of Technology, Stevinweg 1, 2628 CN Delft, The Netherlands*

Abstract

Cars remain the most widely used mode of transport today. However, in many urban areas, high car usage leads to negative externalities such as congestion, pollution, and inefficient land use. Optimising parking policies in cities is a promising approach to reduce these externalities, though it often involves trade-offs; for example, reducing parking space can increase the time drivers spend searching for a spot.

We present a model to optimise parking capacities in urban areas using a multi-objective framework that simultaneously minimises (1) travel time, (2) distance travelled by car, and (3) the number of parking spaces. We address this problem using a bi-level programming framework as parking capacity decisions (upper level) influence driver route and parking choices (lower level), which in turn affect the objective values. Our main methodological contribution lies in enhancing the upper level optimisation through a novel mutation operator, which helps achieve lower objective values.

We apply our model to the city of Delft, the Netherlands, demonstrating that a diverse set of solutions with low objective values can be obtained. Moreover, we show through an example within this case study that our model can help policy-makers assess trade-offs in the conflicting objectives.

Peer-review under responsibility of the scientific committee of the Euro Working Group on Transportation Annual Meeting 2025 - EWGT2025.

Keywords: Optimisation; Parking; Network design; Metaheuristics; NSGA-II

1. Introduction

Currently, the car is the most used mode of transport in Europe (Eurostat, 2024). Nonetheless, high car usage in urban areas leads to negative externalities, such as congestion and emissions. When travelling by car, up to almost half of the travel time to urban areas is used to search for parking (Bonsall and Palmer, 2004). In addition, parking facilities

* Corresponding author.
E-mail address: t.g.c.nijsten@uva.nl

Peer-review under responsibility of the scientific committee of the Euro Working Group on Transportation Annual Meeting 2025 - EWGT2025.
10.1016/j.trpro.2026.02.045

generally take up space that could be used for other purposes, such as green spaces or housing. To help reduce these negative effects, parking policies can be applied.

Recently, there has been an increasing interest in optimising such parking policies with different objectives. The main research on the optimisation of parking policies can be divided into two categories: 1) the optimisation of parking locations and 2) the optimisation of parking fees. Examples of the first type include the optimisation of parking locations to minimise walking distance from parking facilities to destinations (Chen et al., 2001) and to minimise emissions (Shen et al., 2019). Examples of the second type include the optimisation of parking fees to increase the use of alternative modes of transport (D'Acierno et al., 2006) and to maximise parking occupancy (Pierce et al., 2015). Although parking capacities can implicitly be considered when optimising parking locations by looking at the number of facilities, research on optimising the number of parking spaces remains limited.

The existing literature on optimising parking policies has considered different objectives in a single-objective setting. However, trade-offs between different externalities can occur. For example, reducing the number of parking spots may increase park search times and thus total travel time. To address these trade-offs, we simultaneously minimise three objectives: (1) total travel time, which includes both driving and parking search time; (2) total distance travelled by car, which reflects overall car use across the network; and (3) the total number of parking spaces, which influences space usage, but also park search time and driver routing. While travel time and distance are related, the latter does not capture congestion and park search time. We aim to find an entire set of solutions that cannot be improved in any objective without worsening another objective, known as the *Pareto Front*. While studies have explored multi-objective optimisation of parking lot design (e.g., Moradijoz et al. (2013)), there is little to no research on multi-objective optimisation of parking policies. Thus, current models to optimise parking policies do not address the trade-offs between conflicting objectives.

As the solution space for multi-objective optimisation problems is often too large, the exact Pareto front cannot be identified. Metaheuristics like evolutionary algorithms are commonly used to approximate it (Emmerich and Deutz, 2018). An additional local search is sometimes used to identify better solutions regarding the objective values. A drawback is that it often increases computation time, especially in large, real-world problems.

The key contribution of this paper is twofold. First, we formulate a model to optimise parking capacities with the three mentioned objectives in a multi-objective setting. Unlike existing, single-objective approaches to optimise parking policies, our method allows for an analysis of trade-offs between the conflicting goals. To solve our problem, we propose an improved version of the state-of-the-art NSGA-II framework (Deb et al., 2002). Second, as our main methodological contribution, we introduce an additional local mutation operator that helps find better solutions without significantly increasing computation time. Applied to the city of Delft, our method yields a diverse set of high-quality solutions regarding the objective values, allowing policy-makers to study the trade-offs. To illustrate this, we provide an example where we assess the required increase in travel time to decrease the distance travelled by car while maintaining the same number of parking spaces.

The paper is organised as follows. Section 2 introduces the optimisation problem as a bi-level program and describes the method we use to solve this problem. The case study of Delft, as well as the results and possible implications for decision-makers, are discussed in Section 3.

2. Methodology

The model we propose in this paper aims to find the optimal parking capacities in each zone to minimise 1) the travel time, 2) the distance covered by cars and 3) the number of parking spaces. We formulate the model as a bi-level program, which is visualised in Figure 1. In the upper level, the aim is to find optimal parking capacities regarding the three objective values. The objective values can be found by solving a traffic assignment model in the lower level. This process is iteratively executed to identify the optimal capacities.

We proceed by specifying the methodology used in this paper, which is organised as follows. First, we define the road network with parking in Section 2.1. Next, we describe the traffic assignment model to determine objective values given parking capacities in Section 2.2. The optimisation problem to find optimal parking capacities is detailed in Section 2.3. Finally, Section 2.4 covers our application of the NSGA-II framework and the proposed improvement through the additional local mutation operator.

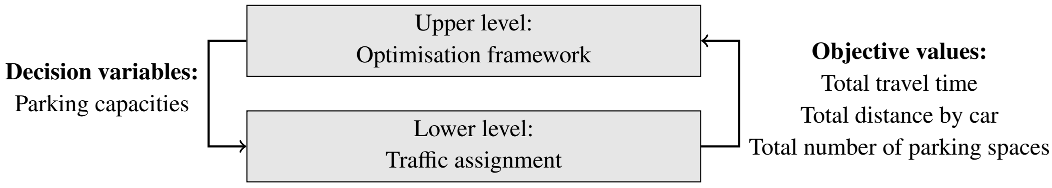


Fig. 1: Bi-level programming problem

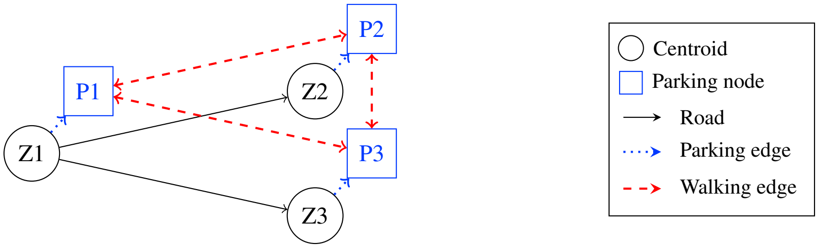


Fig. 2: Example of a road network extended with parking and walking

2.1. Network definition

We define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ representing the road network, with nodes $\mathcal{V}$ representing intersections and edges $\mathcal{E}$ representing roads. Each edge $e \in \mathcal{E}$ is equipped with a length l_e. Furthermore, we denote the flow on each edge $e \in \mathcal{E}$ by x_e and the travel time for this edge by the travel time function $D_e(x_e)$. In line with existing literature, we use a BPR-type function to model the travel time (Ryan, 1979). Explicitly, the travel time function is given by

$$D_e(x_e) = t_e \cdot \left(1 + \alpha_e \left(\frac{x_e}{c_e}\right)^{\beta_e}\right), \tag{1}$$

where t_e is the free flow travel time, c_e is the capacity and $\alpha_e > 0$ and $\beta_e > 0$ are edge specific constants. Ryan (1979) suggests taking $\alpha_e = 0.15$ and $\beta_e = 4$ for American highways, but the first parameter is usually larger for urban areas.

To deal with the computational needs of realistic network sizes, we divide the network into different zones $\mathcal{Z}$ and aggregate trips at the zonal level. Each zone $z \in \mathcal{Z}$ is equipped with a centroid functioning as an origin and destination for trips from and to that zone. Zonal centroids can be a subset of the nodes $\mathcal{V}$, but this is not required, as virtual edges without travel time can be introduced between the centroids and nodes.

To model parking, the road network is extended. Figure 2 shows an example of an extended network consisting of three nodes, also functioning as zonal centroids. Parking is included by adding parking nodes $\mathcal{V}_p$ to represent parking facilities and parking edges $\mathcal{E}_p$ to describe park search time. In our case, parking nodes and edges correspond to parking in a zone $z \in \mathcal{Z}$, i.e. a zonal centroid is connected to some parking node $v_p \in \mathcal{V}_p$ through some parking edge $e_p \in \mathcal{E}_p$. Similar to roads, a BPR-type function can be used to model park search time (Lam et al., 2006). It should be noted that road capacity refers to the number of vehicles where congestion occurs, while parking edge capacity indicates parking availability. Typically, the values for α_e and β_e for parking edges are chosen higher than for roads.

Furthermore, we allow drivers to park at a different location near their destination and walk the last part of their trip. So, for example, in the network of Figure 2, drivers can drive from zone 1 to zone 2, park their car in zone 2 and walk to zone 3. If, for example, zone 2 has high parking capacity and zone 3 has low capacity, this option is attractive. To allow for this, walking edges $\mathcal{E}_w$ can be added between parking nodes if the distance between centroids is less than a specified maximum d_w, depending on the network. As there is usually no congestion while walking, the travel time function on these edges is constant and only depends on the distance between the corresponding centroids.

2.2. Traffic assignment

In our traffic assignment model, drivers minimise their travel times, leading to the *Wardrop equilibrium*, where no driver can reduce travel time by switching routes. This equilibrium is achieved under two mild assumptions: the travel time function is 1) increasing and 2) continuously differentiable. With the chosen BPR-type functions for roads and parking edges and constant functions for walking, these assumptions are satisfied. To approximate the Wardrop equilibrium, we apply the Frank-Wolfe algorithm Frank and Wolfe (1956), ensuring reasonable computation time for solving the lower-level problem in our bi-level framework. Consequently, our model is scalable and applicable to large, real-life networks (Sheffi, 1985).

2.3. Optimisation problem

In the proposed problem, we aim to find parking capacities for each zone to minimise the three objectives defined before. We define a set of finite possible capacities Q_z for each zone $z \in \mathcal{Z}$. A solution can now be represented as a vector $\vec{q} = (q_{z_1}, \ldots, q_{z_n}) \in Q_{z_1} \times \ldots \times Q_{z_n}$ describing the capacities for zones $\mathcal{Z} = \{z_1, \ldots, z_n\}$. The objective values for each solution $\vec{q}$ can be computed using the traffic assignment model. Explicitly, $\vec{q}$ represents the capacities of the parking edges, i.e. q_{z_1} is the capacity for the BPR function corresponding to the parking edge in zone z_1. The model then outputs the flows in the Wardrop equilibrium for all edges, which we denote by $\vec{x} = (x_e)_{e \in \mathcal{E} \cup \mathcal{E}_p \cup \mathcal{E}_w}$. Given these flows, we compute the corresponding objective values. The first objective, the total travel time $t_{\vec{q}}(\vec{x})$, is the sum of the travel times of all travellers. This is equivalent to the sum of the product of the travel times and flows on each edge:

$$t_{\vec{q}}(\vec{x}) = \sum_{e \in \mathcal{E} \cup \mathcal{E}_p \cup \mathcal{E}_w} D_e(x_e) \cdot x_e. \tag{2}$$

Note that we include park search and walking time in the total travel time. Similarly, the second objective, the total distance covered by cars $d_{\vec{q}}(\vec{x})$, can be computed by summing the product of the lengths of each road and the flows:

$$d_{\vec{q}}(\vec{x}) = \sum_{e \in \mathcal{E}} l_e \cdot x_e. \tag{3}$$

The third objective, the total number of parking spaces $s_{\vec{q}}(\vec{x})$, is the sum of the parking capacities of each zone:

$$s_{\vec{q}}(\vec{x}) = \sum_{z \in \mathcal{Z}} q_z. \tag{4}$$

In summary, the formal problem can be written as

$$\min_{\vec{q}} \left(t_{\vec{q}}(\vec{x}), d_{\vec{q}}(\vec{x}), s_{\vec{q}}(\vec{x}) \right) \tag{5}$$

$$\text{s.t.} \quad \vec{q} \in Q_{z_1} \times \ldots \times Q_{z_n} \tag{6}$$

$$\vec{x} \in \text{WE}. \tag{7}$$

Equation (5) describes the minimisation of the three objectives. As discussed, due to the trade-off between the objectives, there is not one single optimal solution. Instead, we aim to approximate the Pareto front in which no solution can be improved in one of the objectives without worsening another objective. Equation (6) specifies the solution space for the parking capacities. Equation (7) indicates that edge flows correspond to the Wardrop equilibrium, which we derive using the traffic assignment model.

2.4. Solution algorithm

We apply an improved version of the state-of-the-art NSGA-II framework (Deb et al., 2002) to solve our multi-objective problem. Below, we describe the steps of this proposed improved algorithm.

1. *Initialisation:* Choose a population size N_p and a number of iterations N_i. Generate an initial population of size N_p by uniformly randomly sampling from the solution space and set $i = 0$.
2. *Evaluation:* Obtain objective values for each solution. In our case, this is done using the traffic assignment model. Rank solutions based on non-dominance to ensure low objective values (for minimisation). A solution dominates another if it has a lower value for at least one objective and no higher value for the remaining. Rank solutions with the same non-dominance ranking using the crowding distance, which measures how distant a solution is compared to its neighbours by examining differences in objective values, to ensure diversity of solutions.
3. *Parent selection:* Select N_p parents from the population using rank-based selection, where the top-ranked solution has weight N_p, the second $N_p - 1$, and so on. The same solutions may be selected multiple times.
4. *Crossover:* For each consecutive pair of parents (e.g., parent 1 & 2, 3 & 4, etc.), apply crossover with some probability p_c. To perform a crossover, randomly select a cut-off point and join the parts before and after it from each parent. When crossover is not applied, child solutions are kept identical to their parents.
5. *Mutation:* Apply a mutation to each solution of the offspring with a small probability p_m. The mutation entails uniformly randomly modifying a capacity to a different feasible capacity.

5b. *Local mutation:* Apply a local mutation with a small probability p'_m if standard mutation is not executed. This involves adjusting a capacity to a slightly higher or lower value, rather than considering all possible capacities. When neither mutation nor local mutation is applied, the child solution remains unchanged.

6. *Elitism:* If $i < N_i$, select the N_p highest-ranked solutions from the $2 \cdot N_p$ parents and offspring as the next population. Set $i = i + 1$ and return to step 2; otherwise, terminate.

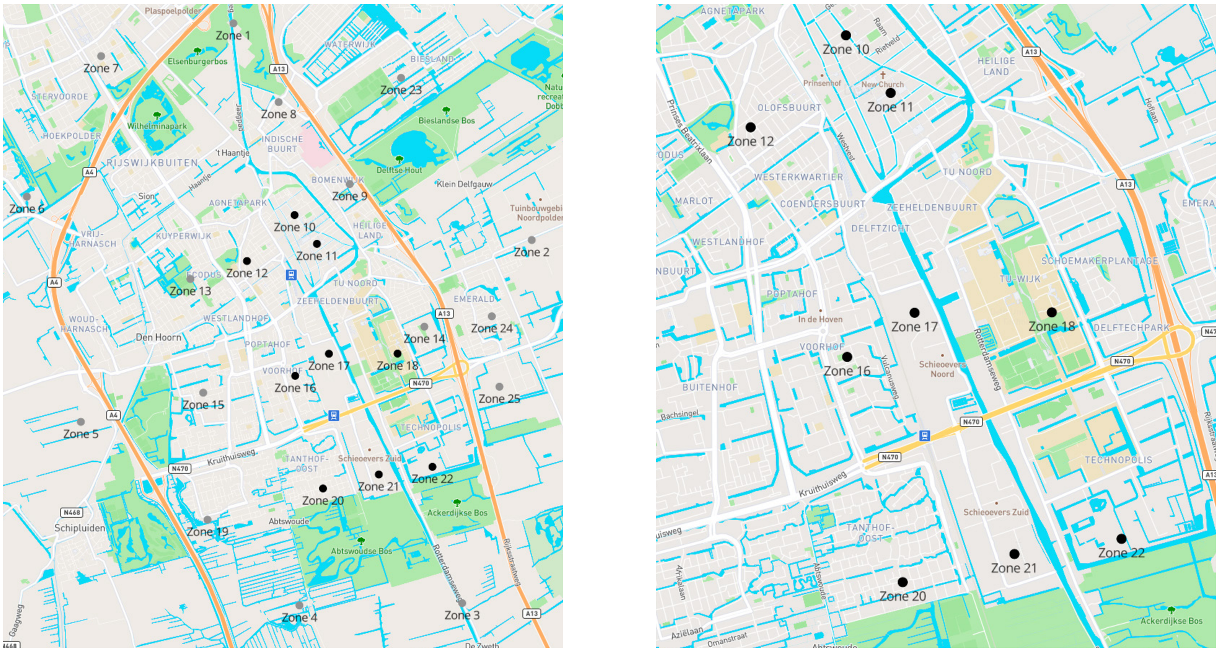
Our algorithm extends NSGA-II by introducing a local mutation operator (step 5b), which increases the likelihood of exploring solutions similar to those already in the population. This is especially useful in later stages, where nearby solutions often yield similarly low objective values. Without this operator, the algorithm would explore these solutions much less frequently, as the standard mutation operator considers the entire solution space. We retain standard mutation to ensure global exploration and avoid premature convergence. Unlike local search, the local mutation operator does not increase the number of traffic assignment evaluations, thereby preventing substantial computation time increases, which is crucial for large, real-life traffic networks.

3. Case study

The solution method is applied to optimise parking capacities in a case study of Delft, a city in the Netherlands, to evaluate the model and demonstrate that the trade-offs between objectives can be analysed. The network includes 491 nodes and 5808 edges, divided into 25 zones. The centroids of each of the 25 zones are shown in Figure 3a. Of these, nine zones, located near the city centre and Delft University of Technology, were selected for parking optimisation due to high congestion. Their centroids are shown in black in Figure 3a and separately in Figure 3b. Traffic between all 25 zones is considered, resulting in 625 OD pairs.

3.1. Parameters

We proceed by describing the parameters regarding the Delft network. The parameters for the BPR-type functions (Equation (1)) for the roads were obtained through OmniTRANS (Goudappel, 2025). The free flow travel time t_e is derived using the lengths and speed limits of each road, while $\alpha_e = 0.87$ and $\beta_e = 4$ have been chosen for all roads. The BPR function parameters for parking edges are based on Lam et al. (2006) and De Romph et al. (2017), resulting in $t_e = 0.06$, $\alpha_e = 300$ and $\beta_e = 4.1$ for all parking edges. Moreover, we have chosen the maximum walking distance d_w as 1.5 km, which is relatively high due to the large zones and distance between centroids. It is unlikely that drivers are willing to walk more than 1.5km from a parking facility to a destination (Van der Waerden et al., 2015). The



(a) Centroids of all zones in Delft

(b) Centroids of selected zones in Delft

Fig. 3: Visualisation of the centroids of the (selected) zones in Delft

possible parking capacities are based on the demands in the selected zones in Delft, resulting in the set of possible capacities $\{30 \cdot n \mid 1 \leq n \leq 100\} = \{30, 60, \ldots, 3000\}$ for each zone. This discretisation keeps computation time reasonable, resulting in a space of size 100^9. To approximate the current situation, we define a base case in which the parking capacities in each zone are set to 1.5 times the demand. Based on driver behaviour observed in the Delft model, this level of capacity is sufficient to accommodate all drivers without incentivising them to travel to a different facility farther from their destination and complete their trip on foot to minimise travel time.

For the evolutionary algorithm, we have chosen the population size $N_p = 200$ and the number of iterations $N_i = 50$, resulting in a computation time of just over one day for a single run. Running the experiments, we concluded that larger values come at a high computational cost, while lower values result in either local or insufficient convergence. In line with the literature, we have chosen a high crossover probability, i.e. $p_c = 0.7$ and a small mutation probability, i.e. $p_m = 0.05$. Regarding the local mutation operator, it modifies a capacity to a direct neighbour, i.e. a capacity of 600 is adjusted to 570 or 630 with equal probability. Recall that the local mutation operator is only applied when standard mutation is not executed with a small probability p'_m, which we set equal to 0.025. Setting this probability too large, i.e. 0.05 or higher, causes too many mutations to occur, which decreases the quality of the solutions.

3.2. Results

To evaluate the impact of the proposed additional local mutation operator, we ran NSGA-II with and without it across five independent runs, using the same initial population for both variants in each run. Table 1 reports the number of non-dominated solutions per variant and after combining both fronts. ‘Combine’ refers to adding the two resulting fronts together and computing the non-dominated solutions of this combined front. For example, in run 1, the standard and local mutation variants yielded 327 and 347 solutions, respectively. After adding the 674 solutions of both fronts together, 577 remained non-dominated: 267 from the standard variant and 310 from the local mutation variant. The results show that the local mutation operator yields a larger Pareto front in all but one run and results in a higher number of remaining solutions in the combined fronts in all runs, suggesting that it can help provide solutions with lower objective values and improve the original NSGA-II framework without increasing computation time.

Table 1: Comparison of the NSGA-II and our proposed algorithm with additional local mutation operator

Run	Method	Number of solutions in Pareto Front	Number of solutions in *combined* Pareto Front
Run 1	Standard	327	267
	Local mutation	**347**	**310**
Run 2	Standard	334	288
	Local mutation	**344**	**291**
Run 3	Standard	307	221
	Local mutation	**334**	**311**
Run 4	Standard	**333**	279
	Local mutation	328	**303**
Run 5	Standard	326	254
	Local mutation	**329**	**275**

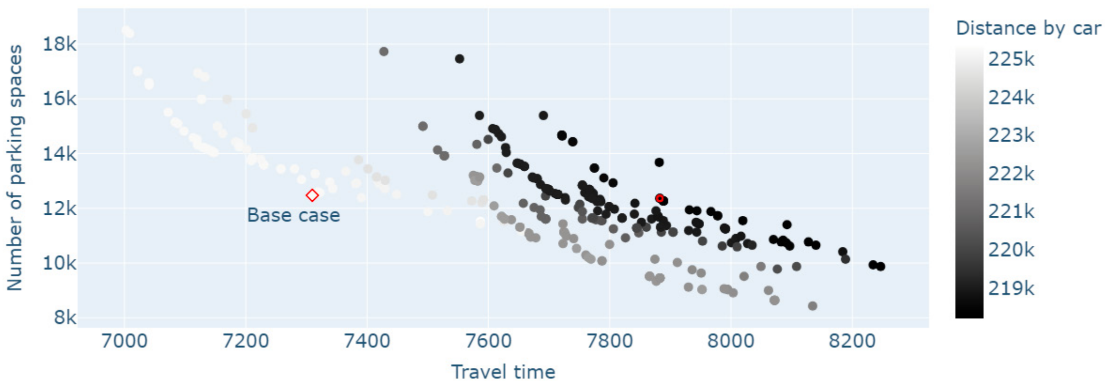


Fig. 4: Objective values of the resulting set of solutions (run 1, with local mutation operator)

As an example, Figure 4 shows the 347 non-dominated solutions from run 1. The x-axis shows travel time, the y-axis represents the number of parking spaces and the colour indicates the distance by car. The base case is marked as a diamond with a red border. The other solution with a red border will be discussed later. From this figure, we conclude that the algorithm provides diverse solutions for all three objectives, some similar to the base case and others significantly improving one or two objectives. It should be noted that the base case itself is infeasible because the parking capacities in the base case are not multiples of thirty, but similar solutions are still feasible.

Three different 'clouds' of solutions appear. A white cloud close to the base case appears. To its right, a grey cloud shows solutions with a lower car distance and fewer parking spaces but higher travel time. Further right, a black cloud with the lowest car distance but higher travel time and more parking spaces can be identified. This makes sense, as less distance by car means more walking, increasing travel time and thus shifting the black cloud to the right.

Interestingly, to obtain lower values for the distance by car, high capacities in Zone 22 and low capacities in Zone 21 seem crucial. This can be explained by the fact that Zone 22 is close to the highway, allowing fast access to this zone without covering much distance. The black cloud in Figure 4 consists of 210 solutions with this property. This also explains why solutions in the black cloud generally have higher capacities than solutions in the grey cloud: capacities in Zone 22 must be high to make it attractive to park there. By analysing this set of solutions in the black cloud, a policy-maker can appropriately consider the trade-offs when aiming for, in this case, a low distance covered by cars. To illustrate this, we consider the solution in the black cloud with the lowest distance covered by the car, which is marked by a red border in Figure 4. For this solution, the number of parking spaces is similar to the base case, but the total distance covered by the car is reduced by roughly 3.1% while the travel time is increased by roughly 7.8%.

4. Conclusion and future work

We have provided a method to optimise parking capacities in urban areas to minimise the total travel time, the distance covered by car and the number of parking spaces simultaneously. Contrary to the existing studies on the optimisation of parking policies, we have considered multi-objective optimisation. The advantage of this approach is that there is no need to define (subjective) weights, allowing a policy-maker to study the trade-offs between objectives. To find solutions with better objective values, we introduced an additional local mutation operator that does not require extra computation time, unlike classical local search procedures.

We applied our model in a case study of Delft, illustrating that we can find a diverse set of high-quality solutions in terms of objective values. Furthermore, we showed that the additional operator indeed helps identify better solutions. On top of this, we demonstrated that the trade-offs between objectives can be analysed and patterns can be recognised to help decision-makers. For example, to decrease the distance covered by cars, parking capacities in Zone 22 (near the highway) should be chosen high, but should be chosen low in Zone 21 (close to Zone 22, but more into the city).

The local mutation operator proposed in this paper is likely applicable to other multi-objective problems solved using the NSGA-II algorithm, making it an interesting subject for further study. Another future research path could be to use realistic parking supply data to define the base case. Additionally, different or more objectives could be considered in our problem, such as emissions. Moreover, it would be interesting to incorporate different parking types and study their effects on the objectives. On the other hand, different decision variables like parking fees can be added to the model. Lastly, drivers forced to walk might divert to other modes of transport. Therefore, it could be interesting to include other modes in the model, for example, through mobility hubs.

Acknowledgements

This research is partially funded by the Dutch Research Council (NWO) through the Gravitation project NETWORKS under grant number 024.002.003 and the Perspectief project XCARCITY with file number P21-08.

References

Bonsall, P., Palmer, I., 2004. Modelling drivers' car parking behaviour using data from a travel choice simulator. Transportation Research Part C: Emerging Technologies 12, 321–347.

Chen, J., Liu, D., Chen, X., Wang, W., 2001. Planning method of urban parking facilities' locating model with its genetic algorithm. China Journal of Highway and Transport 14, 85–88.

D'Acierno, L., Gallo, M., Montella, B., 2006. Optimisation models for the urban parking pricing problem. Transport Policy 13, 34–48.

De Romph, E., Snelder, M., Soekralla, A., 2017. Parkeren in het Verkeersmodel Amsterdam, Voorstudie. Technical Report. TNO.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197.

Emmerich, M.T.M., Deutz, A., 2018. A tutorial on multiobjective optimization: fundamentals and evolutionary methods. Natural Computing 17, 585–609.

Eurostat, 2024. Key figures on European transport – 2023 edition. Publications Office of the European Union, Luxembourg.

Frank, M., Wolfe, P., 1956. An algorithm for quadratic programming. Naval Research Logistics Quarterly 3, 95–110.

Goudappel, 2025. Traffic Modelling software OmniTRANS Expert.

Lam, W., Li, Z., Huang, H., Wong, S., 2006. Modeling time-dependent travel choice problems in road networks with multiple user classes and multiple parking facilities. Transportation Research Part B: Methodological 40, 368–395.

Moradijoz, M., Parsa Moghaddam, M., Haghifam, M., Alishahi, E., 2013. A multi-objective optimization problem for allocating parking lots in a distribution network. International Journal of Electrical Power & Energy Systems 46, 115–122.

Pierce, G., Willson, H., Shoup, D., 2015. Optimizing the use of public garages: Pricing parking by demand. Transport Policy 44, 89–95.

Ryan, J., 1979. Urban Transportation Planning System (UTPS): The Community Aggregate Planning Model (CAPM) users' guide. Report - Urban Mass Transportation Administration, Federal Highway Administration, [Office of Highway Planning], Urban Planning Division, Technical Branch.

Sheffi, Y., 1985. Urban transportation networks: Equilibrium analysis with mathematical programming methods. 1 ed., Prentice-Hall, Englewood Cliffs.

Shen, T., Hua, K., Liu, J., 2019. Optimized public parking location modelling for green intelligent transportation system using genetic algorithms. IEEE Access 7, 176870–176883.

Van der Waerden, P., Timmermans, H., De Bruin-Verhoeven, M., 2015. Car drivers' characteristics and the maximum walking distance between parking facility and final destination. Journal of Transport and Land Use 10.