



# Population-based simulation optimization for urban mass rapid transit networks

David Schmaranzer<sup>1,2</sup> · Roland Braune<sup>1</sup> · Karl F. Doerner<sup>1,2,3</sup>

Published online: 10 May 2019  
© The Author(s) 2019

## Abstract

In this paper, we present a simulation-based headway optimization for urban mass rapid transit networks. The underlying discrete event simulation model contains several stochastic elements, including time-dependent demand and turning maneuver times as well as direction-dependent vehicle travel and passenger transfer times. Passenger creation is a Poisson process that uses hourly origin–destination-matrices based on anonymous mobile phone and infrared count data. The numbers of passengers on platforms and within vehicles are subject to capacity restrictions. As a microscopic element, passenger distribution along platforms and within vehicles is considered. The bi-objective problem, involving cost reduction and service level improvement, is transformed into a single-objective optimization problem by normalization and scalarization. Population-based evolutionary algorithms and different solution encoding variants are applied. Computational experience is gained from test instances based on real-world data (i.e., the Viennese subway network). A covariance matrix adaptation evolution strategy performs best in most cases, and a newly developed encoding helps accelerate the optimization process by producing better short-term results.

**Keywords** Simulation-based optimization · Public transportation · Headway optimization · Transit network frequencies setting problem · Population-based metaheuristic

---

✉ David Schmaranzer  
david.schmaranzer@univie.ac.at

Roland Braune  
roland.braune@univie.ac.at

Karl F. Doerner  
karl.doerner@univie.ac.at

<sup>1</sup> Department of Business Decisions and Analytics, University of Vienna, Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria

<sup>2</sup> Christian Doppler Laboratory for Efficient Intermodal Transport Operations, University of Vienna, Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria

<sup>3</sup> Data Science @ Uni Vienna, University of Vienna, Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria

## 1 Introduction

Back in 1950, 29.6% of the world's population lived in urban areas. Since then, this percentage has increased every year, reaching 55.3% in 2018. The United Nations (2018) expects this trend to continue, such that by 2050 an estimated 68% of the world's population will be living in urban areas. North America and Europe already have reached 82.2% and 74.5% urbanization, respectively; by 2050 those values will likely be 89% in North America and over 83% in Europe. In line with these trends, Vienna's population is growing and likely will exceed two million by 2025 (Statistik Austria 2017; Hanika 2018). Such population growth (induced by a positive birth/mortality rate delta, a positive immigration/emigration delta, and influx from rural areas), combined with traffic congestion, efforts to reduce emissions, and municipal ambitions to improve the quality of life for residents (e.g., by pedestrianization, reducing auto mobile as well as truck traffic, etc.), as well as tourism, make it necessary to readjust urban public transportation systems constantly. Strategic and tactical planners must determine whether existing provisions are effective, now or in future.

The term *headway* refers to the time difference between consecutive vehicles (e.g., 3 minutes). Changes to headway (or its inverse *frequency*, defined by vehicles per unit of time) might lead to overcrowded train stations and vehicles, unless they are carefully planned. This challenge constitutes the *transit network frequencies setting problem (TNFSP)*, for which the solution demands a balance between capital and operational expenditures (including infrastructure preservation and potential expansion) with passenger satisfaction (i.e., service level). A balance between these conflicting goals produces optimal time-dependent headways for each passenger line.

The planning process for public transportation usually proceeds in the following order: (1) network and line planning, (2) frequency (i.e., headway) setting, (3) timetabling, (4) vehicle scheduling, (5) duty scheduling, and (6) crew rostering (Ceder and Wilson 1986; Ceder 2001; Guihaire and Hao 2008; Liebchen 2008). In classical planning approaches, the earlier planning stage provides the input for the subsequent tasks. The second step provides the optimization of headways. Comprehensive surveys on the matter of public transport planning (including problems, objectives, and solution approaches) are available in Guihaire and Hao (2008), Farahani et al. (2013) and Ibarra-Rojas et al. (2015). Guihaire and Hao (2008) and Ibarra-Rojas et al. (2015) review 26 contributions dealing with headway optimization, showing that early works assume fixed demand and are based on analytic models (Newell 1971; Salzborn 1972; Schéele 1980; Han and Wilson 1982). Six of the 26 works (Shrivastava et al. 2002; Shrivastava and Dhingra 2002; Yu et al. 2011; Huang et al. 2013; Li et al. 2013; Wu et al. 2015) and Ruano et al. (2017) more recently, employ genetic algorithms for the headway optimization problem. Most of them also use non-linear models, though Li et al. (2013) offer a simulation model. Evolutionary algorithms have already been applied to this kind of problem in similar settings (Zhao and Zeng 2006; Guihaire and Hao 2008; Yu et al. 2011), though only a few contributions address time-dependent

demand (Niu and Zhou 2013; Sun et al. 2014). Herbon and Hadas (2015) focus on morning and afternoon peaks and use a generalized newsvendor model. Whilst the last two mentioned works as well as others (Ceder 1984, 2001) focus on the optimization of a single line, we are attempting to optimize several intersecting lines (i.e., a whole network) at once (Yu et al. 2011). Following the sparse contributions that apply simulation to a problem-specific context (Vázquez-Abad and Zubietta 2005; Mohaymany and Amiripour 2009; Ruano et al. 2017), we turn to simulation-based optimization. This approach has proven successful in similar application contexts (Osorio and Bierlaire 2013; Osorio and Chong 2015; Chong and Osorio 2018).

Several problems arise when devising a model of a complex service system like an urban mass rapid transit network. The data pertaining to the structure of an existing transportation system (e.g., a subway network's lines and stations) are relatively easy to obtain, but passenger data are not. In order to model demand, we need to know how many passengers want to travel from one specific location to another and when. To the best of our knowledge, related contributions use count data (Ceder 1984), smart card data (Pelletier et al. 2011) and mobile phone data (Friedrich et al. 2010). For a study on various technologies used in pedestrian counting and tracking see Bauer et al. (2009). We employ hourly origin–destination-matrices, originally created by the MatchMobile project (IKK 2017), and infrared count data. It is also difficult to gauge passenger behavior, especially how they decide which route to take (Agard et al. 2007). Raveau et al. (2014) show that there are also regional distinctions. For reviews on route choice the reader is referred to Bovy and Stern (1990) and Frejinger (2008).

The contribution of the paper is threefold: We propose a detailed discrete event simulation model for urban mass rapid transit networks that is inspired by a real-world case but generic enough to fit other cities' rail-bound public transport networks. In this model, the demand and vehicle turning maneuver times are time-dependent, and the vehicle travel and passenger transfer times are direction-dependent. Second, as a microscopic element, passenger distribution along platforms and within vehicles is considered. We employ four different evolutionary algorithms to the associated headway optimization problem, embedding them in a simulation-based optimization framework. That is, in addition to solution encoding variants (continuous values and factors, discrete values), we apply a newly designed problem-specific encoding. Finally, we conduct a comprehensive computational evaluation using test instances based on a real-world urban mass rapid transit network (i.e., the Viennese subway system).

Section 2 describes urban mass rapid transit networks in general. Thereafter, Sect. 3 presents the objective function (including constraints) and related aspects. In Sect. 4 we describe the solution method and its building blocks, namely, the discrete event simulation model and (heuristic) optimization algorithms, together with the employed solution encoding variants. Specific details on the Viennese subway network are presented in Sect. 5, and then in Sect. 6 we explain the computational experiment setup, real-world test instances, and algorithm parameter tuning. Thereafter, Sect. 7 presents and discusses results before Sect. 8 concludes and proposes some possible future research.

## 2 Challenges in urban mass rapid transit networks

Urban mass rapid transit networks (e.g., subway, metro, tube, underground, heavy rail) are a critical component of successful cities. They were introduced to improve movement in urban areas and reduce congestion (Roth et al. 2012; Anderson 2014). These motives remain valid but—as mentioned back Sect. 1—also are reinforced by pedestrianization, tourism, and environmental demands.

These network can be modeled by a directed graph, as in Fig. 1 for two lines. The three interacting entities are the *stations*, *vehicles*, and *passengers*. One line is gray, the other one is white. Both lines contain five stations each ( $s_0$  to  $s_4$  and  $s_5$  to  $s_9$ ). Vehicles only move on the black continuous arcs. At end-of-line stations ( $s_0$ ,  $s_4$ ,  $s_5$ , and  $s_9$ ), additional directed arcs allow vehicles to perform a turning maneuver and possibly start anew. With the exception of stations  $s_2$  and  $s_7$ , all stations have their own geographical location  $g$ ; the others intersect at  $g_2$ . In this case, additional arcs (gray, dotted) allow passengers to move to another line's waiting platform. Initially, each passenger has a geographic origin and a destination (e.g.,  $g_0$  to  $g_8$ ) that defines a path that contains the crucial stations (e.g.,  $s_0$ ,  $s_2$ ,  $s_7$ ,  $s_9$ ). The path allows passengers to know where to begin their journey ( $s_0$ ) and where to exit a vehicle because they reached their final destination ( $s_9$ ) or must perform a transfer ( $s_2$  to  $s_7$ ).

A comprehensive work on the evolution of the structure of the world's largest subway networks, states that 25% of cities with a population of one million, 50% of ones with two million and all cities with a population above 10 million host them (Roth et al. 2012). In some cities, they have existed for more than a century and there are significant similarities among the different networks, despite the unique cultures, economies, and historical developments in each city. That is, most networks consist of a set of stations delimited by a *ring-shaped core* from which *branches* grow extend beyond the city. Subway networks also tend to experience similar peak times, in the morning and afternoon, reflecting increased demand by employees traveling to or from their workplaces (Sun et al. 2014). The topology and demand fluctuations of the Viennese subway network (Sect. 5) do not deviate from the general case.

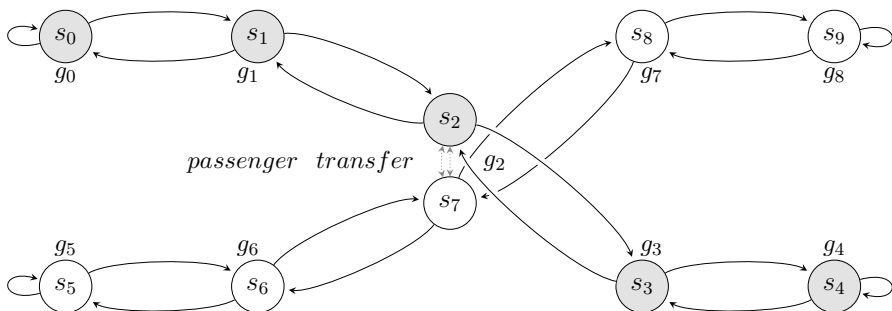


Fig. 1 Example of two intersecting lines (directed graph)

In turn, the same issue confronts all subway systems: Fluctuating demand must be satisfied by adjusting and readjusting transport capacity (i.e., releasing enough vehicles). Branches usually experience lower demand, so turning every second vehicle a few stations prior the end-of-line station may be practiced. The result being that only half the headway (e.g., 10 minutes) is used on the affected part; the remaining parts still have the full headway (e.g., 5 minutes) in effect (i.e., are visited by every vehicle). This method can also be limited to certain time periods (e.g., during off-peak hours). This and similar extensions to the problem (e.g., having more than two end-of-line stations per line) are not part of this contribution. Such (potentially temporary) measures represent an additional intermediate step in the planning process, between frequencies setting and the creation of a schedule.

### 3 Headway optimization and related aspects

As mentioned in Sect. 1, there are two conflicting goals: *cost minimization* (measured in productive fleet mileage) and *service level maximization* (measured in mean waiting time per passenger). We seek an appropriate trade-off that achieves optimal hourly headways for each line of an urban mass rapid transit network. Equation 1 contains the objective function, and Table 1 lists the notation. We employ traditional normalization and weighted sum-based scalarization approaches and thereby transform a bi-objective into a single-objective optimization problem.

$$\min Z = \left( \frac{m - m_{\min^*}}{m_{\max} - m_{\min^*}} \cdot \varphi \right) + \left( \frac{w - w_{\text{opt}}}{w_{\max^*} - w_{\text{opt}}} \cdot (1 - \varphi) \right). \quad (1)$$

Note that the first part of the objective function (Eq. 1), related to fleet mileage, is deterministic (i.e., not subject to randomness). The second part, involving the mean waiting time per passenger, is stochastic, due to the randomness of passenger

**Table 1** Notation

$m$	...	Productive fleet mileage of the current solution
$m_{\min^*}$	...	Lowest observed productive fleet mileage
$m_{\max}$	...	Highest productive fleet mileage at tightest possible headway
$w$	...	Mean waiting time per passenger of the current solution
$w_{\text{opt}}$	...	Lowest mean waiting time per passenger at tightest possible headway
$w_{\max^*}$	...	Highest observed mean waiting time per passenger
$\varphi$	...	Weight (i.e., ratio between fleet mileage and mean waiting time)
$S$	...	Set of stations
$P_s$	...	Set of platforms at station $s$
$u_{sp}$	...	Utilization of platform $p$ at station $s$
$l$	...	Number of lines (i.e., network variant; see Sect. 6.1)
$d$	...	Number of decision variables per line (see Sect. 6.1)

creation (Poisson process) and other stochastic influences (e.g., vehicle travel times between stations, passenger transfer times; see Sects. 4.1 and 5 ). Therefore, the weight  $\varphi$  influences the variance of the objective value  $Z$ . Replications (i.e., simulation re-runs) of the simulation model (Sect. 4.1) are required to account for statistical significance. We employ a varying number of replications, so its average varies, and there is a negative correlation with  $\varphi$ .

The sole constraint type is the stations' respective platform utilization  $u_{sp}$ . If a platform's capacity (e.g., two people per square meter) is exceeded, the solution is considered infeasible. Vehicle capacity is limited too, but it does not directly cause infeasibility, because waiting passengers who are unable to board an overcrowded vehicle continue waiting, increasing  $w$  (i.e., the mean waiting time per passenger).

The aforementioned reviews (Sect. 1) by Guihaire and Hao (2008) and by Ibarra-Rojas et al. (2015) contain several other objectives and constraints used in similar works, some of which we would like to discuss. A classic objective and/or constraint in headway optimization is fleet size. It can be used as minimization objective (Salzborn 1972), its current/future number (i.e., maximum fleet size) could serve as a constraint (Furth and Wilson 1981; Han and Wilson 1982). Because its former use (i.e., minimizing the number of simultaneously active vehicles) is only driven by the peak hours, this—especially its sole use—was no viable option. Using the fleet size as a constraint is reasonable, especially when it serves as a given “budget” (i.e., fixed limit). Since our efforts are driven by both, scientific and practical purposes, we decided to use the resulting fleet size as key performance indicator (KPI) but not as a constraint. Similarly, vehicle runs measure the number of times a line's vehicles go back and forth (Ceder 1984). Unlike the resulting fleet size, it is not driven by peak hours. Unfortunately it disregards the different lines' respective lengths. We use the productive fleet mileage in our objective function for those reasons. Furthermore, we have access to costs rates (operating and maintenance costs per kilometer) of a metropolitan public transport provider.

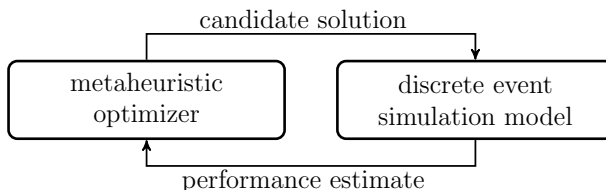
As for the customer's view (i.e., service level), some studies use total or average times like travel time (Schéele 1980; Dollevoet et al. 2015) or waiting time (Furth and Wilson 1981). An extensive sensitivity analysis of a previous version of the simulation model (Schmaranzer et al. 2016) revealed that—in our case—the travel time is only driven by the waiting time, and that a passenger's invehicle time would only increase in bunching situations (i.e., too many vehicles within a line). Since our current model ensures a minimum headway, this is no longer the case. So the invehicle time stays as is and the same goes for the transfer time. The reason for this being that alighting passengers do not use space on the platform and there are no limitations on concurrent transfer operations. For more details on both the reader is referred to Sects. 4.1 and 5. Han and Wilson (1982) use the maximum occupancy level at maximum load point along the line and relates to load and max load methods (Ceder 1984) which can only be applied to one single line at a time. Identifying each line's maximum load point within an extensive network can be difficult, especially when a vehicles utilization at a certain station is approximately identically to neighboring ones. Furthermore, the point may also depend on the direction. In case all points can be identified, the problem of treating all passengers the same remains. So in order to provide a fair measure for the service level we decided to use the mean waiting time

per passenger. The whole network's station and vehicle utilization could be used as key performance indicators (Schmaranzer et al. 2016). But since these figures would be drastically reduced (i.e., driven) by low utilization at the outer stations, they are not used in this work. However, utilization is also an important issue and is discussed in connection with platforms and vehicles in Sect. 4.1.

Another important but less tangible factor is politics. Since urban mass rapid transit networks are usually public property, every decision is potentially subject to public debate. The sheer number of stakeholders (e.g., the subsidized urban public transportation provider, adjacent residents and shopkeepers, community, regional and possibly federal politicians, railway construction companies, etc.) leads to potential conflicts. Even within the same stakeholder group tensions could arise. Since the focus of this contribution is to rely on tangible facts and figures its influence has not been implemented within the model. Our approach of investigating the trade-off between costs (i.e., the provider's view) and service level (i.e., the customer's view) and other key performance indicators (e.g., fleet size) driven by choice of headway, imply political aspects.

## 4 Optimization methodology

A subway system constitutes a queuing network with synchronization and non-exponentially distributed service times. It is thereby too complex to merely apply analytic methods from queuing theory like Jackson networks (Jackson 1963) and its extensions. Instead, we employ *simulation-based optimization*, as introduced by Fu (2002). As a functional principle (Fig. 2), an optimizer (i.e., algorithm)—in our case various evolutionary algorithms—generates candidate solutions that enter into a simulation model for evaluation. The result of this process (i.e., the solution quality and feasibility status) then reenters the optimizer, which generates new and hopefully better solutions. For current reviews on the general method, the reader is referred to Juan et al. (2015) and Amaran et al. (2016). This simulation-based optimization thus comprises a discrete event simulation model (Sect. 4.1) and (heuristic) optimization algorithms (Sect. 4.2).



**Fig. 2** Functional principle of simulation-based optimization (Fu 2002)

## 4.1 Discrete event simulation model

In addition to this detailed description of the discrete event simulation model of urban mass rapid transit networks, we provide details (e.g., parametrization, data sources, data preparation, distribution fitting, model validation) for the Viennese subway network in particular in Sect. 5. The model presented here is an extension of the ones we describe in Schmaranzer et al. (2016, 2018).

The model's three main entities are *stations*, *vehicles* and *passengers*. The remainder of this section describes them and their interactions in greater detail.

Each *station* is assigned to a specific line and has either one island platform or two separate platforms. The island platform variant potentially serves and shares its capacity with both directions. If there are separate platforms, or an island platform has an impassable wall in the middle, half of its capacity applies to each direction. The model distinguishes between platforms with and without shared capacity. The respective platform depends on its surface area (excluding safety distance between the waiting passengers and moving vehicles).

*Vehicles* (e.g., subway trains) are created and released from the lines' end-of-line stations and in accordance with the lines' respective current headway setting, during some period of operation (e.g., 4:30 until 1:00). Their capacity is limited and could be divided into seating and standing. The *vehicle travel time* refers to the time difference between a vehicle's arrival time at two consecutive stations, so it includes the dwelling time at the first station. *Dwelling time* refers to the time difference between a vehicle's arrival at a station and its departure. It includes *boarding and alighting time*, defined as the time difference between a standing vehicle opening its doors, thereby allowing aboard passengers to alight and waiting passengers to board the vehicle, and closing them again. Statistical analyses of vehicles' station arrival times support calculations of vehicle travel times. This model employs log-normal distributed vehicle travel times (see Sect. 5). Once a vehicle has reached one end of a line, it first remains there, to account for dwelling time at the last station. Thereafter, the productive fleet mileage  $m$  of the current solution increases by the respective line's length, and the vehicle must perform a turning maneuver. In some cases, this turning can occur simply by crossing over to the other direction's rail, prior to the arrival at the last station. However, in most cases—according to the infrastructure of the respective end-of-line station—a turning maneuver is required. We use triangular distributions for both dwelling and turning maneuver time at the last station.

For passenger distribution, we divide platforms and vehicles into sections. This macroscopic aspect is going to be described in the next paragraph on passengers.

Last, *passenger* entities: Their creation is driven by a time-dependent Poisson process and requires hourly origin–destination-matrices. An origin–destination-pair represents the number of passengers who wish to travel, within a certain time frame (e.g., from 8:00 until 8:59), from one geographic location to another. Each newly created passenger is assigned a path, generated by Dijkstra's shortest path algorithm (Dijkstra 1959). Their calculation requires a graph of the network, similar to the one depicted in Fig. 1. Non-transfer arcs (i.e., the ones vehicles use) need weights (e.g., vehicle travel times or distance between adjacent stations). The transfer arcs, used



by passengers, require penalty weights to prevent unnecessary transfers. Provided that the platform is not overcrowded, a new or transfer passenger gets assigned to a platform's section and potentially boards the vehicle's corresponding section. If the platform section has no free capacity, the passenger moves on to the neighboring section. In a case in which the overcrowded section has two direct neighbors (i.e., middle section), the chance that the passenger moves to the first neighbor is 50%. If a vehicle's section is overcrowded, the passenger is forced to continue waiting on the platform, thereby increasing the mean waiting time per passenger ( $w$ )—see Sect. 3. For information on this feature's effect on the Viennese subway network the reader is referred to Sect. 5.

Once the vehicle has reached a passenger's final destination or transfer station, the passenger alights. Realistic passenger transfer times can be implemented by measuring the distances of all reasonable combinations (i.e., no transfers within the same line) at a geographic location with several stations, in accordance with the findings of Weidmann (1994) that the mean walking speed of passengers is 1.34 meters per second, with a deviation of  $\pm 19\%$ . The model employs the triangular distribution with  $\pm 20\%$  of the calculated mean as minimum/maximum. The different types of platforms, which either serve one or both directions of a line, mean that passenger transfer times are potentially direction-dependent.

Outside of operational hours, no more vehicles are released. A simulation run ends once no more vehicles are active (i.e., the last one has reached its current direction's end-of-line station). Since there are several stochastic elements, replications are required to account for statistical significance. We employ a varying number of replications with a minimum of three and a maximum of 50 replications. The weight  $\varphi$  used in the objective function (Eq. 1) influences the standard deviation of the objective value  $Z$ , so as noted previously (Sect. 3), the average number of replications varies, producing a negative correlation with  $\varphi$ . The sequential evaluation process terminates once a 99.9% confidence interval with a relative error of one percent on the mean of the objective value  $Z$  has been constructed. If a platform's capacity is exceeded the simulation is terminated, the solution is deemed infeasible, and no (more) replications are performed.

This discrete event simulation model could be used for other urban mass rapid transit systems as well. The biggest obstacle would be the hourly origin–destination-matrices. Data about vehicle and platform capacity, passenger distribution, vehicle transfer times, and turning maneuver times could be difficult to ascertain too, especially without a public transportation provider's support. The network itself (i.e., lines and their stations) and data about vehicle capacity can be found easily. The implementation of vehicle travel times for transit networks who are potentially affected by road traffic (i.e., tramway and bus transit) are likely time-dependent and would require custom distributions.

The model was developed in [AnyLogic 7.0.3](#) (64 Bit, Linux) and uses some additional Java libraries ([JGraphT 1.0.1](#), [Apache POI 3.15](#), [Apache Math 3.6.1](#)).

## 4.2 Heuristic optimization algorithms

For optimization population-based evolutionary algorithms seek inspiration from nature and the theory of evolution and natural selection (Darwin 1859). A set of solutions, called a population, develops over time and ideally becomes better and better (i.e., less fleet mileage, lower mean waiting time per passenger). Solutions that do not perform well, are removed from the population and replaced by new ones, according to various evolutionary operators such as, *crossover* (i.e., breeding new offspring based on the best solutions) and *mutation* (i.e., adding a bit of randomness to the gene pool).

Four pertinent evolutionary algorithms for this study are (1) the (standard) genetic algorithm (GA, see Holland 1975), (2) the offspring selection genetic algorithm (OS-GA, see Affenzeller et al. 2009), (3) the relevant alleles preserving genetic algorithm (RAP-GA, see Affenzeller et al. 2007), and (4) the covariance matrix adaptation evolution strategy (CMA-ES, see Hansen and Ostermeier 2001)

For details on these algorithms we refer to the above cited articles and books. Schmaranzer et al. (2018) apply and age-layered population structure genetic algorithm (ALPS-GA, see Hornby 2006) and its combination with the OS-GA (i.e., the ALPSOS-GA), to the problem in a continuous value encoding setting, but the other algorithms performed better, so these options are not included herein.

The (standard) GA is the most basic one, implementing the principle as outlined above. Basically, the OS-GA aims for new populations in which a certain percentage of the offspring must be of better quality than their parents, thereby attempting to ensure improvement. The RAP-GA is based on the OS-GA and allows population size alterations within a certain range. As long as new and, with reference to the preceding population, better offspring can be created, the population size may grow, up to a maximum size. Last, the CMA-ES generates offspring not directly by crossover, but by a sophisticated sampling approach. New candidate solutions are sampled according to a multivariate normal distribution, which increases the chance of creating better offspring by constantly updating a covariance matrix that represents the pairwise dependencies between variables.

As for the three GA variants, the following four crossover operators (designed for integer and real numbers) have been used: An *average crossover* calculates an average value of two parents' values at the respective position of their gene material. An *arithmetic crossover* randomly performs an average calculation or simply takes the value from the first parent. The *blend alpha* and *blend alpha beta crossover* (Takahashi and Kita 2001) both calculate an interval and use it as boundary for a new random value. In case of the blend alpha beta crossover, the interval is guided beyond the better parent's value, so this is the only crossover operator that takes into account the parent solutions' qualities. For each new offspring to be created, one of these crossover operators is chosen at random.

All algorithms include the currently employed headways and base the creation of the first population on it. For our study, this *base solution* is the one created and employed by the Viennese public transportation provider. Schmaranzer et al. (2016) conclude that  $\pm 20\%$  headway alterations are within reason. So the base solution

refers to the very first individual in the initial population, and the remaining ones are generated by applying a normal distribution with the currently employed headways as mean and 20% standard deviation.

We use four different solution encoding variants. *Continuous values* (i.e., headways) have a lower bound of 1.5 and an upper bound of 20 minutes. These global bounds ultimately apply to all encoding variants. Next, *continuous factors*: Here factors instead of values are applied to the currently employed headways. Bounds for the factors are calculated in advance, thereby assuring that the resulting headway values are within the aforementioned global bounds. The third solution encoding variant uses *discrete values*. Its bounds within the optimizer are 30 and 400. A solution using this encoding is divided by 20 and converted back to continuous values within the discrete event simulation model. Consequently, the global boundaries again remain intact, and a step size of 0.05 minutes is introduced. Last, a *mixed* version of the continuous value and factor solution encoding variants uses a value for the first decision variable of each line. All other decision variables of the line in question are factors that apply to the result of its predecessor. For example: {10, 0.75, 0.5, 1.0, ...} results in the following headways: {10, 7.5, 3.75, 3.75, ...}. As for bounds, the value decision variables have to be within the global bounds, and factors within 0.5 and 2.0. An additional bounds checker ensures that the resulting headways are within global bounds. The idea behind this encoding scheme is to explicitly introduce dependencies between consecutive entries (i.e., headways) in the vector of decision variables.

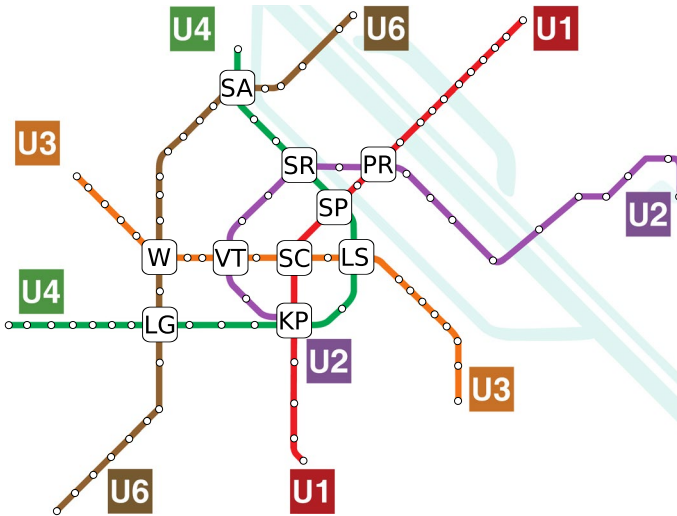
For software, we used several libraries from [HeuristicLab 3.3.15](#) (Wagner et al. 2014), which is a metaheuristics framework developed in C#.

## 5 The Viennese subway network

Table 2 contains some facts and figures about the Viennese subway network. As of 2016, it had a total length of 78.5 km and spanned five lines. The missing U5 line will be constructed between 2019 and 2024. Figure 3 depicts the whole subway network as of 2016. There are 104 stations at 93 locations, and at ten of them, two or three (single case, Karlsplatz KP) lines intersect. The SC marks Stephansplatz, the city center, and its renowned landmark St. Stephen's Cathedral.

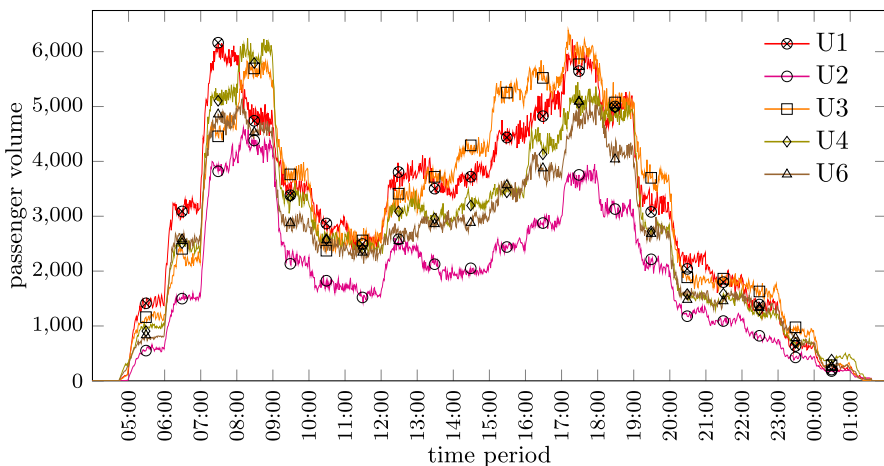
**Table 2** Facts and figures on the Viennese subway system

Line name	Line color	No. of stations	Line length [km]	Ø Station distance [m]
U1	Red	19	14.54	808
U2	Purple	20	16.86	887
U3	Orange	21	13.40	670
U4	Green	20	16.36	861
U6	Brown	24	17.34	754
Total		104	78.50	793



**Fig. 3** Schematic plan of the Viennese subway network as of 2016 (crossing stations are marked with abbreviated station identifiers)

During the course of a regular weekday (ordinary work and school day; Monday to Thursday), there are 1.37 million passenger movements. Figure 4 depicts a simulation of the passenger volume with the currently employed solution. There are two peaks: between 7:00 and 9:00 and between 16:00 and 19:00. The U1 and U3 lines carry the highest passenger volume, followed by U4 and U6; U2 has the lowest passenger load. The U1 has a higher morning peak between 7:00 and 8:00, whereas the others' highest peak is between 8:00 and 9:00. Then during the afternoon peak, U3 experiences the highest volume. This is due to passengers not only



**Fig. 4** Passenger volume over time (simulation of cur. employed headways)

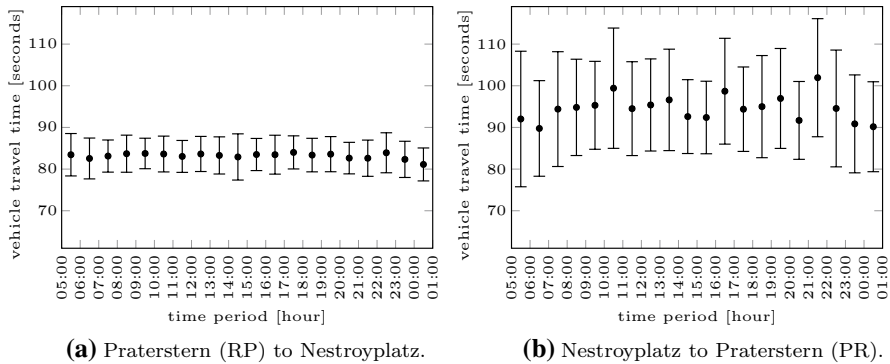
moving from home to work and then straight back again. Between the U3 stations Westbahnhof (W) and Volkstheater (VT), a pedestrianized shopping area named Mariahilfer Straße serves as alternate stop for many passengers. The passenger volume reaches 24,000 in the 2-h morning and 23,000 in the 3-h afternoon peak.

The remainder of this section is structured in the manner of Sect. 4.1 (where the general case is specified) and describes the three main entities *stations*, *vehicles* and *passengers* and their interactions in context of the Viennese subway network.

Among the 104 *stations*, 47 have an island platform without a physical barrier and the remaining 57 have either two separate platforms or an island platform with a physical barrier, creating separate capacities. The platform capacity is two people per square meter, and it depends on its surface area (excluding about half a meter safety distance between waiting passengers and moving vehicles).

*Vehicles* (i.e., subway trains) are created and released from the lines' end-of-line stations and in accordance with the lines' respective current headway setting, during their periods of operation (e.g., 4:30 until 1:00). Their capacity depends on its vehicle type. Since the old U type will be replaced soon, we focus on the remaining two (type V and T). The U1 to U4 lines are served by the vehicle type V, which holds up to 878 people. The U6 line is served by type T, which holds up to 776 people. However, 100% vehicle utilization is highly undesirable and unrealistic, because at each stop, some people would have to make room and even temporarily get out to allow alighting passengers to leave the vehicle. Furthermore, passengers might carry a bag, luggage, baby carriage, or bike with them. To account for these considerations, we reduce the vehicle type capacity by about 20%, to 702 and 621 passengers, respectively. Their top speed is 80 km/h, and their mean speeds during operation are 33.0 km/h (V type) and 30.6 km/h (T type).

A comprehensive statistical analysis of vehicles' station arrival times (about 2500 samples each) reveals that vehicle travel time is not time- but direction-dependent, as we show in Fig. 5. Conventional wisdom might suggest that the vehicle travel time suffers from the effects of increased passenger volume at peak hours (i.e., increased dwelling time), but instead, whenever the dwelling time is longer or shorter than expected, the driver compensates for by changing the vehicle's speed. The top speed of 80 km/h represents a hard limit though. However, there seems to be enough buffer time within the schedule and enough discipline amongst passengers not to cause too many non-compensable delays (e.g., by deliberately blocking the doors). Nonetheless, planned and unplanned disruptions do occur in real urban mass rapid transit networks. For works on disruption management in general and recovery models on railway networks in particular, the reader is referred to Yu and Qi (2004) and Cacchiani et al. (2014), respectively. As for Vienna, Kiefer et al. (2016) address the subject. For works on the related topic of delay management (including deciding whether a vehicle should wait for feeder vehicles) the reader is referred to Dollevoet et al. (2015) and Corman et al. (2017). The Viennese subway system does not suffer from disruptions caused by surface traffic, so these aspects were not considered in this study. However, these topics and other considerations (e.g., passengers with medical conditions) represent an opportunity for our future studies.



**Fig. 5** Hourly mean vehicle travel time and deviation at U1 Praterstern (PR)

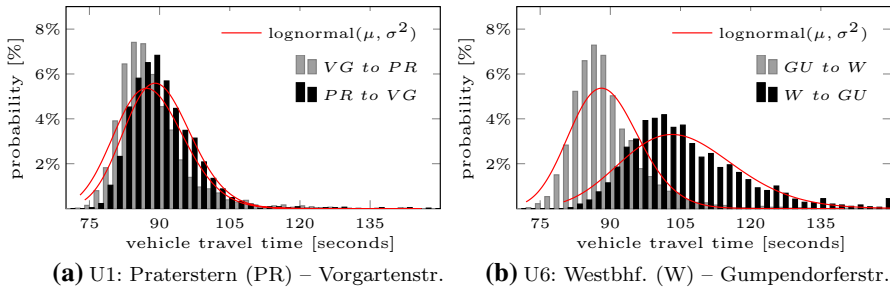
The analysis also shows that the travel time depends on the direction in which the respective vehicle is moving. Figure 5a, b illustrate this predicament: in both cases, there is no correlation between the vehicle travel time and peak hours. Most vehicle travel times over time look like Fig. 5a. Figure 5b on the other hand is the most fluctuating one. According to the Viennese public transportation provider this fluctuation results from driver changes at U1 Praterstern (PR) in the north-eastern direction. Thus Praterstern (PR) to Nestroyplatz<sup>1</sup> takes an average vehicle travel time of about 84 seconds, whereas the opposite direction takes 10 seconds more and has a higher deviation. What happens is that drivers are eager to finish their last tour, but once they arrive at Nestroyplatz, they realize they are a bit early and decide to stay longer or decrease their speed to arrive at Praterstern (PR) on time, which creates a certain degree of disturbance.

Figure 6 contains the frequencies of observed travel times and their respective fitted log-normal probability density function. Most vehicle travel times mimic the patterns in Fig. 6a (Praterstern (PR)—Vorgartenstraße<sup>2</sup>), with no significant differences between directions. However, some stations—especially crossing stations close to the city center—have a longer vehicle travel time away from than toward them. Figure 6b (Westbahnhof (W)—Gumpendorferstraße<sup>3</sup>) illustrates this phenomenon, which is likely due to longer dwelling times, driver changes, and vehicles moving in three dimensions (i.e., up- vs. down-hill; inner vs. outer curve). Most vehicle travel times appear almost normally distributed, with a longer tail on the right side, so we decided to use a *log-normal* distribution. Visual goodness-of-fit tests (Q–Q-plots, density-histogram plots) conducted in R provided strong support for this choice. Figure 7 contains Q–Q-plots of the black histograms from Fig. 6.

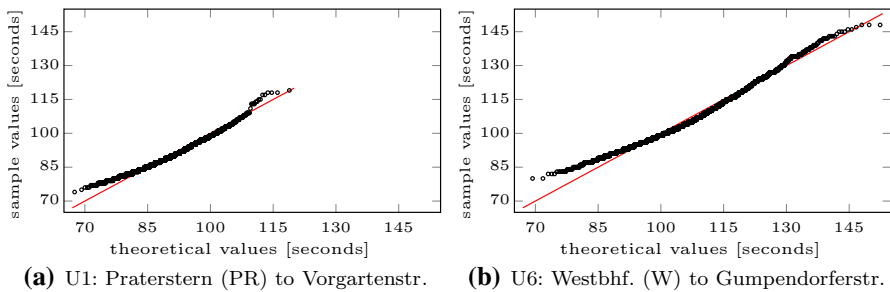
<sup>1</sup> U1 Nestroyplatz lies between station Praterstern (PR) & Schwedenplatz (SW)—Fig. 3.

<sup>2</sup> U1 Vorgartenstraße lies north-east of Praterstern (PR)—Fig. 3.

<sup>3</sup> U6 Gumpendorferstraße lies between Westbahnhof (W) & Längenfeldgasse (LG)—Fig. 3.



**Fig. 6** Frequency of vehicle travel times at Praterstern (a) and Westbhf (b)



**Fig. 7** Q-Q-plots of Praterstern (a) and Westbahnhof (b)

Once a vehicle has reached one end of a line, it remains in the station for 0.41 minutes ( $\pm 0.02$ ), accounting for about 25 seconds of dwelling time at the last station. Next, the vehicle performs a turning maneuver before it (potentially) starts anew. At some end-of-line stations, it is routine to turn vehicles after 20:30 by crossing over to the other direction's rail prior arrival. This is done to save time and vehicles. However, in most cases—depending on the infrastructure of the station in question—a turning maneuver takes 4 to 8 minutes ( $\pm 0.34$  to  $\pm 1.00$ ). We employ a *triangular* distribution for both.

Platforms and vehicles are about 120 meters long. To account for passenger distribution, both are divided into three sections (front, middle, back) of about 40 m, or two wagons each. We included this microscopic aspect at the request of the Viennese public transportation provider. More details on this and its effect are in the upcoming passenger entity description.

Last, *passenger* entities: their creation is driven by a time-dependent Poisson process that starts at 4:45 and continues until 1:00. The underlying hourly origin–destination-matrices were originally created by the MatchMobile project (IKK 2017) and are based on anonymous mobile phone data. They are from 2014 and thereby not quite up to date, so we used infrared count data, provided by our industrial partner, from 2016 to update them. They reflect the accumulated numbers of passengers alighting, boarding, and remaining onboard (i.e., occupancy) at each station and in each direction per day. The absolute difference from all three target values



(alighting, boarding, and occupancy count) per station and direction was equalized and used as optimization objective. Two decision variables (continuous factors; bounds from 0.5 to 2.0) per geographic location were introduced, resulting in 93 decision variables for manipulating the number of passengers originating from a specific location and 93 additional decision variables to allow for changing the number of passengers with a specific location as a destination. We used 48-h parallel optimization runs (one core for the optimizer, 15 cores for the simulation model) with the covariance matrix adaptation evolution strategy (CMA-ES—see Sect. 4.2) to reduce the absolute difference in the count data which resulted in about 11% more passenger volume (1.23 million in 2014, 1.37 million in 2016). This and all other experiments in this contribution were conducted on the Vienna Scientific Cluster 3 (VSC 2018), which is a high performance computing (HPC) cluster comprised of 2,020 nodes, each equipped with two Intel Xeon E5-2650v2 processors (2.6 GHz, eight cores) and at least 64 GB RAM.

As we mentioned back in Sect. 4.1, each passenger is assigned a path whose calculation requires a penalty on transfers. Every line can be reached from another line by up to two transfer operations (Fig. 3), so the weights for the transfer arcs were set to 1400 m and 3.9 minutes. Without these penalty weights on transfers, passengers would be tempted to perform unnecessary transfers (i.e., more than two). The weight for non-transfer arcs is the symmetric distance between stations or the average of the two mean values of the vehicle travel times between two stations. Full factorial experiments (prior to the aforementioned update of origin–destination–matrices) revealed that when 85% of passengers make their path decisions based on distance, and 15% base it on time, we obtain the smallest absolute count data difference; this combination also deemed realistic by our industrial partner. Simulations of the full Viennese network (see Fig. 8a) using the currently employed headways reveal that about 37% of all passengers perform one or two transfer operations.

A passenger's journey begins directly at a station's waiting platform and ends once the passenger alights from a vehicle which has reached the passenger's final destination. Hence, passengers do not interfere with transferring passengers (i.e., re-enter the waiting platform). Due to the structural individuality of stations, some of which also serve as underpasses, modeling each station's entries and passageways to platforms was no option. Provided there is still enough capacity, newly created passengers or ones which have finished a transfer operation and arrive at a waiting platform are assigned to a section. As mentioned above, the Viennese waiting platforms and vehicles were divided into three sections. The distribution is based on vehicles' doors (18 or 12 on each side for vehicle type V and T, respectively) infrared count data. The effect of passenger distribution by introducing such sections becomes especially apparent in high load situations. For example, given that the U3's headways are increased (i.e., stretched) by 20% during peak hours (7:00 to 9:00 and 16:00 to 19:00), the mean waiting time per passenger increases when compared with the case where passenger distribution is not taken into account. At the front section of Westbahnhof (W) in the west-northwest to southeast direction, the mean waiting time per passenger increases by 1.53% during the morning, by 2.78% during the afternoon, and 2.18% across both peak periods. Without headway alterations (i.e., using the currently employed headways) its effect on the whole network



is negligibly low: + 0.19%. These headways thus are quite good in terms of customer satisfaction. In order to make the model even more realistic, more sections—depending on the vehicles' number of doors per side (up to 18 for the V type and up to 12 for the T type)—could be used. However, passengers do not necessarily board and later alight from a vehicle through the same door (or its opposite on the other side), so the degree of fineness has its limits.

Passengers alight once they have finished their journey (i.e., the vehicle has reached the individuals final destination) or to transfer. Using measures of the distances between all reasonable transfer options (i.e., no transfers within a line at a specific location) and the findings of Weidmann (1994), which are that the mean walking speed is 1.34 meters per second with a deviation of  $\pm 19\%$  deviation, we calculate mean transfer times. Several stations have separate platforms (i.e., one per direction), so transfer times can be direction-dependent. The most extreme example is Volkstheater (VT) where transferring between U3 (both directions) and U2 (south-east; towards KP) takes 3 minutes, but in the other direction (north-east; towards SR), it only takes 1.75 minutes (Fig. 3). The model uses a *triangular* distribution with  $\pm 20\%$  as minimum/maximum.

We used selected key performance indicators, such as the vehicle cycle time (i.e., travel time from one end-of-line station to another) and passenger-based count data at crossing stations for validation. The deviation from the respective target values was low and approved of by the Viennese transportation provider.

Due to the huge number of samples (i.e., up to 1.37 million passenger movements), standard deviation in mean waiting time per passenger ( $w$ ) is low. In turn, we could introduce a “global denominator” that reduces the number of passenger entities and the capacities (platforms and vehicles) by a factor of ten. This step increased the standard deviation but reduced the simulation run time significantly, by a factor of about six (0.58 instead of 3.52 s per run on an Intel i7-4770 with up to 3.9 GHz), with almost negligible inaccuracy with regard to the resulting objective function values (Eq. 1).

## 6 Computational experiment setup

Table 3 contains the minimum and maximum values for *productive fleet mileage* and *mean waiting time per passenger* required for the objective function (see Eq. 1 and Table 1). The maximum fleet mileage  $m_{max}$  and optimal mean waiting time  $w_{opt}$  were easy to obtain. The technically lowest possible headway is 1.5 minutes, so we used this value for each line for each hour of operation and network variant (Sect. 6.1), thereby deriving the extreme values. Up to 37% of passengers (depending on the network variant) perform one or two transfer operations, so the optimum mean waiting time is significantly higher than the expected mean waiting time per occurrence (0.75 minutes). Due to the capacity constraint (Sect. 3), the lowest fleet mileage  $m_{min^*}$  and highest mean waiting time  $w_{max^*}$  were harder to obtain. We used 2-day long parallel optimization runs (one core for the optimizer, 15 for the simulation model) with the CMA-ES (Sect. 4.2), applied a weight  $\varphi = 1.0$  and 21 decision

**Table 3** Passenger data,  $m_{min^*}$ ,  $m_{max}$ ,  $w_{opt}$  and  $w_{max^*}$  per network variant

No. of lines $l$	Network length [km]	Passenger volume		Transferring passengers [%]	Fleet mileage [km]		Waiting time [minutes]	
		[million]	[%]		$m_{min^*}$	$m_{max}$	$w_{opt}$	$w_{max^*}$
5	78.5	1.37	100	37	20,563.75	128,738.36	1.0496	7.5678
4	61.6	1.12	82	31	15,409.31	101,094.52	1.0061	7.3869
3	44.3	0.82	60	23	10,415.11	72,658.56	0.9464	7.2957
2	27.9	0.57	42	17	6,287.78	45,826.52	0.9007	7.0751

variables per line  $l$  (i.e., network variant). The experiment was conducted on the VSC3 (VSC 2018), as introduced in Sect. 5.

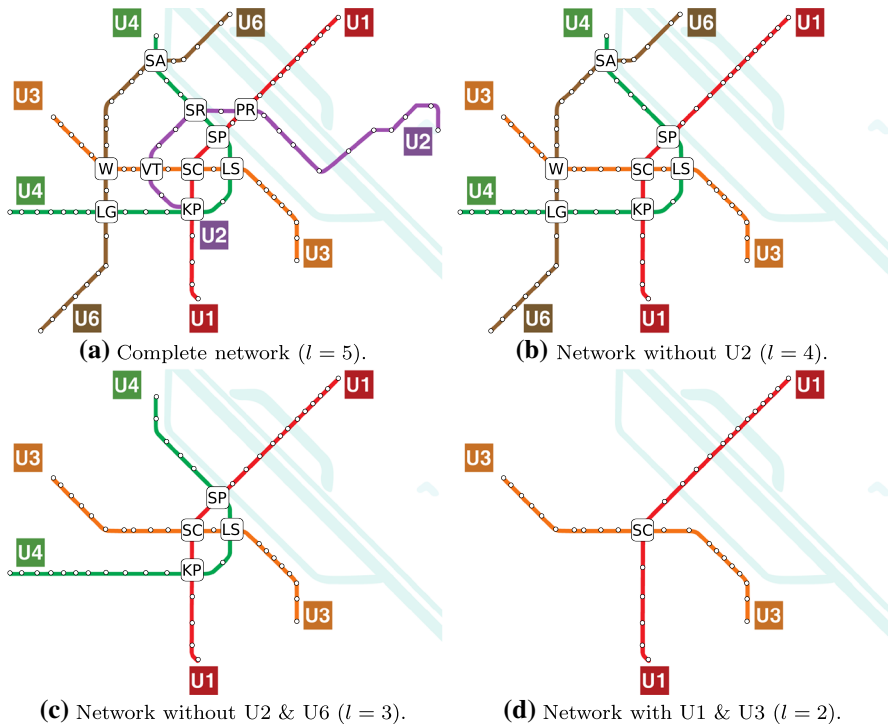
The remainder of this section is divided into an introduction on the test instances (Sect. 6.1) and algorithm parameter tuning (Sect. 6.2).

## 6.1 Real-world test instances

The solution method (Sect. 4) is applied to four different solution encoding variants (as introduced at the end of Sect. 4.2) and 48 different test instances. The 48 real-world test instances per solution encoding serve two purposes: a fast instance combination for tuning the algorithms' respective parameters (Sect. 6.2) was required. Second, several different problem instances to support comparative analyses of the effectiveness of a solution scheme are a necessity. The instances were created using four different versions of the Viennese subway network, four different numbers of decision variables per line, and applying three different weights ( $\varphi = 0.25, 0.50$  and  $0.75$ ). The first two are described in the next paragraphs. The weights are as introduced for the objective function (Eq. 1) in Sect. 3.

As for *network variants*: Fig. 8a contains the whole Viennese subway network with five lines, so we refer to it as  $l = 5$ . The other variants ( $l = 4$ ,  $l = 3$ , and  $l = 2$ ; Fig. 8b–d, respectively) are reduced versions, created by removing one line at a time, according to each line's passenger volume (see Fig. 4).

For the *number of decision variables per line*, because the origin–destination-matrices change hourly, changing headways on an hourly basis comes naturally. The Viennese subway system operates from 4:45 to 1:00 (ordinary work and school day; Monday–Thursday), so each line has 21 decision variables ( $d = 21$ ). To fill and empty the lines with vehicles, their release starts at 4:30 and ends at 1:00. In the hourly variant, the first decision variable of each line applies to the time period prior to 5:00. Other variants are 2- and 3-h long headways, which produce eleven ( $d = 11$ ) and seven ( $d = 7$ ) decision variables per line, respectively. In the smallest version, each line has four decision variables ( $d = 4$ ) and re-uses headways by means of indices which are assigned to 21 specific time periods  $\{0, 0, 1, 2, 2, 3, 1, 1, 3, 3, 3, 3, 2, 2, 2, 3, 1, 1, 0, 0, 0\}$ . The fourth and fifth as well as the 13th, 14th and 15th, for example, all have an index of two. So, the solution's line's value at this particular



**Fig. 8** Network variants (schematic plans of the Viennese subway network)

index provides the headway for the morning (7:00 to 9:00) and afternoon (16:00 to 19:00) peaks.

The total number of decision variables lies between eight and 105 ( $l \cdot d$ ), and the latter variant is referred to as *largest* or *full real-world instance*. Its was set to 10 h (i.e., 10 h of optimization on one single CPU core). The run time of the other instances was set in relation to the total number of decision variables (15 minutes accuracy). The smallest one has a optimization run time of 75 minutes.

One evaluation of all 48 instances takes about 200 h of computation on a single CPU core. Given, that four different algorithms are used, five independent optimization runs (not to be confused with replications) are performed, and four different solution encoding variants are tested, a total of almost 16,000 h is required. Up to 100 nodes (i.e., individual machines) of the VSC3 (VSC 2018) with 16 CPU cores each were used simultaneously.

## 6.2 Algorithm parameter turning

The four algorithms (as introduced back in Sect. 4.2) have various parameters that must to be tuned to fit the problem. We defined a set of reasonable values for the parameters and ran full factorial experiments, as follows. The population size of the

GA, OS-GA and RAP-GA was set to 50, 75, 100, 150, and 200. The CMA-ES usually tends to smaller population sizes, so lower ones (35, 50, 65, 80, and 100) were tested. The elites parameter (available in all but the CMA-ES) was set with respect to the population size: 1%, 5%, 10%, 15%, 20%, and one elite as a minimum resulting value. The mutation probability was set to 5%, 10%, 15%, 20%, 25%, 30%, 40%, and 50% in the (standard) GA. Because the OS-GA and RAP-GA have additional parameters, fewer mutation probabilities (10%, 15%, 20%, 30%, and 40%) could be tested. The success ratio (OS-GA) and comparison factor (RAP-GA) were set to 0.6, 0.8, and 1.0. The RAP-GA's maximum population size was set in respect to the population size (factor of 1.5 and 2.0). As for the remaining CMA-ES parameters: 0, 50, 100, 150, 200, 300, and 500 initial iterations were tested;  $\mu$  was set to null, 1, 5, and 10. The initial  $\sigma$  was set as a fraction of the parameter range, so its resulting value depends on the solution encoding's bounds, equivalent to  $1/8$ ,  $1/6$ , and  $1/4$ .

The instance used for tuning has only two lines ( $l = 2$ ) and 21 parameters ( $d = 21$ ) per line (Sect. 6.1). Therefore, this instance has 42 decision variables ( $l \cdot d$ ) in total. This combination of the network and number of decision variables per line offers the best combination of low run time—due to significantly reduced passenger volume—and still a high number of headways to be set. Tables 4 and 5 contain the tuned parameter values for all solution encoding variants. The resulting population size does not vary much in most cases. High population sizes (up to 200 here and up to 300 in Schmaranzer et al. 2018) were tested, but lower ones lead to better results, likely due to the long evaluation time. The continuous mixed encoding needs lower mutation probabilities, due to the dependence of a line's starting value to its following factors. The closer a mutation takes place at a position to a line's beginning, the stronger the effect on the solution (i.e., all remaining parameters of the affected line).

## 7 Computational evaluation

This section begins with an tabular overview of the considered solution methods' performance on all test instances (Sect. 7.1), then conclusions on the matter of which solution encoding performs best (Sect. 7.2) are presented, and a closer look into the real-world test instance with 105 decision variables algorithm's respective best results over time (Sect. 7.3) is taken. Last, Sect. 7.4 focuses on the real-world instance, and investigates the trade-off between both target measures, namely fleet mileage and mean waiting time per passenger.

### 7.1 Results overview

Tables 6, 7 and 8 contain the final results for all instances, solution encoding variants and all three applied weights  $\varphi$ . Five independent and reproducible optimization runs per variant and algorithm were performed. The currently employed headways served as a baseline (i.e., base solution), and the reported values represent average percentage deviation from that baseline. The best and second best results are

**Table 4** Tuned parameters for continuous and discrete value solution encoding

Parameter designation	Continuous values					Discrete values				
	GA	OS-GA	RAP-GA	CMA-ES		GA	OS-GA	RAP-GA	CMA-ES	
	GA	OS-GA	RAP-GA	CMA-ES		GA	OS-GA	RAP-GA	CMA-ES	
Re-evaluate elites	False	False	False	-		False	False	False	-	
Population size	75	50	50	80		100	50	50	80	
Elites in % (min. 1 elite)	20%	10%	1%	-		20%	15%	20%	-	
Mutation probability	10%	40%	20%	-		25%	30%	10%	-	
Selected parents	-	100	-	-		-	100	-	-	
Success ratio	-	0.6	-	-		-	0.6	-	-	
Maximum selection pressure	-	100	-	-		-	100	-	-	
Offspring selection before mutation	-	False	-	-		-	False	-	-	
Fill population with parents	-	False	-	-		-	False	-	-	
Min. population size	-	-	5	-		-	-	5	-	
Max. population size	-	-	75	-		-	-	75	-	
Batch size	-	-	10	-		-	-	10	-	
Comparison factor	-	-	0.6	-		-	-	0.60	-	
Effort	-	-	750	-		-	-	750	-	
Initial iterations	-	-	-	500		-	-	-	150	
$\mu$	-	-	-	10		-	-	-	10	
Initial $\sigma$ (fraction of parameter range)	-	-	-	$\frac{1}{4}$		-	-	-	$\frac{1}{4}$	
Recombinator	-	-	-	Log-weighted		-	-	-	Log-weighted	

**Table 5** Tuned parameters for continuous factors and mixed solution encoding

Parameter designation	Continuous factors					Continuous mixed						
	GA	OS-GA	RAP-GA	CMA-ES	GA	OS-GA	RAP-GA	CMA-ES	GA	OS-GA	RAP-GA	CMA-ES
	False	False	False	False	False	False	False	False	False	False	False	False
Re-evaluate elites	75	50	50	50	50	50	50	50	50	50	50	35
Population size	20%	5%	10%	10%	20%	15%	15%	15%	20%	20%	20%	20%
Elites in % (min. 1 elite)	10%	40%	10%	10%	5%	20%	20%	20%	5%	100	20%	20%
Mutation probability	-	100	-	-	-	-	-	-	-	100	-	-
Selected parents	-	0.8	-	-	-	0.6	-	-	-	0.6	-	-
Success ratio	-	100	-	-	-	100	-	-	-	100	-	-
Maximum selection pressure	-	False	-	-	-	False	-	-	-	False	-	-
Offspring selection before mutation	-	False	-	-	-	False	-	-	-	False	-	-
Fill population with parents	-	False	-	-	-	False	-	-	-	False	-	-
Min. population size	-	-	5	-	-	-	5	-	-	-	5	-
Max. population size	-	-	75	-	-	-	75	-	-	-	75	-
Batch size	-	-	10	-	-	-	10	-	-	-	10	-
Comparison factor	-	-	1.0	-	-	-	0.6	-	-	-	0.6	-
Effort	-	-	750	-	-	-	750	-	-	-	750	-
Initial iterations	-	-	-	150	-	-	-	150	-	-	-	0
$\mu$	-	-	-	NULL	-	-	-	NULL	-	-	-	NULL
Initial $\sigma$ (fraction of parameter range)	-	-	-	$\frac{1}{8}$	-	-	-	$\frac{1}{8}$	-	-	-	$\frac{1}{6}$
Recombinator	-	-	-	Log-weighted	-	-	-	Log-weighted	-	-	-	Log-weighted

**Table 6** Diff. (%) from base solution's  $Z$  at  $\varphi = 0.25$  (priority on  $\emptyset$  waiting time)  
(a) Continuous and discrete values encoding variants

No. of lines $l$	Dec. var. $l \cdot d$	Run time [h]	Continuous values				Discrete values			
			GA [%]	OS [%]	RAP [%]	CMA [%]	GA [%]	OS [%]	RAP [%]	CMA* [%]
2	8	1.25	-18.46	-18.34	-18.20	-18.70	-17.27	-17.68	-17.93	-18.80
	14	2.50	-19.96	-19.98	-19.82	-20.57	-18.71	-18.46	-19.54	-20.66
	22	2.50	-18.09	-18.44	-18.02	-18.90	-16.85	-16.12	-17.67	-18.86
	42	5.00	-17.14	-17.74	-16.74	-18.79	-15.40	-13.64	-15.79	-18.91
3	12	2.50	-20.48	-20.60	-20.48	-21.01	-19.02	-18.52	-20.05	-21.05
	21	2.50	-20.54	-20.61	-20.29	-21.54	-17.38	-16.83	-19.61	-21.79
	33	3.75	-18.83	-19.55	-18.77	-20.30	-16.61	-15.99	-18.13	-20.38
	63	6.25	-17.12	-17.96	-16.56	-19.73	-13.41	-11.61	-14.62	-19.45
4	16	2.50	-19.69	-20.89	-20.05	-21.13	-16.49	-16.66	-19.30	-21.53
	28	2.50	-16.29	-19.11	-17.05	-19.58	-14.74	-13.56	-16.30	-20.34
	44	5.00	-18.16	-19.12	-18.28	-19.71	-15.47	-13.83	-16.14	-20.32
	84	8.75	-16.05	-16.71	-16.14	-19.85	-12.19	-10.49	-13.46	-19.58
5	20	2.50	-19.66	-22.34	-20.38	-22.15	-16.90	-16.42	-18.79	-22.27
	35	3.75	-19.20	-21.39	-19.93	-14.79	-15.24	-14.64	-17.90	-16.63
	55	5.00	-18.08	-19.93	-18.31	-17.20	-14.56	-13.37	-16.85	-16.63
	105	10.00	-17.10	-16.31	-17.49	-19.39	-11.98	-10.41	-15.31	-17.10
Total		-18.43	-19.31	-18.53	-19.58	-15.76	-14.89	-17.34	-19.64	

Table 6 (continued)

(b) Continuous factors and mixed encoding variants		Continuous factors										Continuous mixed			
No. of lines $l$	Dec. var. $l \cdot d$	Run time [h]	Continuous factors				Continuous factors				Continuous mixed				
			GA [%]	OS [%]	RAP [%]	CMA [%]	GA [%]	OS [%]	RAP [%]	CMA [%]	GA [%]	OS [%]	RAP [%]	CMA [%]	
2	8	1.25	-18.28	-18.28	-17.95	-18.73	-18.45	-18.38	-18.46	-18.58					
	14	2.50	-19.93	-19.70	-19.53	-20.29	-20.44	-20.28	-20.45	-20.27					
	22	2.50	-18.31	-18.05	-17.73	-18.79	-18.35	-18.37	-18.37	-18.62					
	42	5.00	-17.48	-17.00	-15.77	-18.07	-17.84	-17.96	-18.18	-18.44					
3	12	2.50	-20.43	-20.27	-20.10	-20.96	-20.75	-20.75	-20.67	-20.95					
	21	2.50	-20.21	-19.51	-19.54	-21.20	-21.49	-21.38	-21.44	-21.52					
	33	3.75	-19.28	-18.40	-18.28	-20.22	-19.52	-19.73	-19.76	-20.10					
	63	6.25	-17.69	-16.42	-15.58	-19.91	-19.31	-19.19	-19.50	-19.73					
4	16	2.50	-19.68	-19.52	-18.82	-21.50	-21.40	-21.60	-21.47	-21.70					
	28	2.50	-15.48	-15.54	-15.56	-20.83	-21.32	-21.44	-21.47	-20.55					
	44	5.00	-17.63	-16.69	-17.10	-21.03	-20.50	-20.65	-20.64	-20.06					
	84	8.75	-15.85	-14.31	-13.71	-20.40	-19.83	-20.04	-20.19	-19.84					
5	20	2.50	-20.63	-19.76	-19.32	-23.85	-23.82	-23.67	-23.67	-23.36					
	35	3.75	-18.53	-17.10	-17.49	-23.98	-24.35	-24.58	-24.56	-23.00					
	55	5.00	-17.73	-16.26	-16.82	-22.78	-22.60	-22.86	-22.96	-20.75					
	105	10.00	-16.93	-14.81	-14.70	-22.67	-22.39	-22.89	-22.89	-21.20					
Total															
			-18.38	-17.60	-17.38	-20.95	-20.77	-20.86	-20.92	-20.54					



**Table 7** Diff. (%) from base solution's Z at  $\varphi = 0.50$  (equal priority)

(a) Continuous and discrete values encoding variants

No. of lines <i>l</i>	Dec. var. <i>l</i> · <i>d</i>	Run time [h]	Continuous values				Discrete values			
			GA [%]	OS [%]	RAP [%]	CMA [%]	GA [%]	OS [%]	RAP [%]	CMA* [%]
			2	8	1.25	-3.27	-3.05	-3.17	-3.88	-2.87
	14	2.50	-3.44	-3.59	-3.58	-4.27	-2.87	-3.19	-3.63	-4.23
	22	2.50	-3.40	-3.73	-3.29	-4.69	-2.94	-3.13	-3.28	-4.44
	42	5.00	-3.40	-3.63	-3.63	-4.92	-3.20	-2.95	-3.11	-4.44
3	12	2.50	-1.87	-1.76	-2.13	-2.65	-1.82	-1.91	-2.21	-2.39
	21	2.50	-2.35	-2.20	-2.23	-2.46	-2.07	-2.02	-2.19	-2.63
	33	3.75	-2.44	-2.75	-2.77	-3.28	-2.39	-2.40	-2.50	-3.24
	63	6.25	-2.24	-2.53	-2.19	-3.20	-1.97	-1.65	-1.74	-3.37
4	16	2.50	-2.30	-2.24	-2.22	-2.46	-2.15	-2.10	-2.11	-2.48
	28	2.50	-1.92	-1.80	-1.68	-1.96	-1.86	-1.78	-1.53	-2.11
	44	5.00	-2.68	-2.99	-2.64	-3.40	-2.60	-2.50	-2.48	-3.34
	84	8.75	-2.32	-2.83	-2.27	-3.55	-2.14	-2.09	-1.99	-3.46
5	20	2.50	-2.13	-2.14	-2.12	-2.07	-2.10	-2.05	-2.06	-2.10
	35	3.75	-2.24	-2.18	-2.23	-2.28	-2.29	-2.19	-2.18	-2.51
	55	5.00	-2.40	-2.52	-2.21	-2.71	-1.93	-2.11	-2.14	-2.61
	105	10.00	-2.28	-2.32	-2.29	-3.32	-1.78	-1.92	-2.00	-3.49
Total			-2.54	-2.64	-2.54	-3.19	-2.31	-2.31	-2.39	-3.14

Table 7 (continued)

(b) Continuous factors and mixed encoding variants											
No. of lines $l$	Dec. var. $l \cdot d$	Run time [h]	Continuous factors				Continuous mixed				
			GA [%]	OS [%]	RAP [%]	CMA [%]	GA [%]	OS [%]	RAP [%]	CMA [%]	
2	8	1.25	-2.97	-2.79	-3.15	-2.37	-3.15	-3.41	-3.06	-3.31	
	14	2.50	-3.44	-3.12	-3.49	-3.42	-3.32	-3.61	-3.57	-3.86	
	22	2.50	-3.39	-3.26	-3.70	-4.07	-2.47	-2.43	-2.73	-3.78	
	42	5.00	-3.22	-3.63	-3.53	-3.80	-2.58	-2.89	-2.99	-3.41	
3	12	2.50	-1.85	-1.89	-1.92	-1.91	-2.03	-2.09	-1.91	-2.07	
	21	2.50	-1.97	-2.10	-1.97	-2.46	-2.15	-2.03	-2.07	-2.12	
	33	3.75	-2.50	-2.57	-2.40	-3.19	-2.19	-1.93	-2.30	-2.31	
	63	6.25	-1.89	-2.30	-1.88	-3.32	-1.96	-1.94	-2.16	-2.53	
4	16	2.50	-2.15	-2.07	-2.11	-2.59	-2.31	-2.35	-2.18	-2.22	
	28	2.50	-1.89	-1.68	-1.62	-2.62	-2.03	-2.08	-1.91	-1.68	
	44	5.00	-2.79	-2.84	-2.71	-3.70	-2.38	-2.55	-2.50	-2.63	
	84	8.75	-2.37	-2.51	-2.49	-3.98	-2.41	-2.52	-2.68	-2.43	
5	20	2.50	-2.03	-1.79	-1.79	-2.61	-2.28	-2.26	-2.14	-2.01	
	35	3.75	-2.31	-1.88	-1.91	-3.16	-2.43	-2.46	-2.39	-1.36	
	55	5.00	-2.19	-1.96	-1.81	-3.36	-2.20	-2.32	-2.13	-0.98	
	105	10.00	-2.15	-1.99	-1.78	-3.99	-2.42	-2.68	-2.66	-1.96	
Total		-2.44	-2.40	-2.39	-3.16	-2.39	-2.47	-2.46	-2.42		

**Table 8** Diff. (%) from base solution's Z at  $\varphi = 0.75$  (priority on fleet mileage)

(a) Continuous and discrete values encoding variants

No. of lines <i>l</i>	Dec. var. <i>l</i> · <i>d</i>	Run time [h]	Continuous values				Discrete values			
			GA [%]	OS [%]	RAP [%]	CMA [%]	GA [%]	OS [%]	RAP [%]	CMA* [%]
2	8	1.25	-28.29	-28.73	-28.11	-29.27	-27.73	-27.61	-27.45	-29.41
	14	2.50	-27.66	-28.08	-27.80	-28.42	-27.05	-26.89	-26.50	-28.53
	22	2.50	-28.13	-29.27	-28.61	-30.22	-28.10	-28.34	-26.09	-29.88
	42	5.00	-28.04	-29.43	-29.04	-30.63	-28.14	-27.34	-25.44	-30.67
3	12	2.50	-25.98	-26.39	-26.06	-27.34	-25.87	-25.61	-24.92	-27.14
	21	2.50	-24.78	-25.54	-24.71	-26.69	-24.83	-24.27	-22.01	-26.67
	33	3.75	-26.51	-27.45	-26.22	-28.26	-26.27	-25.61	-22.83	-28.50
	63	6.25	-26.31	-27.42	-25.53	-29.16	-26.33	-24.15	-20.23	-29.04
4	16	2.50	-24.71	-25.67	-25.01	-26.45	-25.19	-25.11	-22.92	-26.64
	28	2.50	-21.80	-23.70	-22.15	-25.69	-23.63	-23.19	-18.87	-25.61
	44	5.00	-25.64	-26.86	-25.73	-28.43	-25.91	-25.34	-21.28	-28.30
	84	8.75	-25.16	-26.80	-25.24	-29.16	-25.92	-23.29	-19.60	-29.32
5	20	2.50	-20.03	-22.07	-20.97	-23.34	-21.59	-21.26	-17.88	-23.25
	35	3.75	-19.30	-21.63	-19.96	-23.34	-20.78	-19.26	-15.87	-23.54
	55	5.00	-20.03	-22.65	-20.51	-24.99	-20.61	-18.29	-17.43	-25.05
	105	10.00	-20.46	-22.10	-21.73	-26.82	-21.14	-16.56	-17.19	-27.04
Total			-24.55	-25.86	-24.84	-27.39	-24.94	-23.88	-21.66	-27.41



highlighted in bold and italics, respectively. The worse are in bold italic. The first column contains the number of lines  $l$ , the second the total number of decision variables ( $l \cdot d$ ), the third the run time limit per optimization run (one CPU core).

With regard to the results of the continuous and discrete value solution encoding (Tables 6a, 7a and 8a), the CMA-ES is designed for continuous numbers. So its discrete variant (CMA\*) uses continuous values which are rounded to discrete values just prior to the evaluation.

Optimizing the mean waiting time per passenger optimization is harder than fleet mileage optimization. So when investigating the results, we must recall that there is a negative correlation between the weight  $\varphi$  and the number of replications (Sect. 3). At weights  $\varphi = 0.25$  and  $0.75$  the number of evaluated solutions decreases by about 30% and increases by about 16%, respectively, when compared with the equal weight  $\varphi = 0.50$ . We must also consider that mean waiting time per passenger optimization is harder than fleet mileage optimization. The reason for this is that the average waiting time per passenger, is time-dependent, because so is the passenger volume (Fig. 4). So a slightly tighter headway during a peak hour may very well lead to a better result than a much tighter headway in an off-peak hour. The fleet mileage, however, is not time-dependent, because as long as a looser headway does not lead to infeasibility and reduces the number of vehicle releases, the resulting fleet mileage is lower. Of course the affected line's length has an influence on how high the savings are but the affected position in the solution vector has no influence.

At a weight  $\varphi = 0.25$  (Table 6), the CMA-ES performs best in all but the continuous mixed encoding variant, where the RAP-GA ranks first. The OS-GA ranks second in the continuous value and mixed encoding, the (standard) GA does so in the continuous factor variant. In the discrete value encoding, the RAP-GA performs second best. For the continuous mixed encoding, the performance of the CMA-ES slightly degrades, even though it still yields the largest improvements in eight of 16 test instances. By investigating each network variant  $l$  separately, it seems that a smaller number of decision variables leads to higher overall improvements with regard to the initial solution. In Sect. 7.3 we are going to discover, that the continuous and discrete value encoding variants at a weight  $\varphi = 0.25$  would require more time, i.e., there remains room for more improvements. However, the run-time budget (as introduced in Sect. 6.1) could not be extended beyond the 10 h mark to still allow for conducting optimization during night-runs at our industrial partner.

At equal prioritization,  $\varphi = 0.50$  (Table 7), again the CMA-ES performs best. Similar to a weight  $\varphi = 0.25$ , the OS-GA performs second best in the continuous value variant and the RAP-GA performs second best with regard to its discrete counterpart. Once again, the (standard) GA performs second best in the continuous factor encoding. The CMA-ES performs best, with the continuous mixed encoding being the sole exception (best only in seven of 16 cases); the RAP-GA performed best in this encoding setup.

When compared with the preceding weight variant  $\varphi = 0.25$ , the improvements are rather small. The reason for this being the currently employed headways, which offer a good balance (or are even slightly in favor of service level) between both target measures.

For this weight setting, the achieved improvement percentages notably increase with the number of decision variables (e.g.,  $l \cdot d = 28$  and  $44$ ). However, here (and especially in the mixed encoding setup) the range of resulting values is pretty narrow, which is also an additional indicator for the currently employed headways being in balance.

Last, a higher priority on mileage, a weight  $\varphi = 0.75$  (Table 8), leads the CMA-ES to win in almost all cases (60 of 64). Like in the other two weight variants, the OS-GA performs second best in the continuous value encoding setup; the (standard) GA does in its factor counterpart. No clear pattern emerges in the discrete value encoding, but similar to the findings of weight  $\varphi = 0.50$ . The number of decision variables and the achieved improvements are even more apparently positively correlated, indicating that more fine-grained headway settings are obviously better able to adapt to changes of the passenger volume. The improvements are quite high because the currently employed headways geared towards balance—as the results of  $\varphi = 0.50$  show. Furthermore, due to the weight's influence on the variance of the objective value (Sect. 3), this setting allows for more solution evaluations within the given run time budget, because less replications are required.

All in all, the CMA-ES performed best in 162 of 192 cases. It did not perform as well in combination with the mixed encoding variant (especially with a weight  $\varphi = 0.25$  and  $0.50$ ) in test instances with more lines ( $l = 4$  or  $5$ ). A possible explanation for this is that the dependence between decision variables is already part of the encoding, such that it cannot be fully and directly be captured by the CMA-ES's correlation analysis features. This encoding setting also leads to lower mutation probabilities (see Sect. 6.2). At an equal weight  $\varphi = 0.50$ , the achieved improvements (i.e., deviation from the respective base solutions'  $Z$ ) are considerably lower than in the other two weight variants. Currently employed headways already lead to a quite balanced solution. Thereby, further improvements are hard to find. In small instances, the difference among the population-based algorithms seems almost negligible.

## 7.2 Comparing overall performance of encoding variants

Table 9 contains the average values over all instances of the tabular results back from Sect. 7.1. The differences, especially with a weight  $\varphi = 0.50$ , are often very small, making it difficult to identify a clear winner.

With these results, we derive the best encoding variants in general, for each algorithm, and for each weight  $\varphi$  (Table 10). The continuous mixed encoding appears advantageous for all algorithms except for CMA-ES, which performs best with continuous factors in general, but especially at a weight  $\varphi = 0.25$ . At a weight  $\varphi = 0.50$ , all algorithms seem to prefer the classic continuous value encoding.

**Table 9** Performance comparison of all encoding variants

$\varphi$	GA [%]	OS [%]	RAP [%]	CMA [%]	Total [%]
(a) Continuous values					
0.25	- 18.43	- 19.31	- 18.53	- 19.58	- 18.96
0.50	- 2.54	- 2.64	- 2.54	- 3.19	- 2.73
0.75	- 24.55	- 25.86	- 24.84	- 27.39	- 25.66
Total	- 15.17	- 15.94	- 15.30	- 16.72	- 15.78
(b) Discrete values					
0.25	- 15.76	- 14.89	- 17.34	- 19.64	- 16.91
0.50	- 2.31	- 2.31	- 2.39	- 3.14	- 2.54
0.75	- 24.94	- 23.88	- 21.66	- 27.41	- 24.47
Total	- 14.34	- 13.69	- 13.79	- 16.73	- 14.64
(c) Continuous factors					
0.25	- 18.38	- 17.60	- 17.38	- 20.95	- 18.58
0.50	- 2.44	- 2.40	- 2.39	- 3.16	- 2.60
0.75	- 24.49	- 24.25	- 22.74	- 26.98	- 24.61
Total	- 15.10	- 14.75	- 14.17	- 17.03	- 15.26
(d) Continuous mixed					
0.25	- 20.77	- 20.86	- 20.92	- 20.54	- 20.77
0.50	- 2.39	- 2.47	- 2.46	- 2.42	- 2.44
0.75	- 24.45	- 24.93	- 25.05	- 26.19	- 25.15
Total	- 15.87	- 16.09	- 16.14	- 16.38	- 16.12

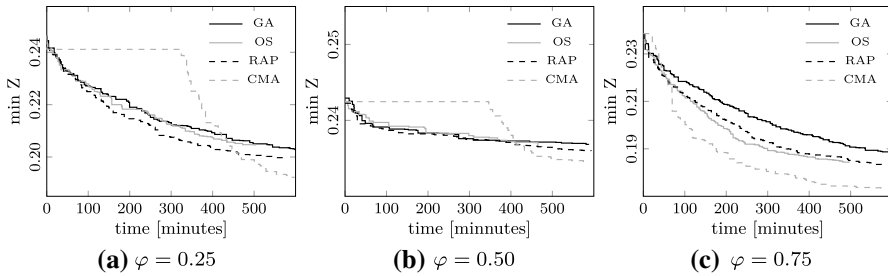
**Table 10** Best encoding in general, for each algorithm and weight  $\varphi$ 

$\varphi$	GA	OS	RAP	CMA	Total
0.25	Cont. mixed	Cont. mixed	Cont. mixed	Cont. factors	Cont. mixed
0.50	Cont. values	Cont. values	Cont. values	Cont. values	Cont. values
0.75	Disc. values	Cont. values	Cont. mixed	Disc. values	Cont. values
Total	Cont. mixed	Cont. mixed	Cont. mixed	Cont. factors	Cont. mixed

### 7.3 Details on the full real-world instance

Next, time plots depict the algorithms' best runs of the largest instance ( $l \cdot d = 105$ ). This instance is the full real-world setup, so it is the most important for our industrial partner.

Figure 9 contains the continuous value versions. At a weight  $\varphi = 0.25$  (Fig. 9a), the CMA-ES needs about 340 minutes to start its descent and about 460 minutes to exceed its competitors. The RAP-GA performs second best. The same holds for the  $\varphi = 0.50$  variant (Fig. 9b), where the CMA-ES needs about 350 minutes to start descending and about 420 to catch up. The improvement at an equal weight—as mentioned at the end of Sect. 7.1—ultimately is smaller than in the non-equal setup. When  $\varphi = 0.75$ , the CMA-ES starts a much steeper descent and performs best from minute 80 onward.

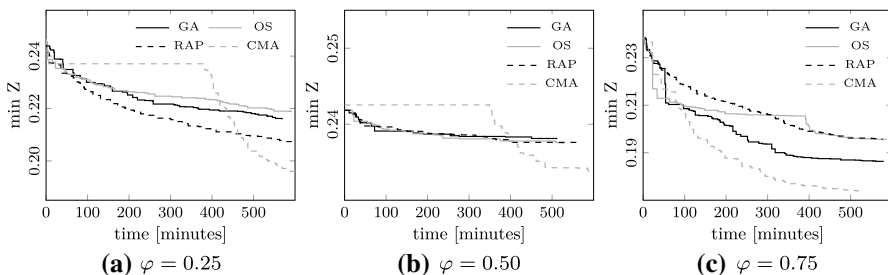


**Fig. 9** Largest instance, best  $Z$  per algorithm over time (cont. values)

Figure 10 depicts the plots for the discrete value solution encoding. At a weight  $\varphi = 0.25$  (Fig. 10a), the CMA-ES needs about 400 minutes to start its decline and about 460 to beat its competitors. Again the RAP-GA performs second best. However, compared with its continuous value counterpart (Fig. 9a), there is a bigger difference between the remaining algorithms. In particular the (standard) GA and OS-GA perform worse than before. However, at  $\varphi = 0.50$  the differences for all but the CMA-ES are very small and even smaller than at a weight  $\varphi = 0.50$  (Fig. 9b) in the continuous value version. The CMA-ES performs similar, starts its descent again at about 350 minutes and beats its competitors at about 400 minutes. A weight  $\varphi = 0.75$  (Fig. 10c) produces greater differences between the algorithms. Again, the CMA-ES performs best and (like back in Fig. 9c) starts its descent very swiftly. The OS-GA did not work well in this setting.

For the continuous factor encoding variant (Fig. 11), at a weight  $\varphi = 0.25$  (Fig. 11a), the CMA-ES starts its descent much quicker than before (at around 150 minutes) and beats the other three algorithms at around 200 minutes. The (standard) GA performed second best. At an equal weight  $\varphi = 0.50$  (Fig. 11b), the CMA-ES again starts its descent sooner, than before at about 150 minutes, and passes its competitors around 160 minutes. At a weight  $\varphi = 0.75$  (Fig. 11c), the CMA-ES has no challenger from the start. This time, the RAP-GA performs worse.

Figure 12 contains the plots for the continuous mixed encoding variant. The CMA-ES does not perform as well as in the previous encoding variants. Only at a weight  $\varphi = 0.75$  does it come close to its competitors. However, when compared



**Fig. 10** Largest instance, best  $Z$  per algorithm over time (discrete values)



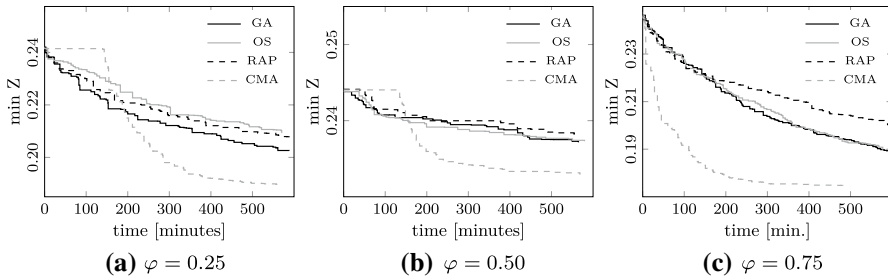


Fig. 11 Best Z per algorithm over time (cont. factors)

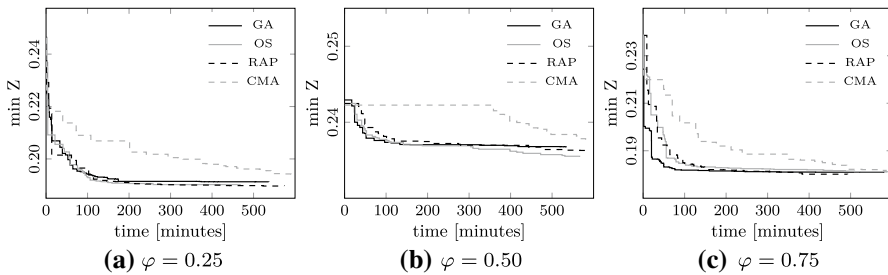


Fig. 12 Best Z per algorithm over time (cont. mixed)

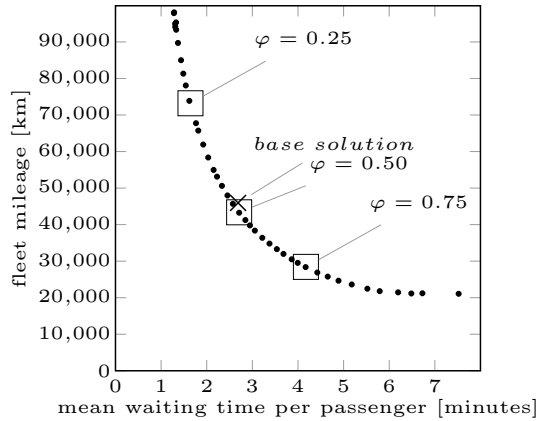
with the other variants, the deviation from the other algorithms is rather small. More important, the descent—especially in Fig. 12c (i.e., at a weight  $\varphi = 0.75$ )—is significantly steeper than that for the other encoding variants. This particular solution encoding thus accelerates the retrieval of good solutions and allows all algorithms to perform better within the first 100 minutes.

#### 7.4 Fleet mileage versus mean waiting time per passenger

We close with an investigation of the trade-off between productive fleet mileage and mean passenger waiting time per passenger. The data points (i.e., coordinates) in Fig. 13 present the best results of an optimization experiment using the CMA-ES on the large instance with 105 decision variables where the weight  $\varphi$  was altered from zero to one in steps of 0.025. The continuous factor solution encoding variant performed best at an equal weight for this particular instance (see Table 7b), so we chose this encoding. The run time was increased to a whole day per optimization run. We again conducted five independent and reproducible runs per variant.

The currently employed headways (solution marked by  $\times$ ) result in 45,943 km fleet mileage and 2.69 minutes mean waiting time per passenger. The best optimized solution (best Z of five) with  $\varphi = 0.475$  leads to 45,689 km and 2.57 minutes, thereby reducing the target measures by 0.55% and 4.46% respectively. Both require a fleet size of 132 vehicles, such that in this solution, both target values improve

**Fig. 13** Fleet mileage versus mean waiting time (best Z; 105 decision variables)



without increasing fleet size. The currently employed solution is almost equally balanced (i.e., close to  $\varphi = 0.50$ ). Table 11 contains some the standard deviations in fleet mileage and mean waiting time per passenger over all five optimization runs per  $\varphi$  near low and high weight values.

At low weight values, the standard deviation correspond negatively with the weight  $\varphi$ , but at high values, they correspond positively. This finding reflects the positions of these solutions near extreme values of  $\varphi$  (i.e., zero and one) being more diverse.

At a weight  $\varphi = 0.25$ , the CMA-ES manages to evaluate 14,920 solutions. This number increases to 21,170 and 24,640 at weights  $\varphi = 0.50$  and  $0.75$ , respectively, due to the aforementioned (Sect. 4.2) negative correlation between weight  $\varphi$  and the

**Table 11** Std. deviations (mean Z; excerpt)

Weight ( $\varphi$ )	Standard deviation	
	Fleet Mileage	Waiting time
0.00	<b>1755</b>	0.03
0.05	<b>1771</b>	0.03
0.10	<b>1600</b>	0.02
0.15	862	0.02
0.20	994	0.02
0.25	621	0.02
...	...	...
0.75	199	0.03
0.80	268	0.04
0.85	296	0.07
0.90	318	<b>0.07</b>
0.95	331	<b>0.13</b>
1.00	208	<b>0.22</b>

number of replications. As mentioned in Sect. 3, the extent to which  $Z$  is exposed to randomness varies.

Practitioners are often interested in multiple solutions from which they then can choose. Furthermore, extreme solutions that would result in 50% more fleet mileage (with similar increases in fleet size and drivers) create real-world implementation issues, especially in the short term. A huge order of additional vehicles requires time and preparations by manufacturers, as does the process for recruiting and training of new drivers on the buyer's side. Table 12 contains some notable key performance indicators—some of which were discussed back in Sect. 3—in the domain near the base solution, which may be of interest to decision makers. The solution close to the base solution at  $\varphi = 0.475$  (in bold) does not require more vehicles. Fleet mileage and size are in a direct, positive relationship, but they conflict with the mean waiting time per passenger. The mean invehicle time and both mean transfer times (one over all passengers, the other one only for the ones who perform transfers) do not vary much. The mean total travel time thus appears driven mainly by the mean waiting time per passenger, which justifies our choice to use it as a service level indicator within the objective function (Eq. 1).

## 8 Conclusion and perspectives

In this paper, we presented a detailed discrete event simulation model (Sect. 4.1) for urban mass rapid transit networks, inspired by a real-world case study (Sect. 5). The model is embedded in a simulation-based optimization framework in which headways provide the decision variables. The bi-objective optimization problem of cost minimization and service level maximization was transformed into a single-objective optimization problem through normalization and weighted sum-based scalarization (Sect. 3). Four different population-based algorithms were applied (Sect. 4.2). In addition to three traditional solution encoding variants, a newly developed,

**Table 12** Selected fleet- and passenger-related performance indicators (best  $Z$ s)

Weight ( $\varphi$ )	Fleet mileage [km]	Fleet size [pc.]	Mean passenger times [minutes] (all)				Mean transfer time [minutes] (transferring passengers only)
			Invehicle	Waiting	Transfer	Travel	
0.350	58,372	180	9.11	2.03	0.72	11.86	2.30
0.375	54,965	162	9.08	2.15	0.71	11.95	2.30
0.400	53,130	158	9.08	2.23	0.71	12.02	2.29
0.425	50,611	147	9.06	2.33	0.71	12.11	2.29
0.450	47,983	146	9.08	2.45	0.72	12.25	2.30
<b>0.475</b>	<b>45,689</b>	<b>132</b>	<b>9.07</b>	<b>2.57</b>	<b>0.72</b>	<b>12.36</b>	<b>2.29</b>
0.500	43,255	126	9.07	2.71	0.71	12.49	2.29
0.525	41,250	124	9.08	2.85	0.71	12.64	2.29
0.550	39,786	117	9.11	2.95	0.72	12.77	2.30
0.575	38,359	113	9.07	3.05	0.71	12.84	2.30
0.600	36,388	105	9.08	3.22	0.72	13.01	2.30

problem-specific one was introduced. The computational evaluation uses 48 test instances based on a real-world subway network. All in all, the CMA-ES performed best in 84% of all test cases (162 of 192; Sect. 7.1), despite partial difficulties with the problem-specific continuous mixed solution encoding variant. The reason likely is that the encoding itself tries to mimic the dependence between decision variables. However, this encoding offers benefits in terms of finding better solutions within a shorter running time (Sect. 7.3). This might be an indication that the decision variables should not be considered completely independently from each other. In larger instances, especially with unequal weights, the gap increases. The equally weighted instances are more difficult to improve, due to the relatively good performance of the base solution. The currently employed headways already offer a good balance between the objectives of fleet mileage and mean waiting time per passenger reduction (Sect. 7.4). However, the optimized solution at a weight  $\varphi = 0.475$  offers both, cost reductions of 0.55% (less fleet mileage) and service quality improvements of 4.46% (lower mean waiting time per passenger), even without changing the fleet size (i.e., 132 vehicles).

In future research efforts, we plan to develop problem-specific crossover operators in order to improve the results. Furthermore, different types of metaheuristics or a Pareto-based solution approach could be additional options to tackle the problem. Another extension direction might consider planned disruptions. Possible future impacts of this study on the Viennese subway system include changes in the lines' respective hourly headways (i.e., new schedules), planning of vehicle acquisition, infrastructure alterations, and disruption management.

**Acknowledgements** Open access funding provided by University of Vienna. The financial support by the Christian Doppler Research Association, the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development, and Wiener Linien GmbH & Co KG (i.e., the Viennese public transportation provider) is gratefully acknowledged. The authors thank the reviewers for their careful reading of this paper and their helpful comments and constructive suggestions.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

- Affenzeller M, Wagner S, Winkler S (2007) Self-adaptive population size adjustment for genetic algorithms. In: Moreno Díaz R, Pichler F, Quesada Arencibia A (eds) Computer aided systems theory—EUROCAST 2007, vol 4739. Springer, Berlin, pp 820–828. [https://doi.org/10.1007/978-3-540-75867-9\\_103](https://doi.org/10.1007/978-3-540-75867-9_103)
- Affenzeller M, Winkler S, Wagner S, Beham A (2009) Genetic algorithms and genetic programming: modern concepts and practical applications. No. 6 in Numerical insights. Chapman & Hall/CRC Press, Boca Raton, oCLC: 836514862
- Agard B, Morency C, Trépanier M (2007) Mining public transport user behaviour from smart card data. Technical report, Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT). <https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2007-42.pdf>
- Amaran S, Sahinidis NV, Sharda B, Bury SJ (2016) Simulation optimization: a review of algorithms and applications. *Ann Oper Res* 240(1):351–380. <https://doi.org/10.1007/s10479-015-2019-x>

- Anderson ML (2014) Subways, strikes, and slowdowns: the impacts of public transit on traffic congestion. *Am Econ Rev* 104(9):2763–2796. <https://doi.org/10.1257/aer.104.9.2763>
- Bauer D, Brändle N, Seer S, Ray M, Kitazawa K (2009) Measurement of pedestrian movements: a comparative study on various existing systems. In: Timmermans H (ed) *Pedestrian behavior: models, data collection and applications*. Bingley, Emerald
- Bovy PHL, Stern E (1990) Route Choice: wayfinding in transport networks, studies in operational regional science, vol 9. Springer, Dordrecht. <https://doi.org/10.1007/978-94-009-0633-4>
- Cacchiani V, Huisman D, Kidd M, Kroon L, Toth P, Veelenturf L, Wagenaar J (2014) An overview of recovery models and algorithms for real-time railway rescheduling. *Transp Res Part B Methodol* 63:15–37. <https://doi.org/10.1016/j.trb.2014.01.009>
- Ceder A (1984) Bus frequency determination using passenger count data. *Transp Res Part A General* 18(5–6):439–453. [https://doi.org/10.1016/0191-2607\(84\)90019-0](https://doi.org/10.1016/0191-2607(84)90019-0)
- Ceder A (2001) Bus timetables with even passenger loads as opposed to even headways. *Transp Res Record J Transp Res Board* 1760:3–9. <https://doi.org/10.3141/1760-01>
- Ceder A, Wilson NH (1986) Bus network design. *Transp Res Part B Methodol* 20(4):331–344. [https://doi.org/10.1016/0191-2615\(86\)90047-0](https://doi.org/10.1016/0191-2615(86)90047-0)
- Chong L, Osorio C (2018) A simulation-based optimization algorithm for dynamic large-scale urban transportation problems. *Transp Sci* 52(3):637–656. <https://doi.org/10.1287/trsc.2016.0717>
- Corman F, D’Ariano A, Marra AD, Pacciarelli D, Samà M (2017) Integrating train scheduling and delay management in real-time railway traffic control. *Transp Res Part E Logist Transp Rev* 105:213–239. <https://doi.org/10.1016/j.tre.2016.04.007>
- Darwin C (1859) *On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life*. John Murray, London
- Dijkstra EW (1959) A note on two problems in connexion with graphs. *Numer Math* 1(1):269–271. <https://doi.org/10.1007/BF01386390>
- Dollevoet T, Huisman D, Kroon L, Schmidt M, Schöbel A (2015) Delay management including capacities of stations. *Transp Sci* 49(2):185–203. <https://doi.org/10.1287/trsc.2013.0506>
- Farahani RZ, Miandoabchi E, Szeto W, Rashidi H (2013) A review of urban transportation network design problems. *Eur J Oper Res* 229(2):281–302. <https://doi.org/10.1016/j.ejor.2013.01.001>
- Frejinger E (2008) *Route choice analysis: data, models, algorithms and applications*. Ph.D. thesis, École polytechnique fédérale de Lausanne, Lausanne. [https://www.researchgate.net/profile/Emma\\_Frejinger/publication/37454429\\_Route\\_choice\\_analysis\\_data\\_models\\_algorithms\\_and\\_applications/links/0046352922b197b0d5000000.pdf](https://www.researchgate.net/profile/Emma_Frejinger/publication/37454429_Route_choice_analysis_data_models_algorithms_and_applications/links/0046352922b197b0d5000000.pdf)
- Friedrich M, Immisch K, Jehlicka P, Otterstätter T, Schlaich J (2010) Generating origin–destination matrices from mobile phone trajectories. *Transp Res Record J Transp Res Board* 2196:93–101. <https://doi.org/10.3141/2196-10>
- Fu MC (2002) Optimization for simulation: theory vs. practice. *INFORMS J Comput* 14(3):192–215. <https://doi.org/10.1287/ijoc.14.3.192.113>
- Furth PG, Wilson NH (1981) Setting frequencies on bus routes: theory and practice. *Transp Res Record* 818:1–7
- Guihaire V, Hao JK (2008) Transit network design and scheduling: a global review. *Transp Res Part A Policy Pract* 42(10):1251–1273. <https://doi.org/10.1016/j.tra.2008.03.011>
- Han AF, Wilson NH (1982) The allocation of buses in heavily utilized networks with overlapping routes. *Transp Res Part B Methodol* 16(3):221–232. [https://doi.org/10.1016/0191-2615\(82\)90025-X](https://doi.org/10.1016/0191-2615(82)90025-X)
- Hanika A (2018) *Zukünftige Bevölkerungsentwicklung Österreichs und der Bundesländer 2017 bis 2080 (Teil 2)*. *Stat Nachr* 3:218–225
- Hansen N, Ostermeier A (2001) Completely derandomized self-adaptation in evolution strategies. *Evolut Comput* 9(2):159–195. <https://doi.org/10.1162/106365601750190398>
- Herbon A, Hadas Y (2015) Determining optimal frequency and vehicle capacity for public transit routes: a generalized newsvendor model. *Transp Res Part B Methodol* 71:85–99. <https://doi.org/10.1016/j.trb.2014.10.007>
- Holland JH (1975) *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, 1st edn. Complex adaptive systems. MIT Press, Cambridge
- Hornby GS (2006) ALPS: the age-layered population structure for reducing the problem of premature convergence. In: *Proceedings of the 8th annual conference on genetic and evolutionary computation*. ACM Press, Seattle, pp 815–822. <https://doi.org/10.1145/1143997.1144142>

- Huang Z, Ren G, Liu H (2013) Optimizing bus frequencies under uncertain demand: case study of the transit network in a developing city. *Math Probl Eng* 2013:1–10. <https://doi.org/10.1155/2013/375084>
- Ibarra-Rojas O, Delgado F, Giesen R, Muñoz J (2015) Planning, operation, and control of bus transport systems: a literature review. *Transp Res Part B Methodol* 77:38–75. <https://doi.org/10.1016/j.trb.2015.03.002>
- IKK (2017) Matchmobile-multimodal trip chains from mobile phones. <https://www.ikk.at/projekt/match-mobile-multimodal-trip-chains-from-mobile-phones/>
- Jackson JR (1963) Jobshop-like queueing systems. *Manag Sci* 10(1):131–142. <https://doi.org/10.1287/mnsc.10.1.131>
- Juan AA, Faulin J, Grasman SE, Rabe M, Figueira G (2015) A review of simheuristics: extending metaheuristics to deal with stochastic combinatorial optimization problems. *Oper Res Perspect* 2:62–72. <https://doi.org/10.1016/j.orp.2015.03.001>
- Kiefer A, Kritzingner S, Doerner KF (2016) Disruption management for the Viennese public transport provider. *Pub Transp* 8(2):161–183. <https://doi.org/10.1007/s12469-016-0123-1>
- Li Y, Xu W, He S (2013) Expected value model for optimizing the multiple bus headways. *Appl Math Comput* 219(11):5849–5861. <https://doi.org/10.1016/j.amc.2012.11.098>
- Liebchen C (2008) The first optimized railway timetable in practice. *Transp Sci* 42(4):420–435. <https://doi.org/10.1287/trsc.1080.0240>
- Mohaymany AS, Amiripour SM (2009) Creating bus timetables under stochastic demand. *Int J Ind Eng Prod Res* 20(3):83–91
- Newell GF (1971) Dispatching policies for a transportation route. *Transp Sci* 5(1):91–105. <https://doi.org/10.1287/trsc.5.1.91>
- Niu H, Zhou X (2013) Optimizing urban rail timetable under time-dependent demand and oversaturated conditions. *Transp Res Part C Emerg Technol* 36:212–230. <https://doi.org/10.1016/j.trc.2013.08.016>
- Osorio C, Bierlaire M (2013) A simulation-based optimization framework for urban transportation problems. *Oper Res* 61(6):1333–1345. <https://doi.org/10.1287/opre.2013.1226>
- Osorio C, Chong L (2015) A computationally efficient simulation-based optimization algorithm for large-scale urban transportation problems. *Transp Sci* 49(3):623–636. <https://doi.org/10.1287/trsc.2014.0550>
- Pelletier MP, Trépanier M, Morency C (2011) Smart card data use in public transit: a literature review. *Transp Res Part C Emerg Technol* 19(4):557–568. <https://doi.org/10.1016/j.trc.2010.12.003>
- Raveau S, Guo Z, Muñoz JC, Wilson NH (2014) A behavioural comparison of route choice on metro networks: time, transfers, crowding, topology and socio-demographics. *Transp Res Part A Policy Pract* 66:185–195. <https://doi.org/10.1016/j.tra.2014.05.010>
- Roth C, Kang SM, Batty M, Barthelemy M (2012) A long-time limit for world subway networks. *J R Soc Interface* 9(75):2540–2550. <https://doi.org/10.1098/rsif.2012.0259>
- Ruano E, Cobos C, Torres-Jimenez J (2017) Transit network frequencies-setting problem solved using a new multi-objective global-best harmony search algorithm and discrete event simulation. In: Pichardo-Lagunas O, Miranda-Jiménez S (eds) *Advances in soft computing*, vol 10062. Springer, Cham, pp 341–352. [https://doi.org/10.1007/978-3-319-62428-0\\_27](https://doi.org/10.1007/978-3-319-62428-0_27)
- Salzborn FJM (1972) Optimum bus scheduling. *Transp Sci* 6(2):137–148. <https://doi.org/10.1287/trsc.6.2.137>
- Schéele S (1980) A supply model for public transit services. *Transp Res Part B Methodol* 14(1–2):133–146. [https://doi.org/10.1016/0191-2615\(80\)90039-9](https://doi.org/10.1016/0191-2615(80)90039-9)
- Schmaranzer D, Braune R, Doerner KF (2016) A discrete event simulation model of the Viennese subway system for decision support and strategic planning. In: Roeder T, Frazier P, Szechtman R, Zhou E, Huschka T, Chick S (eds) *Proceedings of the 2016 winter simulation conference*, Institute of Electrical and Electronics Engineers, Inc., Piscataway, pp 2406–2417. <https://doi.org/10.1109/WSC.2016.7822280>, <https://www.informs-sim.org/wsc16papers/210.pdf>
- Schmaranzer D, Braune R, Doerner KF (2018) Simulation-based headway optimization for a subway network: a performance comparison of population-based algorithms. In: Rabe M, Angel AJ, Mustafee N, Skoogh A, Jain S, Johansson B (eds) *Proceedings of the 2018 winter simulation conference*, Institute of Electrical and Electronics Engineers, Inc., Piscataway, pp 1957–1968. <https://doi.org/10.1109/WSC.2018.8632362>, <https://www.informs-sim.org/wsc18papers/includes/files/164.pdf>
- Shrivastava P, Dhingra SL (2002) Development of coordinated schedules using genetic algorithms. *J Transp Eng* 128(1):89–96. [https://doi.org/10.1061/\(ASCE\)0733-947X\(2002\)128:1\(89\)](https://doi.org/10.1061/(ASCE)0733-947X(2002)128:1(89))

- Shrivastava P, Dhingra SL, Gundaliya PJ (2002) Application of genetic algorithm for scheduling and schedule coordination problems. *J Adv Transp* 36(1):23–41. <https://doi.org/10.1002/atr.5670360103>
- Statistik Austria (2017) Download Bevölkerungsprognosen (ausführliche Tabellen Wien). [https://www.statistik.at/wcm/idc/idcplg?IdcService=GET\\_NATIVE\\_FILE&RevisionSelectionMethod=LatesReleased&dDocName=027327](https://www.statistik.at/wcm/idc/idcplg?IdcService=GET_NATIVE_FILE&RevisionSelectionMethod=LatesReleased&dDocName=027327)
- Sun L, Jin JG, Lee DH, Axhausen KW, Erath A (2014) Demand-driven timetable design for metro services. *Transp Res Part C Emerg Technol* 46:284–299. <https://doi.org/10.1016/j.trc.2014.06.003>
- Takahashi M, Kita H (2001) A crossover operator using independent component analysis for real-coded genetic algorithms. In: Proceedings of the 2001 congress on evolutionary computation, vol 1. IEEE, pp 643–649. <https://doi.org/10.1109/CEC.2001.934452>, <https://ieeexplore.ieee.org/document/934452/>
- United Nations (2018) Download annual percentage of population at mid-year residing in urban areas by region, subregion and country, 1950–2050. [https://esa.un.org/unpd/wup/Download/Files/WUP2018-F21-Proportion\\_Urban\\_Annual.xls](https://esa.un.org/unpd/wup/Download/Files/WUP2018-F21-Proportion_Urban_Annual.xls)
- Vázquez-Abad FJ, Zubieta L (2005) Ghost simulation model for the optimization of an urban subway system. *Discrete Event Dyn Syst* 15(3):207–235. <https://doi.org/10.1007/s10626-005-2865-9>
- VSC (2018) Vienne Scientific Cluster. <https://vsc.ac.at/systems/vsc-3/>
- Wagner S, Kronberger G, Beham A, Kommenda M, Scheibenpflug A, Pitzer E, Vonolfen S, Kofler M, Winkler S, Dorfer V, Affenzeller M (2014) Architecture and design of the heuristiclab optimization environment. In: Klempous R, Nikodem J, Jacak W, Chaczko Z (eds) Advanced methods and applications in computational intelligence, topics in intelligent engineering and informatics, vol 6. Springer, Heidelberg, pp 197–261. [https://doi.org/10.1007/978-3-319-01436-4\\_10](https://doi.org/10.1007/978-3-319-01436-4_10)
- Weidmann U (1994) Der Fahrgastwechsel im öffentlichen Personenverkehr. Dissertation, ETH Zürich. <http://dx.doi.org/10.3929/ethz-a-000948819>
- Wu J, Song R, Wang Y, Chen F, Li S (2015) Modeling the coordinated operation between bus rapid transit and bus. *Math Probl Eng* 2015:1–7. <https://doi.org/10.1155/2015/709389>
- Yu G, Qi X (2004) Disruption management: framework, models and applications. World Scientific, New York, oCLC: 255210733
- Yu B, Yang Z, Sun X, Yao B, Zeng Q, Jeppesen E (2011) Parallel genetic algorithm in bus route headway optimization. *Appl Soft Comput* 11(8):5081–5091. <https://doi.org/10.1016/j.asoc.2011.05.051>
- Zhao F, Zeng X (2006) Optimization of transit network layout and headway with a combined genetic algorithm and simulated annealing method. *Eng Optim* 38(6):701–722. <https://doi.org/10.1080/03052150600608917>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**David Schmaranzer** is a predoctoral researcher at the Christian Doppler Laboratory for Efficient Intermodal Transport Operations and Ph.D. student at the Department of Business Decisions and Analytics (University of Vienna). He holds master degrees in operations and in supply chain management from Upper Austria University of Applied Sciences Steyr. His research interests are simulation and combinatorial optimization.

**Roland Braune** is a postdoctoral researcher at the Department of Business Decisions and Analytics (University of Vienna). He holds a Ph.D. in computer science from Johannes Kepler University Linz. His research interests include combinatorial optimization problems, different kinds of scheduling problems, branch and bound, problem-specific heuristics, constraint programming and metaheuristics.

**Karl F. Doerner** is a full professor at the Department of Business Decisions and Analytics (University of Vienna). He is head of the Christian Doppler Laboratory for Efficient Intermodal Transport Operations and core member of research platform Data Science @ Uni Vienna. He holds a Ph.D. in business informatics and the *venia legendi* in business administration. His main research interests focus on the development of intelligent search and optimization techniques based on meta- and matheuristics primarily for decision problems in logistics and transportation.