A matheuristic for transfer synchronization through integrated timetabling and vehicle scheduling

Fonseca, João Paiva; van der Hurk, Evelien; Roberti, Roberto; Larsen, Allan

Published in:
Transportation Research Part B: Methodological

Link to article, DOI:
10.1016/j.trb.2018.01.012

Publication date:
2018

Document Version
Peer reviewed version

Citation (APA):
Fonseca, J. P., van der Hurk, E., Roberti, R., & Larsen, A. (2018). A matheuristic for transfer synchronization through integrated timetabling and vehicle scheduling. *Transportation Research Part B: Methodological*, *109*, 128-149. https://doi.org/10.1016/j.trb.2018.01.012

A matheuristic for transfer synchronization through integrated timetabling and vehicle scheduling

João Paiva Fonseca[a], Evelien van der Hurk[a], Roberto Roberti[b], Allan Larsen[a]

[a] *Management Science Division, DTU Management Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark*

[b] *Department of Information, Logistics and Innovation, VU Amsterdam, 1081 HV Amsterdam, Netherlands*

Abstract

Long transfer times often add unnecessary inconvenience to journeys in public transport systems. Synchronizing relevant arrival and departure times through small timetable modifications could reduce excess transfer times, but may also directly affect the operational costs, as the timetable defines the set of feasible vehicle schedules. Therefore better results in terms of passenger service, operational costs, or both, could be obtained by solving these problems simultaneously.

This paper addresses the tactical level of the Integrated Timetabling and Vehicle Scheduling Problem as a bi-objective mixed integer programming problem that minimizes transfer costs and operational costs. Given an initial non-cyclical timetable, and time-dependent service times and passenger demand, the weighted sum of transfer time cost and operational costs is minimized by allowing modifications to the timetable that respect a set of headway constraints. Timetable modifications consist of shifts in departure time and addition of dwell time at intermediate stops with transfer opportunities.

A matheuristic is proposed that iteratively solves the mathematical formulation of the Integrated Timetabling and Vehicle Scheduling Problem allowing timetable modifications for a subset of timetabled trips only, while solving the full vehicle scheduling problem. We compare different selection strategies for defining the sub-problems. Results for a realistic case study of the Greater Copenhagen area indicate that the matheuristic is able to find better feasible solutions faster than a commercial solver and that allowing the addition of dwell time creates a larger potential for reducing transfer costs.

Keywords: Public Transport, Bus Timetabling, Vehicle Scheduling, Mixed Integer Linear Programming, Matheuristic

1. Introduction

Transfers add substantial amounts of travel time to journeys in large public transport systems. Reduced transfer times resulting from a better synchronization of trips in timetables could increase ridership of public transport and thereby potentially diminish congestion (Ibarra-Rojas and Rios-Solis, 2012). A higher mode share for public transportation furthermore aids to reduce rising pollution levels. Therefore, the integration of timetabling and

Email addresses: jfpf@dtu.dk (João Paiva Fonseca), evdh@dtu.dk (Evelien van der Hurk), r.roberti@vu.nl (Roberto Roberti), alar@dtu.dk (Allan Larsen)

vehicle scheduling is important because it improves passenger service at limited operating costs (Guihaire and Hao, 2010). This paper addresses this issue in the context of tactical timetable and vehicle schedule design.

Timetables and vehicle schedules are closely related problems, but are traditionally solved sequentially (Desaulniers and Hickman, 2007). Indeed, a small change in the timetable could render an initial vehicle schedule infeasible, *or* could create options for less costly vehicle schedules. Sequentially optimizing these plans can therefore result in suboptimal solutions. That there is a benefit in integrating timetabling and vehicle scheduling was already demonstrated by e.g. Ceder (2001),Van den Heuvel et al. (2008), Petersen et al. (2013), Ibarra-Rojas et al. (2014) and Laporte et al. (2017), who report savings of up to 20% in transfer waiting times while keeping operational costs at a similar level.

This paper addresses the tactical level of the *Integrated Timetabling and Vehicle Scheduling Problem* (IT-VSP) as a bi-objective mixed integer programming problem that minimizes transfer costs and operational costs. Given an initial non-cyclical timetable, and time-dependent service times and fixed passenger demand per transfer, the weighted sum of transfer time cost and operational costs is minimized by allowing modifications to the timetable that respect a set of headway constraints. Timetable modifications consist of shifts in departure time and addition of dwell time at intermediate stops with transfer opportunities. Novelty of the current work lies in the far wider set of allowed timetable modifications in the IT-VSP, the detailed representation of vehicle schedules, and the new matheuristic that for the first time allows to compare results to a lower bound on the problem.

The contributions of this paper are threefold: (i) we present a mathematical formulation for the IT-VSP that allows for a far wider set of timetable modifications by allowing both a change in departure time as well as increases in dwell time under headway constraints; (ii) we propose a matheuristic approach that generates good quality solutions for real size instances faster than a general purpose commercial solver; and (iii) we apply our methodology to a real case study for the express bus network in the Greater Copenhagen area. Results of the case study indicate that the integrated planning of timetables and vehicle schedules can reduce both excess transfer times for passengers and the operational costs. The solutions of the matheuristic in one hour of computation time are substantially better than the solutions of a general purpose solver after seven days of computation time.

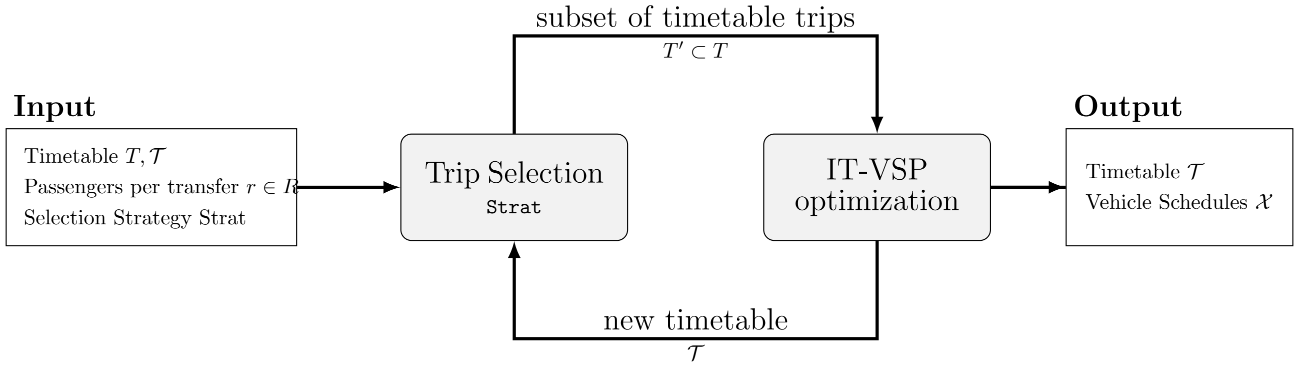

Figure 1: Flow diagram of the matheuristic approach

The matheuristic, depicted in Figure 1, selects and solves a sub-problem of the IT-VSP in each iteration. The input consists of the set of timetabled trips, an initial timetable, a fixed passenger demand per transfer opportunity, and a selection strategy for defining the subproblem. We propose and compare four selection strategies. The matheuristic consists of the two blocks in Figure 1, executed iteratively. First a subset of timetable trips is selected according to the selection strategy. Next the IT-VSP mixed integer programming formulation is solved, rescheduling timetables for selected trips only while simultaneously optimizing the vehicle schedules. Specific features of the model are the dynamic assignment of transferring passengers to transfer-to trips, the allowance of non-cyclic timetables, pre-defined changes in on- and off-peak travel times of vehicles, and the creation of detailed vehicle schedules for the planning horizon (e.g. 24 hours). The output defines the new, best known timetable, as well as vehicle schedules that cover that timetable. The iterations stop when either a maximum time or a maximum number of iterations have been reached.

The remainder of this paper consists of a problem description (Section 2), a literature review (Section 3), a formal problem definition including the MIP formulation (Section 4), the matheuristic (Section 5), and a case study and discussion of results (Sections 6 and 7), as well as conclusions and suggestions for future research (Section 8).

2. The integration of timetabling and vehicle scheduling

Given a set of bus lines and desired frequencies, the *Transit Network Timetabling* (TNT) problem defines the departure and arrival times for each stop visited by each trip. The general TNT problem aims at maximizing passenger service, and may consider schedule synchronization and transfer times. The *Multiple Depot Vehicle Scheduling Problem* (MDVSP) has the goal of operating a set of timetabled trips using vehicles from a set of depots at a minimum cost. When only one depot is available, the problem is referred to as the *Single Depot Vehicle Scheduling Problem* (SDVSP), which is solvable in polynomial time. Inputs to the problem are the number of vehicles available at each depot, a set of timetabled trips to be serviced (which is the output of the TNT), and a distance matrix between all terminal stops and depots. Vehicles must start and end at the same depot.

The IT-VSP applies modifications to a provided timetable to minimize a weighted sum of passenger costs and operational costs resulting from the vehicle schedules. Passenger demand is known and fixed, and defined as a number of passengers that wish to transfer from a specific trip to another line at a specific stop. As an example, Figure 2 depicts a transfer opportunity between two lines, 300S and 400S, at a Lynby station. Passengers are assumed to transfer from an arriving trip of a feeder line (e.g. 300S) to the first available trip of their desired line (e.g. 400S). Operational costs are defined as the costs for vehicles waiting at stops (when stretching), vehicles performing non-service trips (dead heading), and fixed costs per vehicle schedule.

A transfer opportunity exists at the crossing of two bus or train lines, such as the crossing of bus lines 300S and 400S at Lyngby station in Figure 2. The transfer depicted in Figure 3 requires a minimum time of 4 minutes to disembark the 300S vehicle, walk to the stop of the 400S line and embark the 400S vehicle at Lyngby Station. In the current timetable, trip i of the 300S line arrives at 9:30 leaving a 10 minute transfer time to the next departing 400S line trip j_3 at 9:40, thus resulting in 6 minutes of excess transfer time.

Timetable modifications that postpone trip j_2's departure time from 9:30 to 9:34 would remove all excess transfer time for the 300S trip i transfer to line 400S. However, such a change could also influence the departure times of other trips on this line. Figure 4 depicts the allowed timetable modifications. The scheduled time of each trip is allowed to vary within half the scheduled headway time (interval of time between two consecutive trips in the same line) to the next trip, thus allowing trip j_2's departure time to be within 9:26 and 9:35. A shift changes the departure of a trip from its first stop. A stretch adds dwell time at any of the intermediate stops of a trip, as represented by the grey nodes and resulting arcs in Figure 4. The addition of stretches could represent the distribution of buffer time over a trip, in such a way that it enables better transfer opportunities. The enlargement of arrivals and departures at Lyngby station for line 400S in Figure 4 illustrates that timetable modifications are limited to respect a minimum headway h_u^- and maximum headway h_u^+ from the predecessor trip. The change in trip j_2 to 9:34 would thus require a change in the departure times of trips j_1 and j_3 to ensure even headways.

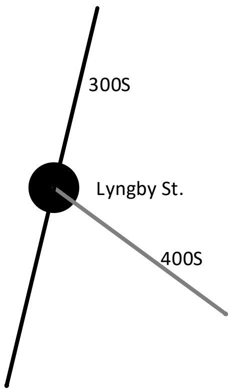

Figure 2: Geographic representation of two bus lines, 300S and 400S, and a station (Lyngby St.), where passengers can transfer

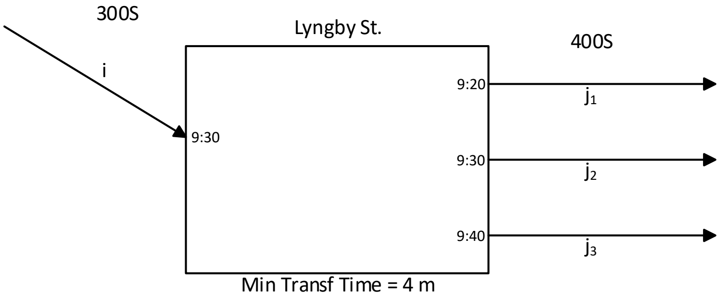

Figure 3: Example of passenger transfer from trip i of line 300S to line 400S at Lyngby St. given a minimum transfer time of 4 minutes, passengers can embark into trip j_3 only after waiting 6 minutes

Alternatively, the required shift in the departure time of trip j_2 could be limited if the trip i would be changed to arrive earlier. Such earlier departure could only result from a shift, a change in departure times from the first stop in a trip, while a later departure can result from both shifts and/or stretches adding additional dwell time.

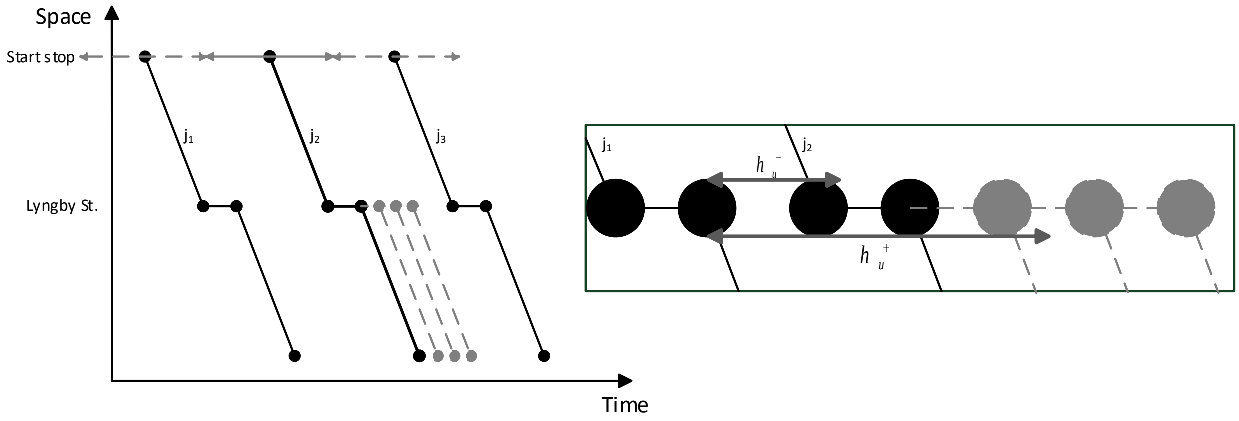

Figure 4: Representation of allowed timetable modifications and headway bounds for trip j_2 at Lyngby station

Timetable modifications may influence which groups of trips can be serviced together by one vehicle, as shifts and stretches applied to trips can alter the required arrival time of a vehicle at the start terminal, or change the arrival time of the selected vehicle at the end terminal of the trip. Trips in Figure 5 are depicted as a short arc between a start node and an arrival node (i.e., the start and end terminals of the trip). The grey bars represent the possible shifts forward and backward in time of the trip. The dashed arcs in Figure 5 represent a feasible vehicle schedule that starting from depot k in $O(k)$ serves trip $i-1$, deadheads from $i-1$ to serve trip j_1, deadheads from j_1 to serve trip j_3 and finally returns to the depot in $D(k)$. An earlier departure of trip j_1 would make the depicted vehicle schedule infeasible, while alternative shifts may allow to serve more trips with the same vehicle.

The MIP and the matheuristic we propose in Sections 4 and 5 aim at simultaneously optimizing the timetabling problem and the vehicle scheduling problem by allowing changes in the input timetables in the form of shifts and stretches illustrated in Figures 2-5.

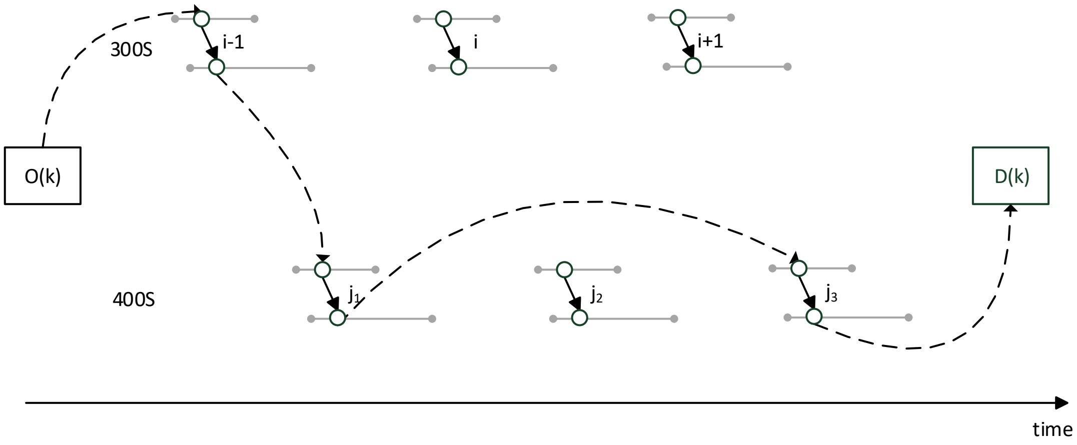

Figure 5: Representation of the vehicle scheduling part of the problem

3. Literature review

In this section, we review previous work on the domains of the MDVSP, public transport timetabling, and integration of timetabling and vehicle scheduling. We refer to Guihaire and Hao (2008a) for a review on transit network design and scheduling, who classify and describe over 60 approaches dealing with design, frequency setting, timetabling and combinations of these problems. For a review on the topics of planning, operation and control of bus transport systems we refer to Ibarra-Rojas et al. (2015), who present an extensive literature review with two chapters devoted to timetabling and vehicle scheduling problems.

3.1. Multiple depot vehicle scheduling

The IT-VSP is an extension of the MDVSP, which is a classical problem within operations research and more specifically in the domain of transport optimization problems. In their book chapter on Time Constrained Routing and Scheduling, Desrosiers et al. (1995) present a detailed description of the problem and provide a review on the literature existent at the time. For a vast survey on the MDVSP, we refer the reader to Desaulniers and Hickman (2007), where also some extensions to the MDVSP are discussed. Real-life MDVSP instances of up to 7000 trips can be solved to optimality using column generation by Kliewer et al. (2006). The instances in Kliewer et al. (2006) possess a specific structure that contributes to finding optimal solutions in reasonable time for these large instances. Hadjar et al. (2006) is able to solve randomly generated instances developed by Carpaneto et al. (1989) with up to 800 trips only, with an exact branch-and-bound approach. Due to its complexity, the MDVSP is usually solved using heuristics or metaheuristics. Pepin et al. (2009) compare the performance of five different heuristics to solve instances of the problem with 500, 1000, and 1500 trips, both with 4 and 8 depots. Their computational experiments indicate that when enough computational time is available truncated column generation is the best performing approach, with upper bounds on the average with 0.17% and 0.837% from the optimal solution, and using up to one hour of computing time. However, if the goal is to obtain good quality solutions in fast computational times then large neighbourhood search performs best.

3.2. Transit network timetabling

The TNT determines arrival and departure times at stops visited, meeting a given frequency, and satisfying demand. The TNT problem is addressed in the literature using different objectives, for example minimization of excess transfer times (transfer waiting times), maximization of synchronization, or multi-objective approaches. In this section, we review previous research work on the TNT, and group the contributions according to similarities in the objectives considered.

Klemt and Stemme (1988) are the first authors to use minimization of excess transfer times as an objective to the TNT, and proposed a quadratic semi-assignment formulation to model the problem. The authors describe a constructive heuristic to schedule trips one by one and considering transfer synchronization. Domschke (1989) propose a branch-and-bound, local search, and simulated annealing algorithms that outperform Klemt and Stemme (1988) in a simplified example for West Berlin's subway. Bookbinder and Desilets (1992) minimize transfer costs (costs associated with excess transfer times) considering stochastic travel times and constant headways for each line. Their decision variables are defined as the departures of the first trip in each line. They employ a combination of simulation procedure and the mathematical formulation of Klemt and Stemme (1988). Small example networks are used for the experimental results, and results are compared for optimizing a single transfer node and multiple transfer nodes. Daduna and Voß (1995) minimize excess transfer times and modify the quadratic semi-assignment formulation to restrict transfer opportunities to be within a maximum waiting time. The authors use regret heuristics and unidirectional improvement procedures to compute initial solutions, which are then improved using simulated annealing and tabu search procedures. The different approaches are tested in a series of examples of different sizes, ranging from 14 to 27 lines and from 15 to 38 transfer nodes. Schröder and Solchenbach (2006) start from an original timetable and allow shifts in the start times of trips to optimize the quality of transfers; they also introduce a new way to assess transfer quality dependent on the amount of time available for transferring and the perceived quality by the users. The TNT is modelled as a quadratic semi-assignment problem, and they solve a linearization of the problem in `CPLEX` for a small real-life case study. Their results indicate improvements between 0.5% and 5% in comparison to the original timetables, depending on which shifts are allowed and how many nodes are synchronized. Wong et al. (2008) also adjust an original timetable to minimize excess transfer times, but formulate the TNT as a MIP and solve it using Lagrangian heuristics. They allow a wide range of timetable modifications in the form of headway variation, dispatch times, station dwell times and train run times. Their heuristic showed improvements in the solutions obtained for a real case study with 4 lines and 16 transfer stops, when comparing with fixed headways and trip times. Shafahi and Khani (2010) minimize the waiting time at transfer stations and allow the addition of dwell time to improve the transfers. The effect of the added dwell time on through passengers is considered. A large case study is solved using a genetic algorithm approach and reductions of 11.5% of transfer waiting times are reported. Wu et al. (2015) present a timetabling model that minimizes the total waiting time costs for three classes of passengers: transferring, boarding, and through passengers. They consider stochastic travel times and allow the addition of slack time to benefit the passenger transfer feasibility. The authors report that the model is especially effective if the ratio of through-passengers to transfer passengers is below a certain threshold.

Ceder et al. (2001), Liu et al. (2007), and Ibarra-Rojas and Rios-Solis (2012) use maximization of synchronization. The three studies address similar problems where timetables are constructed using minimum and maximum values for headways, and allow solutions with headway variations. Ceder et al. (2001) formulate the TNT as a MIP and develop a heuristic algorithm so solve small examples with up to 4 transfer nodes. Synchronization is defined as the number of simultaneous bus arrivals at connection nodes of the network. Their solution method is able to find 240 simultaneous bus arrivals for a real-life example with 3 transfer nodes and 14 bus lines, and with approximately 3 hours of operation. Liu et al. (2007) redefine synchronization as a coefficient relating the number of lines with synchronized arrivals at a transfer stop with the total number of lines visiting that stop. They develop a nested

Tabu Search procedure to generate feasible solutions to a small example with 8 lines and 3 transfer stops, but computational times are not reported. Ibarra-Rojas and Rios-Solis (2012) prove that the TNT is NP-hard and create a pre-processing stage that eliminates variables and constraints, improving the tractability of their MIP model. Synchronization is defined as the arrival of two trips of different lines within a certain time window, at a specific stop. The authors propose an *Iterated Local Search* (ILS) procedure to solve large test instances of the problem, with up to 200 lines and 40 transfer points, and compare it with a branch-and-bound procedure. The ILS obtains a gap of 15.55% from the lower bound in under one minute of computational time, while the branch-and-bound only gets to 22.64% in two hours of computational time.

Kwan and Chang (2008), Hassold and Ceder (2012), Liu and Ceder (2016), and Wu et al. (2016) consider the TNT as a multi-objective problem. Kwan and Chang (2008) start from an original timetable and allow changes in frequency, dwell time, layover time, and run time to minimize the cost of transfers and costs caused by deviations from the initial timetable. They use an upper bound on the number of vehicles available to limit the changes made to the timetables. The authors implement a genetic algorithm, a multi objective evolutionary algorithm, and a local search procedure to solve the problem. They report computational experiments on an example with 6 lines and 5 transfer points. Hassold and Ceder (2012) compare their results with the current timetable and minimize empty seat penalties and expected passenger waiting times. The authors formulate the TNT as a network flow bi-objective problem and consider different vehicle types. They use a multi-objective label-correcting algorithm to solve the problem and computational experiments for a real life case show savings of up to 43% in passenger waiting times, associated with acceptable passenger loads on all vehicles. Liu and Ceder (2016) present a two-objective MIP that minimizes the expected total passenger waiting time and variation in vehicle occupancy. The authors consider fluctuating passenger demand and multiple vehicle types. A decomposition method is used to solve an example and a real life network. Comparing with the current timetables, their approach obtains solutions that reduce total passenger waiting times by approximately 70%, while also reducing variation in vehicle occupancy by approximately 60%. Wu et al. (2016) study a multi-objective re-synchronizing problem for bus timetables, characterized by headway-sensitive passenger demand, uneven headways, service regularity, and flexible synchronization. The objectives considered are the maximization of the number of passengers benefited by smooth transfers and the minimization of the deviation from the existing initial timetable. They use a genetic algorithm to solve the problem and report that high-quality non-dominated solutions are obtained within reasonable CPU time.

For cyclical timetabling problems, a variety of authors apply methodologies using the *Periodic Event Scheduling Problem* (PESP), introduced by Serafini and Ukovich (1989). The idea is to schedule the events for a cycle, which is then repeated throughout the day. PESP-based approaches are not suitable for solving the IT-VSP in our case, since they impose a periodicity constraint which we want to break apart from. We refer the reader to Nachtigall (1999) for a strong formulation of the PESP applied to railway timetabling. Also in the railway timetabling field, Liebchen and Möhring (2007) extend the PESP to include important decisions of network planning, line planning, and vehicle scheduling into the task of periodic timetabling. A recent state-of-the-art review on cyclic railway timetabling can be found in Kümmling et al. (2015) and other PESP applications to solve the cyclic railway timetabling problem can be found in Liebchen and Möhring (2002), Peeters (2003), or Kroon et al. (2007).

3.3. Integrated timetabling and vehicle scheduling

To our best knowledge, Ceder (2001) is the first to study the integration of timetabling and the SDVSP. The author develop a 4-step sequential approach with a single feedback loop that determines a timetable and vehicle schedules, obtaining good solutions for both the operator and passengers. The approach is tested in an example with 3 hours of operations and two lines, with a total of 22 trips. Chakroborty et al. (2001) are the first to include in the TNT the decision of an “optimal fleet size”, which is the number of buses available for each line. They propose a genetic algorithm to solve the problem and present results for a 3-line test instance and total scheduling period of 4 hours. Liu and Shen (2007) use bi-level programming to integrate the TNT formulation of Liu et al. (2007) with the MDVSP, where the upper level minimizes the number of vehicles and deadhead costs and the lower level minimizes the excess transfer time of passengers at intermediate stops. They develop a bi-level nesting tabu search algorithm to solve the problem and present results for an example network with four lines and 3 connection stops, with 3 hours of operation, where the algorithm runs in under 3 seconds of computational time.

Van den Heuvel et al. (2008) integrate the TNT and the MDVSP with the objective of minimize operational costs, and do not include passenger transfer costs in their model. The authors allow shifts in trip starting times. A Tabu Search algorithm is presented, where timetables are first modified and then the MDVSP problem is optimized. The proposed approach is applied in a real case study with up to 49 lines and 1862 trips, and indicate operational cost reductions of up to about 8% when compared to the original timetable. Guihaire and Hao (2008b) include a weighted objective function considering the number of vehicles, number and quality of transfers, and headway evenness. The problem is solved using an ILS procedure, where at each iteration trips are shifted and the VSP problem is solved. The performance of the solution approach is analyzed for an example with 318 trips, and results show that the number of vehicles is reduced by up to 26%, while the number of feasible transfers increases by up to 44%. Guihaire and Hao (2010) propose a Tabu Search approach that adjusts the timetables by shifts in departure and arrival times, after the vehicle and driver scheduling is solved, with the purpose of providing better transfer opportunities. Without changing either the vehicle or the driver schedules, the operational costs remain constant.

Ceder (2011) creates timetables with even headways and balanced vehicle occupancy, considering multiple vehicle types. The problem is formulated as a cost-flow network problem with NP-hard complexity level, and a heuristic is developed for solving it. The author demonstrates the application of the algorithm in an example with 8 trips and three terminals. Petersen et al. (2013) integrate the MDVSP with timetable modifications in the form of a set of shifts, limiting this set to a pre-defined maximum size, and assume fixed passenger demand. They propose a large neighbourhood search metaheuristic with the objective to optimize a weighted sum of passenger service and operational costs. Results for a case study in the Copenhagen area indicate a decrease in excess transfer time of up to 20%, using the same number of vehicles and a small increase in deadhead cost. Ibarra-Rojas et al. (2014) propose a bi-objective problem that solves the SDVSP and the timetable synchronization problem, assuming passenger demand is fixed. They limit shifts within given time windows around the departure time in a base timetable. With the objectives of minimizing fleet size and maximizing the number of passengers benefited by synchronized transfers, a ϵ-constraint method is implemented to obtain Pareto optimal solutions. Results on case study instances with up to 50 lines, up to 5 transfer nodes, and 4 hours of operation show that in some instances increasing the number of vehicles by one could improve considerably the passenger transfers. Recently, Laporte et al. (2017) integrate timetabling and vehicle scheduling with special attention to route choice. The timetables are designed taking into account operational costs, expressed as number of vehicles per line and restricted to a budget. The authors use a ϵ-constraint solution approach to obtain the exact Pareto front of solutions. Liu and Ceder (2017) also extend the integration of timetabling and vehicle scheduling to include passenger assignment. They present a bi-objective, bi-level IP formulation that optimizes fleet size and user travel and waiting times. An initial vehicle scheduling is given as input. Their integrated vehicle scheduling component does not allow vehicles to deadhead from one trip to another, which greatly reduces the complexity of vehicle scheduling, but would lead to unreasonably costly vehicle schedules for dense networks, like the Copenhagen Network, considered in this paper. Timetabling allows for shifts in departure time. They use a deficit function based sequential search to solve small examples with up to 4 unidirectional lines, 4 transfer stops, and one hour of operations.

Our main contribution in comparison to the related papers of Petersen et al. (2013), Ibarra-Rojas et al. (2014), and Laporte et al. (2017), is that we allow a far wider set of timetable modifications in the form of newly defined stretches, and further extend the set of shifts by removing constraints on the set size. Moreover, we are able to quantify the matheuristic solutions for the IT-VSP in relation to the best feasible and best lower bounds derived from solving the new formulation directly using a general purpose solver for 24 hours. Our results for the case study indicate that the wider set of timetable modifications result in less excess transfer time, while maintaining the same level of operational costs.

4. A mathematical model for the IT-VSP

In this section, we formally define the IT-VSP and formulate it as a mixed integer linear program. In our formulation the assumptions are:

- The number of passengers wishing to transfer at a stop is fixed;
- Passengers transfer to the earliest trip departing after their arrival time plus a minimum time required for transferring;
- The minimum transfer time is the same for all passengers, and may depend on the transfer stop, the feeder line, and the receiving transfer line;
- The travel time between two stops is deterministic and may depend on the time of day;
- Passenger demands are fixed and given, described as transfer opportunities as explained in the subsection "Passenger transfers". The model assumes that all transfer opportunities passed as input have a feasible transfer in both the original and final timetables;
- The minimum required transfer time at a transfer node is assumed constant and independent of the individual passenger.

4.1. Lines and trips

Let S be the set of all stops. We define a direct line $l \in L$ as a sequence of stops visited by a vehicle, with L being the set of all directed lines. Let a timetable be defined by a set $T = \{1, \dots, n\}$ of all timetabled trips. Each trip $i \in T$ is defined as having an id, a directed line l_i, a total minimum travel time t_i, and a set of visited stops $S_i \subseteq S$. Notice that, as the travel time is specified for each trip, then two trips $i, j \in T$ in the same line can have different travel times t_i, t_j, which can depend, for example, on the time of the day the trip is scheduled. We define $st_i \in S_i$ as the start terminal, $et_i \in S_i$ as the end terminal, and $J_i \subseteq S_i$ as the set of all intermediate stops visited by trip $i \in T$, i.e., $J_i = S_i \setminus \{st_i, et_i\}$. For each directed line $l \in L$, we define $T_l \subseteq T$ as the subset of all trips in the directed line l - notice that $T = \bigcup_{l \in L} T_l$ and $T_{l'} \cap T_{l''} = \emptyset$ for all $l', l'' \in L, l' \neq l''$. Furthermore, we define the set T^1 as the set of all trips which are the first in their directed line.

4.2. Timetable modifications

We define minimum and maximum headways, h^-_{is} and h^+_{is} respectively, in relation to the timetable's headways, for each trip $i \in T$ at each stop $s \in J_i \cup \{st_i\}$. The departure time from $st_i \in S_i$ of any trip $i \in T$ can be modified by a shift within an interval $\{d^-_{i,st_i}, d^+_{i,st_i}\}$ defined in relation to its departure time in the original timetable. A dwell time extension is allowed for all intermediate stops of trip i. Let w^-_{is} be the dwell time in the original timetable of a trip $i \in T$ at stop $s \in J_i$, and w^+_{is} the maximum allowed dwell time at the same stop. An upper limit w is imposed on the total added dwell time to all stops of any trip of the set T. For each trip $i \in T$, all timetable modifications define earliest and latest arrival times $\{a^-_{is}, a^+_{is}\}$ at all stops $s \in J_i \cup \{et_i\}$, and earliest and latest departure times $\{d^-_{is}, d^+_{is}\}$ from all stops $s \in J_i \cup \{st_i\}$.

4.3. Passenger transfers

Let R be the set of all transfer opportunities, where a transfer opportunity $r \in R$, defined by a triplet (i, l, s), represents a transfer request from passengers disembarking trip $i \in T$ at stop $s \in J_i \cup \{et_i\}$ with the intent of embarking a trip $j \in T_l$ of line $l \in L$ such that $l \neq l_i$ and $s \in J_j \cup \{st_j\}$. Let f_r be the number of passengers requesting transfer $r \in R$. Passengers are assumed to transfer to the earliest feasible trip $j \in T_l$. Let e_r be the minimum transfer time for transfer $r \in R$. For a transfer $r = (i, l, s) \in R$, transferring to trip $j \in T_l$ is feasible when e_r is not greater than the difference between the departure time of trip j from stop s and the arrival time of trip i at stop s. All transfer opportunities in R are feasible in any feasible timetable. That is, for all $r = (i, l, s) \in R$ there is at least one trip $j \in T_l$ such that a transfer from trip i to j at stop s is feasible. For example, if trip j_3 of Figure 3 did not exist, transfer opportunity $(i, 400S, LyngbySt.)$ would not be feasible in the original timetable and thus not be part of R.

4.4. Compatible trips

The minimum and maximum turnaround times are denoted by $\{q^-, q^+\}$ respectively, and consist of a buffer time that guarantees arriving and enough time for physically turnaround the vehicle. As extending dwell times in the form of stretches redistributes buffer time, the minimum turnaround time can be reduced by the amount of additional dwell time added to a trip. Two trips $i, j \in T$ are compatible in a given timetable if three conditions hold. First, the distance between et_i and st_j, $Dist(et_i, st_j)$, has to be smaller than the maximum deadhead distance u . Second, the earliest arrival time of trip i to its end terminal (a^-_{i,et_i}) plus the minimum turnaround time (q^-) plus the driving time between et_i and st_j, denoted as b_{ij}, has to be smaller or equal to the latest departure time of trip j from its start terminal (d^+_{j,st_j}). And third, the latest arrival time of trip i to its end terminal (a^+_{i,et_i}) plus the maximum turnaround time (q^+) plus b_{ij} has to be greater or equal to the earliest departure time of trip j from its start terminal (d^-_{j,st_j}).

4.5. Vehicle scheduling

Let K denote the set of depots, with each depot $k \in K$ housing v_k vehicles. Each vehicle used in a feasible solution covers a sequence of compatible trips and must return to the depot from which it departed. Each depot $k \in K$ is associated with a graph $G_k = (V_k, A_k)$. The set of nodes V_k contains a node for each trip $i \in T$, as well as for depot $k \in K$ which is denoted $n + k$, thus $V_k = T \cup \{n + k\}$. The set of arcs A_k defines the deadhead trips $I = \{(i, j) | i, j \in T : i \neq j, Dist(et_i, st_j) \leq u, a^-_{i,et_i} + q^- + b_{ij} \leq d^+_{j,st_j}, a^+_{i,et_i} + q^+ + b_{ij} \geq d^-_{j,st_j}\}$, the pull-out trips $\{n + k\} \times T$ and the pull-in trips $T \times \{n + k\}$. A deadhead trip exists for any set of pairwise compatible trip nodes in V_k. Thus, A_k is defined as $A_k = I \cup (\{n + k\} \times T) \cup (T \times \{n + k\})$.

The movement of vehicles is defined using triplets (i, j, k) representing a vehicle from depot $k \in K$ covering the pair of trips $(i, j) \in A_k$. Let $Q = Q^D \cup Q^O \cup Q^H$ be the set of all compatible triplets (i, j, k), where Q^D is the set of all deadhead triplets $Q^D = \{(i, j, k) : k \in K, (i, j) \in I\}$, Q^O is the set of all pull-out triplets $Q^O = \{(n+k, j, k) : k \in K, j \in T\}$, and Q^H is the set of all pull-in triplets $Q^H = \{(i, n+k, k) : i \in T, k \in K\}$. Let us also define $T(Q)$ as the set of all pairs of trips $i, j \in T$ for which a triplet involving i and j exists, $T(Q) = \{(i, j) | i, j \in T : \exists (i, j, k) \in Q\}$.

4.6. Passenger and operating costs

The IT-VSP aims at minimizing a weighted sum of operating costs (defined as vehicle driving costs, fixed costs per schedule, and additional dwell time costs) and passenger costs (defined as excess transfer time costs and travel cost increase due to additional dwell times).

Vehicle driving costs and fixed costs for schedule creation are captured by costs c_{ijk} associated with each triplet $(i, j, k) \in Q$. The cost c_{ijk} of triplet $(i, j, k) \in Q$ is equal to the deadhead time b_{ij} multiplied by a driving cost per time unit. The costs for creating new schedules are included in the pull-out trips: if $(i, j, k) \in Q^O$, so the arc represents a vehicle leaving the depot, c_{ijk} includes a fixed cost for creating a new schedule in addition to the costs for deadheading to the service trip. The costs for creating a new schedule correspond to the fixed cost for using a vehicle. The operating costs associated with additional dwell time are captured by costs $c_i^{DW_O}$, affected to each minute of additional dwell time in trip $i \in T$.

Passenger costs are defined as the sum of excess transfer time per passenger and the increase in in-vehicle travel time for on-board passenger due to additional dwell times. In the model, the excess transfer time per transfer opportunity is calculated exactly from the timetable adjustments, and is penalized by a factor c^{TR} per passenger. The addition of dwell time to a trip is penalized by $c_i^{DW_P}$ per minute of additional dwell time. This cost reflects the travel cost increase from one minute of additional dwell time multiplied by the number of on-board passengers of trip $i \in T$. Note that since $c_i^{DW_O}$ and $c_i^{DW_P}$ are constants which depend on trip $t \in T$, they can be joined in a cost c_i^{DW}.

The relative weights assigned to c_{ijk}, c^{TR}, c_i^{DW} will influence the solution outcome. In the case study, we have discussed with public transport authority Movia how to select the weights so that they express costs in monetary units. This calibration allows to directly compare the objectives. In addition, a sensitivity analysis will be performed.

4.7. Decision variables and mathematical model

The problem is formulated using the following sets of decision variables:

- Binary variables x_{ijk} for all $(i, j, k) \in Q$, which take the value 1 if and only if a vehicle from depot k travels from node i directly to node j, and 0 otherwise;
- Non-negative integer variables τ_{is}^d indicating the departure time of trip $i \in T$ from stop $s \in J_i \cup \{st_i\}$ in minutes from midnight;
- Non-negative integer variables τ_{is}^a indicating the arrival time of trip $i \in T$ at stop $s \in J_i \cup \{et_i\}$ in minutes from midnight;
- Non-negative real variables γ_r which store the excess transfer time for passengers using transfer opportunity $r \in R$ in minutes;
- Binary variables α_{ijs} which take the value 1 if and only if passengers of transfer opportunity $r = (i, l_j, s) \in R$ are embarking trip $j \in T$, and 0 otherwise;
- Non-negative integer variables δ_i which store the amount of dwell time added to trip $i \in T$ in minutes. These variables are not necessary in our formulation and are only used to simplify the presentation of the model and improve readability.

The α_{ijs} variables indicating transfer opportunities are created only for each transfer opportunity $r = (i, l, s) \in R$ and for a set $W(r) = \{(i, j, s) | j \in T_l, i \neq j, a^-_{is} + e_r \leq d^+_{js}, a^+_{is} + e_r + 1.5h_l \geq d^-_{js}\}$, where h_l is the largest frequency observed for line $l \in L$ throughout the day. Moreover, let W be the set of all triplets for all transfer opportunities defined as $W = \cup_{r \in R} W(r)$. In this way, the number of α_{ijs} variables created is reduced, improving tractability of the model without imposing any practical constraints, since at least one transfer to a trip in $l \in L$ will be available given the timetable modifications.

A MIP formulation for the IT-VSP is:

$$\min \sum_{(i,j,k) \in Q} c_{ijk} x_{ijk} + \sum_{i \in T} c^{DW}_i \delta_i + c^{TR} \sum_{r \in R} f_r \gamma_r \tag{1}$$

$$\text{s.t.} \sum_{(i,j,k) \in Q} x_{ijk} = 1 \qquad i \in T \tag{2}$$

$$\sum_{(i,j,k) \in Q} x_{ijk} - \sum_{(j,i,k) \in Q} x_{jik} = 0 \qquad k \in K \quad j \in V_k \tag{3}$$

$$\sum_{(i,j,k) \in Q^O} x_{ijk} \leq v_k \qquad k \in K \tag{4}$$

$$d^-_{i,st_i} \leq \tau^d_{i,st_i} \leq d^+_{i,st_i} \qquad i \in T \tag{5}$$

$$0 \leq \tau^d_{is} - \tau^a_{is} - w^-_{is} \leq w^+_{is} \qquad i \in T \quad s \in J_i \tag{6}$$

$$\delta_i \leq w \qquad i \in T \tag{7}$$

$$\delta_i = \tau^a_{i,et_i} - \tau^d_{i,st_i} - t_i \qquad i \in T \tag{8}$$

$$h^-_{is} \leq \tau^d_{is} - \tau^d_{i-1,s} \leq h^+_{is} \qquad l \in L \quad i \in T_l : i \notin T^1 \quad s \in J_i \cup \{st_i\} \tag{9}$$

$$\tau^a_{i,et_i} + b_{ij} + q^- - \delta_i - M(1 - \sum_{(i,j,k) \in Q} x_{ijk}) \leq \tau^d_{j,st_j} \qquad (i,j) \in T(Q) \tag{10}$$

$$\sum_{(i,j,s) \in W(r)} \alpha_{ijs} = 1 \qquad r = (i, l, s) \in R \tag{11}$$

$$\tau^d_{js} - \tau^a_{is} - e_r \geq M(\alpha_{ijs} - 1) \qquad r \in R \quad (i,j,s) \in W(r) \tag{12}$$

$$M \sum_{\substack{(i,k,s) \in W(r), \\ k \leq j}} \alpha_{iks} \geq \tau^d_{js} - \tau^a_{is} - e_r \qquad r \in R \quad (i,j,s) \in W(r) \tag{13}$$

$$\tau^d_{js} - \tau^a_{is} - e_r - M(1 - \alpha_{ijs}) \leq \gamma_r \qquad r \in R \quad (i,j,s) \in W(r) \tag{14}$$

$$x_{ijk} \in \{0, 1\} \qquad (i,j,k) \in Q \tag{15}$$

$$\tau^d_{is} \in \mathbb{Z}_+ \qquad i \in T \quad s \in J_i \cup \{st_i\} \tag{16}$$

$$\tau^a_{is} \in \mathbb{Z}_+ \qquad i \in T \quad s \in J_i \cup \{et_i\} \tag{17}$$

$$\delta_i \in \mathbb{Z}_+ \qquad i \in T \tag{18}$$

$$\gamma_r \in \mathbb{R}_+ \qquad r \in R \tag{19}$$

$$\alpha_{ijs} \in \{0, 1\} \qquad (i,j,s) \in W \tag{20}$$

The objective function (1) minimizes a weighted sum of operational and passenger costs. The first term accounts for the deadhead, pull out, and pull in costs for the vehicle movements selected by the model. The second term penalizes operational and on-board passenger costs incurred when adding dwell times. The third term addresses the transfer costs. Indeed, the addition of dwell time to trips increases the travel time for on-board passengers, as well as the in-service time of vehicles.

Constraints (2) - (4) are classical MDVSP constraints. Constraints (2) guarantee that each trip $i \in T$ is included in exactly one vehicle schedule. Constraints (3) are flow conservation constraints for the trip and depot nodes, guaranteeing the continuity of the vehicle schedules created. Constraints (4) are capacity constraints that limit the number of pull-out trips to the maximum number of vehicles available at each depot $k \in K$.

The allowed timetable modifications are modelled in constraints (5) - (9). Constraints (5) force the departure time from the first stop of each trip to lie within the bounds defined for its lower and upper shifts. Constraints (6) ensure that the dwell time at each stop of a trip is increased by no more than the maximum dwell time allowed, with respect to the original timetable. Constraints (7) impose that the total added dwell time to all stops of a trip does not exceed the maximum allowed w. Constraints (8) set the values of the δ_i variables to the total added dwell time in the corresponding trip. The minimum and maximum headways between each trip $i \in T$ and its precedent trip in the same directed line at each stop $s \in J_i \cup \{st_i\}$ are modelled with constraints (9).

The vehicle scheduling and the timetable modification parts of the problem are linked in constraints (10). These guarantee that if trips i and j are operated consecutively by the same vehicle, then the vehicle has time to deadhead from et_i to st_j without violating the minimum turnaround time q^-.

Constraints (11) guarantee that passengers from all transfer opportunities $r = (i, l, s) \in R$ are able to transfer, by selecting exactly one transfer to trip $j \in T_l$. The transfer variables α_{ijs} are linked with the departure and arrival times of trips through constraints (12) and (13). Constraints (12) prevent variable α_{ijs} from taking value 1 whenever passengers do not have enough time to transfer from trip i to trip j at stop s, where $(i, l(j), s) \in R$. Constraints (13) are lifting constraints which ensure that passengers arriving from trip i at stop s transfer to one of the trips j, such that $(i, l(j), s) \in R$, if the arrival and departure times allow the transfer to take place. Constraints (13) are in fact not needed for the model to produce feasible solutions, but strengthen the performance of the model. The excess transfer times are stored in the γ_r variables by constraints (14), which determine this value for each transfer opportunity based on the selected transfers. Together, constraints (11)-(14) ensure that passengers transfer to the first available trip in the desired line. The selected trip depends on the timetable modifications. Finally, the range of the sets of decision variables used in the model is defined in constraints (15)-(20).

Numerical example of constraints (11)-(14)**:** To illustrate how constraints (11)-(14) influence the feasibility of the solutions, consider the transfer opportunity $r = (i, l, s) \in R$ depicted in Figure 3, where $l = 400\text{S}$ and $s = \text{Lyngby St}$. Let us assume that trip i can arrive between 9:23 and 9:42 and that trips j_1, j_2 and j_3 can shift 5 minutes forward or backward, derived from the headway of line 400S. Input to the model is the set $W(r)$, that in this example contains two triplets, (i, j_2, s) and (i, j_3, s). Because timetable modifications do not allow to transfer from trip i to trip j_1, the triplet (i, j_1, s) is not part of $W(r)$. Indeed, trip i arrives at s at 9:23 at the earliest, and trip j_1 can depart at 9:25 at the latest, so the minimum transfer time of 4 minutes does not allow passengers to embark on trip j_1. Constraints (11)-(14) ensure that at least one of the transfer options in $W(r)$ is feasible, and that transfer time is calculated as the minimum transfer time over all feasible options. Specifically, constraint (11) becomes $\alpha_{i,j_2,s} + \alpha_{i,j_3,s} = 1$, modeling that passengers disembarking from trip i have to transfer either to trip j_2 or to trip j_3, distinguishing two cases: either a) $\alpha_{i,j_2,s} = 1$, or b) $\alpha_{i,j_3,s} = 1$. Let's consider case a). Constraints (11) force $\alpha_{i,j_3,s}$ to be equal to zero, since passengers of a transfer opportunity transfer to only one trip. Constraints (12) become $\tau^d_{j_2,s} - \tau^a_{is} - e_r \geq 0$ and $\tau^d_{j_3,s} - \tau^a_{is} - e_r \geq -M$ respectively for j_2 and j_3, forcing the transfer from i to j_2 to be feasible. Constraints (13) become $M \geq \tau^d_{j_2 s} - \tau^a_{is} - e_r$ and $M \geq \tau^d_{j_3 s} - \tau^a_{is} - e_r$. Constraints (14) become $\tau^d_{j_2,s} - \tau^a_{is} - e_r \leq \gamma_r$ and $\tau^d_{j_3,s} - \tau^a_{is} - e_r - M \leq \gamma_r$ respectively for j_2 and j_3, which force γ_r to equal the waiting time for the transfer i-j_2. Case b) is similar to case a), with the inverse order of which constraints must hold.

5. A matheuristic approach

The matheuristic approach, which is denoted as MHeu, is based on the MIP formulation of Section 4. Real-life instances of the IT-VSP are intractable when solving the MIP directly with a general solver, so a heuristic approach is needed.

5.1. Outline

The MHeu is an iterative algorithm where at each iteration timetable modifications are allowed for a subset of timetabled trips $T' \subseteq T$ only. An iteration consists of solving the thus restricted IT-VSP problem, which we denote as IT-VSP(T'), where variables τ^a_{is} and τ^d_{is} for all trips $i \notin T'$ are fixed to their values in the current best solution. Although timetable changes are only allowed for T', the vehicle scheduling part of the IT-VSP is solved for all trips. Different selection strategies for constructing T' are compared and will be explained in Section 5.2.

Algorithm 1 outlines the MHeu in pseudo code. Input consists of a set of original timetabled trips T, an initial timetable $\mathcal{T}_0$, the set of transfer opportunities R, a *stopCriterion*, a selection strategy `Strat`, and a set of `Strat` dependent parameters ϑ(`Strat`). The *stopCriterion* is either the number of iterations or total running time, depending on `Strat`. The algorithm starts by solving the MDVSP in Line 1 without allowing timetable modifications, thus τ^d_{is}, τ^a_{is} fixed to $\mathcal{T}_0$ for all $i \in T$. This generates an initial solution $\mathcal{S}^*$ composed by vehicle schedules X_0 and the initial timetable $\mathcal{T}_0$. The iterative procedure is described in Lines 3 - 7, which runs until the *stopCriterion* is met.

Algorithm 1 : MHeu

```
   Input: $T$, $\mathcal{T}_0$, $R$, stopCriterion, Strat, $\vartheta$(Strat)
   Initialization:
1: $\mathcal{S}^* = (\mathrm{X}_0, \mathcal{T}_0) \leftarrow$ solve IT-VSP(1)-(20) with $\tau^d_{is}$, $\tau^a_{is}$ fixed to $\mathcal{T}_0$ for all $i \in T$
2: $\eta = 0$
   Matheuristic:
3: while stopCriterion not reached do
4:     $T' \leftarrow$ selectTrips($\mathcal{S}^*$,Strat,$\vartheta$(Strat))
5:     $\mathcal{S}^* = (\mathrm{X}_\eta, \mathcal{T}_\eta) \leftarrow$ solve IT-VSP(1)-(20) with $\tau^d_{is}$, $\tau^a_{is}$ fixed to $\mathcal{T}_{\eta-1}$ for all $i \in T \setminus T'$
6:     $\eta = \eta + 1$
7: end while
8: return $\mathcal{S}^*$
```

Each iteration starts in Line 4 by selecting the subset of trips $T' \subset T$ according to `Strat` (one of the trip selection strategies described in Section 5.2), and using the set of parameters ϑ(`Strat`). These parameters define how many and which trips are selected, and the maximum running time for each iteration. Timetable modifications in arrival and departure times are allowed for trips in T' only. A new solution is calculated in Line 5 by solving the restricted IT-VSP(T'), with τ^d_{is}, τ^a_{is} fixed to $\mathcal{T}_{\eta-1}$ for all $i \in T \setminus T'$. The solution obtained is always at least as good as the current best solution, since the current best solution is always feasible. To ensure that a solution is always found, each iteration starts from the current best solution, using `CPLEX` warm-start. The best solution $\mathcal{S}^*$ found is returned once the *stopCriterion* is met.

5.2. Trip selection strategies

A run of the MHeu uses one and only one of the four trip selection strategies defined in this section. A trip selection strategy consists of a parameter specifying the size of the selected trip set, and a rule for selecting trips from $\mathcal{S}^*$. Each selection strategy also uses a parameter ψ that defines the maximum running time of each iteration. We propose the following selection strategies:

- Random (`Rand`, ϑ(`Rand`)=$\{\psi, \kappa\}$): selects κ trips of $\mathcal{S}^*$, where any trip $t \in \mathcal{S}^*$ has an equal probability of being selected for T' at each iteration. The stopping criterion for the `Rand` strategy is total running time.
- Rolling Horizon (`RolH`, ϑ(`RolH`)=$\{\psi, \Omega, \xi\}$): deterministic procedure that defines a set Ω of equally long time intervals. Each time interval $\omega \in \Omega$ is defined by a time window $[s_\omega, e_\omega]$, where s_ω is the start time and e_ω is the end time of the interval. The strategy runs in $|\Omega|$ iterations, each of them referring to one time interval $\omega \in \Omega$. At iteration ω, T' is composed by all trips in $\mathcal{S}^*$ with start time belonging to $[s_\omega, e_\omega]$. Consecutive intervals overlap each other by a percentage defined by a parameter ξ. The stopping criterion is the number of iterations, and these are indirectly constrained by a maximum total running time.

  Consider a small example with trips starting between 6:00 and 22:00. Suppose we want to solve the MHeu with the `RolH` with $|\Omega| = 5$ and $\xi = 25\%$. We define the 5 equally long intervals: [6:00,10:00], [9:00,13:00], [12:00,16:00], [15:00,19:00], [18:00,22:00], which overlap each other by 60 minutes (25% of the size of the interval). If the total computational time allowed is equal to one hour, then $\psi = \frac{60}{5} = 12$ minutes.
- Cost Probability (`CostP`, ϑ(`CostP`)=$\{\psi, \kappa\}$): selects κ trips of $\mathcal{S}^*$, where the probability of selecting a trip $t \in \mathcal{S}^*$ is calculated as

  $$p(t) = \frac{\texttt{TrC}(\mathcal{S}^*, t)}{\sum\limits_{i \in T} \texttt{TrC}(\mathcal{S}^*, i)}$$

  where $\texttt{TrC}(\mathcal{S}^*, t)$ are the transfer costs associated with trip t in solution $\mathcal{S}^*$, i.e., transfer costs incurred in $\mathcal{S}^*$ by transferring to or from t. If the number of trips with transfer costs is lower than κ, all trips with transfer costs are selected and trips without transfer costs are randomly selected until κ is reached. The stopping criterion for the `CostP` strategy is total running time.
- Relatedness (`Relat`, ϑ(`Relat`)=$\{\psi, \kappa\}$): We define a new subset $\widehat{T}_t \subseteq \mathcal{S}^*$, which contains all trips $\widehat{t}$ related to a trip $t \in T$, in $\mathcal{S}^*$. A trip $\widehat{t}$ is related to t in $\mathcal{S}^*$ when either t and $\widehat{t}$ belong to the same vehicle schedule in $\mathcal{S}^*$, or when there exist passenger transfers between trips t and $\widehat{t}$ in $\mathcal{S}^*$. Starting with $T' = \emptyset$, this selection strategy iteratively selects a random trip $t \in \mathcal{S}^*$, and adds t and all trips $\widehat{t} \in \widehat{T}_t$ to T'. This process is repeated until $|T'| = \kappa$. The stopping criterion for the `Relat` strategy is total running time.

The `Rand` strategy provides a base scenario. The `RolH` represents a methodology where start time of trips is important, selecting for modifications trips that have a higher chance of sharing transfers. The `CostP` strategy addresses transfer cost optimization, by selecting trips that in the current solution have high transfer costs. Finally, the `Relat` strategy is directly linked to the objectives of minimizing excess transfer time and vehicle schedule costs, by selecting trips that are related with each other in the current solution, in terms of vehicle schedules or transfers.

6. Case study

In this work, we focus on the 8 bi-directional express-bus lines (S-Bus) in the Greater Copenhagen area, which provide faster routes than regular bus lines, with fewer stops, and complement the local train service (S-Train) lines across and radially. Figure 6 is a geographical representation of the S-Bus and S-Train service. The bus network in the Greater Copenhagen area acts as a supplement to the S-Train service, which forms the so-called five finger plan resembling the five fingers of a hand. Movia is the public transport agency responsible for the planning of buses in the region of Zealand, and provided data for the case study. Shifts and stretches are allowed for S-Bus lines' trips, and the S-Train lines operate according to a fixed timetable. The vehicle scheduling component of the problem includes the bus trips only; the vehicle scheduling of trains is not included.

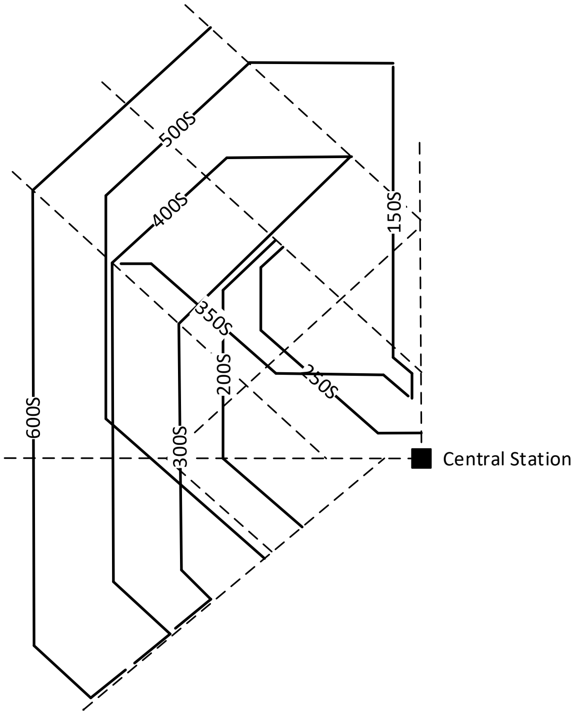

Figure 6: Geographic representation of the S-Bus network. The dashed lines represent the S-Train lines.

The IT-VSP data input components are: (i) an *initial timetable* for the bus lines, which defines the set of all trips and also bus line frequencies used to calculate minimum and maximum headways in the new solution; (ii) *fixed timetables* for train lines; (iii) a *distance matrix* which includes all distances between trip terminal stops and depots; (iv) the *number of transferring passengers* using each transfer opportunity, which can be a bus-bus, bus-train, or train-bus transfer; (v) *costs and parameters*, namely minimum and maximum turnaround times, minimum transfer times at different transfer opportunities, vehicle operational costs, fixed costs per vehicle schedule, value of time costs for passenger excess transfer time, driving speed for vehicles while deadheading, maximum deadhead distance, maximum added dwell time per trip and per stop, and depot capacities.

Input components (i) and (ii) are publicly available. Deadhead distances (iii) were obtained using geographical data. The number of passengers using each transfer opportunity (iv) was estimated based on Movia's data on the number of embarking and disembarking passengers at each stop for each bus trip. The set of transfer opportunities R is input to our model, and is the same from iteration to iteration in the matheuristic. It was defined and provided by Movia, and the computation of this set is therefore not part of our model. The costs and parameters (v) were estimated in collaboration with Movia. They provided estimates of operational waiting time, distance, and schedule costs expressed in monetary units, which together defines the operational costs. Excess waiting time is weighted by a value of time factor, this weighted sum defining transfer costs. The objective minimizes the sum of operational costs and transfer costs. Due to lack of data on on-board passengers, our case study assumes a unique c_i^{DW} for all trips.

We derive 3 instances of different sizes from the case study, with respectively 3, 5, and 8 S-Bus lines, which are described in Table 1. The 3-line instance consists of the most central circular bus lines (200S, 300S, and 400S), the 5-line instance adds 2 more rural circular bus lines to the 3-line instance (500S, and 600S), and the 8-line instance adds 3 additional radial bus lines to the 5-line instance (150S, 250S, and 350S). The first column in Table 1 is the instance index, and the second column is the number of undirected bus lines considered. The three remaining columns are the number of transfers for respectively bus-bus, bus-train, and train-bus transfer opportunities.

Instance	Bus Lines	Trips	Bus-Bus Transfers	Bus-Train Transfers	Train-Bus Transfers
1	3	556	48	375	853
2	5	864	128	554	1150
3	8	1585	360	1109	1644

Table 1: Information on the different instances considered

It can be observed that the instances increase in both number of trips and number of transfer opportunities. All three instances include 7 train lines with a total of 1308 trips with fixed timetables.

To create vehicle schedules that cover the initial timetables, we solve the MDVSP without allowing timetable modifications. In the 8 line instance, the vehicle schedules consist of 1585 bus trips assigned to 205 vehicles. Each schedule starts and ends in the same depot and it is allowed to service trips from different lines in the same schedule, thus allowing deadheading between consecutive trips in a schedule.

The case study includes time dependent service bus travel times and constant deadhead speeds along the day. Vehicles can service trips from different bus lines in the same schedule, known at Movia as *interlining*. The maximum deadhead distance is 15 kilometres (i.e., $u = 15$), the minimum turnaround time is 12 minutes (i.e., $q^- = 12$), and the maximum turnaround time is 30 minutes (i.e., $q^+ = 30$). The dwell time at each stop with transfers can be increased by up to 3 minutes (i.e., $w^+_{is} = 3, i \in T, s \in J_i$), and a maximum of 10 minutes of dwell time can be added in total to a trip (i.e., $w = 10$). The additional dwell time is deducted from the buffer in the turnaround time at the end of the trip. The shifts allowed in each trip departure time were created based on the original timetables for each bus line. Considering consecutively timetabled trips $(i-1), i, (i+1) \in T_l$ and with departure time from the first stop $d_{i-1,st_i}, d_{i,st_i}, d_{i+1,st_i}$ respectively, the lower and upper shift limits for trip i are calculated with the expressions

$$d^-_{i,st_i} = d_{i,st_i} - \left\lfloor \frac{d_{i,st_i} - d_{i-1,st_i} - 1}{2} \right\rfloor$$

$$d^+_{i,st_i} = d_{i,st_i} + \left\lfloor \frac{d_{i+1,st_i} - d_{i,st_i}}{2} \right\rfloor$$

ensuring that trips can never overtake each other in the timetable. At each stop, trip modifications are also bounded by the minimum and maximum headways. For each trip $i \in T_l$ at each stop $s \in J_i \cup \{st_i\}$, minimum and maximum headways, h^-_{is} and h^+_{is}, are calculated based on the scheduled headway in the original timetable. Table 2 shows the allowed variations on headways based on the scheduled headways.

Scheduled headway (minutes)	Minimum and maximum headway variation (minutes)
$\leq$ 4	+/- 1
$\leq$ 12	+/- 2
$\leq$ 20	+/- 3
$\leq$ 39	+/- 4
$\geq$ 40	+/- 5

Table 2: Allowed headway variations based on scheduled headways

7. Computational experiments

This section evaluates the performance of the MHeu in a set of computational experiments. It consists of a set of parameter tuning experiments for the selection strategies (Section 7.1), an analysis of computational performance (Section 7.2), and an analysis of solution quality in terms of transfer cost, operational costs and total cost (Section 7.3). The parameter tuning is based on the 3-line instance allowing the addition of stretches for a total running time of 30 minutes. Computational performance and solution quality are evaluated and compared for solving the MHeu with and without allowing stretches for all selection strategies, for all 3 instances (3, 5 and 8 lines) and for a total running time of 1, 5 and 12 hours, inspired by the running time limits used in Petersen et al. (2013) and justified by the fact that the problem is addressed at the tactical level. Furthermore, convergence, trade-off between operational and transfer costs, the quality of resulting vehicle schedules, and the distribution of excess transfer time are discussed.

The algorithm was implemented in C and the mathematical formulations were solved using CPLEX version 12.6. The experiments were conducted on HPC servers, using Intel Xeon E5-2660 v3 2.60GHz processors, and 1 computation core. Each iteration used CPLEX warm-start to start from the best solution found so far. All results presented are average results over five runs for each different setting, except for the RolH strategy as it is deterministic.

The following measures are used to express computational performance and solution quality. The computational performance of the MHeu algorithm is expressed in relation to solving the IT-VSP(T) directly in CPLEX with a maximum computation time of 24 hours. Performance is expressed as Gap: the average percentage gap to the best lower bound obtained in 24 hours; as well as Gap*: the average percentage gap to the best known upper bound obtained in 24 hours, e.g. the best obtained integer solution. The formulas to calculate Gap and Gap* are:

$$\mathtt{Gap} = \frac{\mathcal{S}^{AVG}_{\mathtt{MHeu}} - \mathcal{S}^{LB}_{\mathtt{IT-VSP}}}{\mathcal{S}^{LB}_{\mathtt{IT-VSP}}} \qquad , \qquad \mathtt{Gap}^* = \frac{\mathcal{S}^{AVG}_{\mathtt{MHeu}} - \mathcal{S}^{UB}_{\mathtt{IT-VSP}}}{\mathcal{S}^{UB}_{\mathtt{IT-VSP}}}$$

where $\mathcal{S}^{AVG}_{\mathtt{MHeu}}$ is the average objective value over all five runs of an instance and setting of MHeu, $\mathcal{S}^{LB}_{\mathtt{IT-VSP}}$ is the best lower bound obtained with CPLEX solving IT-VSP(T) in 24 hours, and $\mathcal{S}^{UB}_{\mathtt{IT-VSP}}$ is the objective value of the best integer solution obtained with CPLEX solving IT-VSP(T) in 24 hours. The values $\mathcal{S}^{LB}_{\mathtt{IT-VSP}}$ and $\mathcal{S}^{UB}_{\mathtt{IT-VSP}}$ are computed for each instance and each setting of timetable modifications (with or without stretches) separately. Even when extending the computation time from 24 hours to 7 days we were not able to identify the optimal solution, and the decrease in Gap is only between 0.04% to 0.64% over all instances.

The solution quality of $\mathcal{S}^{AVG}_{\mathtt{MHeu}}$ is expressed in terms of transfer cost (TrC), operational costs (OpC), and total costs (TC), which are average percentage differences to the optimal base solution $S_{\mathtt{MDVSP}}$ of IT-VSP($\varnothing$). The $S_{\mathtt{MDVSP}}$ represents the best solution without integration of vehicle scheduling and timetabling, on which we aim to improve in terms of transfer time and operational costs. Let $x = \{\texttt{TrC}, \texttt{OpC}, \texttt{TC}\}$ and $f_x(S)$ denote the x-type cost of a solution S, then these quality measures are computed as:

$$x = \frac{f_x(\mathcal{S}_{\texttt{MHeu}}^{AVG}) - f_x(\mathcal{S}_{\texttt{MDVSP}})}{f_x(\mathcal{S}_{\texttt{MDVSP}})} \times 100\%$$

A negative percentage for x corresponds to a reduction of costs in the MHeu solution in comparison to the non-integrated $S_{\texttt{MDVSP}}$ base solution.

7.1. Parameter tuning

Table 3 shows the parameter tuning results for the `Rand`, `CostP`, and `Relat` selection strategies. All three strategies require the same set of parameters as input, which consists of the number of trips selected κ for T' at each iteration, and the maximum running time of each iteration in minutes ψ. We tested $\kappa = 150, 200, 250, 300, 350, 400, 450$ trips and $\psi = 0.5, 1, 2, 5, 10$ minutes, and report average results over five runs.

		κ						
Strategy	ψ (m)	150	200	250	300	350	400	450
	0.5	-2.15%	-3.21%	-4.36%	-5.71%	-6.51%	-6.32%	5.01%
	1	-1.87%	-3.74%	-4.57%	-5.80%	-6.02%	-6.35%	-4.23%
`Rand`	2	-1.85%	-2.90%	-4.61%	-5.67%	-6.23%	**-6.70%**	-3.67%
	5	-2.13%	-2.51%	-4.43%	-4.90%	-6.18%	-5.69%	-4.74%
	10	-2.42%	-3.05%	-4.36%	-5.09%	-4.82%	-3.41%	-3.37%
	0.5	-2.36%	-2.98%	-4.74%	-5.25%	-5.21%	-6.01%	5.01%
	1	-1.94%	-3.35%	-4.73%	-5.44%	-5.99%	**-6.16%**	-4.98%
`CostP`	2	-2.11%	-2.61%	-4.52%	-5.55%	-5.66%	-6.00%	-3.75%
	5	-1.87%	-3.54%	-4.58%	-5.19%	-5.43%	-6.01%	-5.40%
	10	-1.96%	-2.66%	-5.01%	-4.93%	-5.69%	-5.13%	-3.89%
	0.5	-2.35%	-2.80%	-4.51%	-5.47%	-6.06%	-4.58%	5.01%
	1	-2.42%	-3.28%	-3.93%	-5.17%	**-6.45%**	-5.39%	1.78%
`Relat`	2	-2.69%	-3.58%	-4.57%	-4.97%	-6.01%	-6.41%	-3.20%
	5	-2.51%	-3.05%	-4.04%	-4.85%	-5.21%	-4.94%	-4.08%
	10	-2.79%	-3.76%	-4.07%	-4.51%	-4.63%	-3.23%	-3.37%

Table 3: Average `Gap`* for the `Rand`, `CostP`, and `Relat` selection strategies, with a total running time of 30 minutes

Table 3 contains the average `Gap`* per selection strategy for all parameters, where minimal average `Gap`* per selection strategy is marked in bold. The average `Gap`* per selection strategy is smallest for $\psi = 2$, $\psi = 1$, and $\psi = 1$ for `Rand`, `CostP` and `Relat` respectively, thus indicating that running more iterations with a short computational time may be more beneficial than running less iterations with a long computational time for solving `CPLEX` in each iteration. Furthermore, the lowest average `Gap`* is obtained for $\kappa = 400$, $\kappa = 400$, and $\kappa = 350$ for `Rand`, `CostP` and `Relat` respectively, indicating that a larger sub-problem, with a larger solution space, may provide better solutions. However, for $\kappa = 450$, computation times of 0.5 minutes may be too small: all selection strategies find worse solutions for this case, and for the `Relat` this also holds for $\psi = 1$ minute. For all other parameters the MHeu improves on the `CPLEX` solution, with `Gap`* between -1.85% and -6.70%. However differences per strategy for improved solutions can be up to 4%, while the second-best setting is no more than 0.35% from the best one and in 1 out of 3 cases is obtained with different values for both κ and ψ than the ones for the lowest gap. Furthermore, `Gap`s sometimes form an oscillating pattern, for example for `Rand` $\kappa = 150$.

Table 4 contains average `Gap`* for the `RolH` selection strategy, with the lowest average `Gap`* marked in bold. The parameters for this selection strategy are the number of time intervals $|\Omega|$ and the percentage overlap ξ between consecutive time intervals. Provided the overall computation time limit of 30 minutes and a fixed number of iteration for this approach, the time limit per iteration is set to $\psi = \frac{30}{|\Omega|}$. The values tested for the number of time intervals are $|\Omega| = 2, 4, 6, 8, 10, 12, 14, 16$ intervals, and the percentage overlaps tested are $\xi = 20, 25, 30, 35, 40$. Both RolH parameters influence how many trips are selected for T' at each iteration. For each setting, we report the size of $|\Omega|$ and the maximum number of trips κ^{max} selected in each time interval of Ω between brackets: $|\Omega|(\kappa^{max})$. Since this strategy is deterministic, only one run of each setting is conducted.

		$\|\Omega\|(\kappa^{max})$							
Strategy	ξ (%)	16 (75)	14 (82)	12 (95)	10 (117)	8 (134)	6 (167)	4 (220)	2 (413)
	20	-2.48%	-3.05%	-2.26%	-2.25%	-1.96%	-2.74%	-1.17%	-1.04%
	25	-3.03%	-2.97%	-4.60%	-3.49%	-4.55%	-3.58%	-1.40%	-1.13%
RolH	30	-3.07%	-3.99%	-4.27%	-4.33%	-5.34%	-1.43%	-2.42%	2.33%
	35	-4.14%	-3.06%	-2.68%	-3.46%	**-5.83%**	-1.33%	-3.31%	-1.28%
	40	-4.29%	-5.28%	-3.17%	-4.55%	-3.37%	-2.57%	-1.36%	-1.11%

Table 4: Average Gap* for the RolH selection strategy

The results presented in Table 4 show that contrary to previous selection methods the RolH strategy has the best performance for a relatively small sub-problem of around 134 trips ($|\Omega|$=8) rather than for larger sub-problems of around 400 trips ($|\Omega|$=2). The Gap* decreases when the number of time intervals increases from 4 to 8. The overall smallest Gap* of -5.83% is obtained for $\xi = 35\%$. Further increasing $|\Omega|$ does not improve the results.

The parameter settings resulting from the parameter tuning experiments are summarized in Table 5 per instance. It was necessary to increase the runtime per iteration for the 8-line instance as its increased size and complexity did not allow building the model within the specified time. The number of time intervals for the RolH strategy in the 5 and 8 lines instances were selected to resemble the κ^{max} parameter of the 3-line instance at $|\Omega| = 8$ and $\xi = 30\%$.

		Instance		
Strategy	Parameter	3 Lines	5 Lines	8 Lines
Rand	κ	400	400	400
	ψ (m)	2	2	5
RolH	$\|\Omega\|(\kappa^{max})$	8 (134)	14 (141)	24 (149)
	ξ(%)	35	35	35
CostP	κ	400	400	400
	ψ (m)	1	1	5
Relat	κ	350	350	350
	ψ (m)	1	1	5

Table 5: Parameter values used for each of the 3 instances and each of the selection strategies

7.2. Computational performance

In this section, we analyze the computational performance of the different selection strategies given different total running time limits for the 3, 5 and 8 lines instances.

7.2.1. Selection Strategies and Running Time

Table 6 contains the (upper bound) Gap* and (lower bound) Gap defined in Section 7, which express the performance of the MHeu to the performance of a standard solver (CPLEX) for solving the IT-VSP directly within a time limit of 24 hours.

The MHeu always improves on the CPLEX solution even when running it for only 1 hour, with the only exception being the RolH selection strategy that requires more than 1 hour to find a better feasible solution. For 12 hours of total computational time, the Gap* to the CPLEX solution is between -4,61% and -8,52% when allowing stretches and between -2,81% and -6,48% when not allowing stretches. The improvement in Gap* for 12 hours compared to 5

	Stretches	No						Yes					
	Instance	3 Lines		5 Lines		8 Lines		3 Lines		5 Lines		8 Lines	
Running time (h)	Strategy	Gap*	Gap	Gap*	Gap	Gap*	Gap	Gap*	Gap	Gap*	Gap	Gap*	Gap
1	Rand	-5.59	2.09	-4.44	4.85	-1.58	10.74	-6.87	4.01	-5.00	6.00	-5.78	13.01
	RolH	-4.89	2.86	-4.79	4.46	5.21	18.38	-3.48	7.79	-3.16	8.06	-0.21	19.68
	CostP	-5.43	2.27	-5.05	4.18	-0.80	11.61	-6.96	3.91	-5.14	5.84	-5.81	12.97
	Relat	-5.54	2.15	-4.14	5.17	-0.93	11.46	-6.02	4.96	-4.43	6.64	-5.78	13.01
5	Rand	-5.75	1.92	-5.90	3.24	-2.87	9.29	-7.34	3.48	-5.86	5.04	-8.12	10.20
	RolH	-4.57	3.20	-4.79	4.46	-2.82	9.34	-6.09	4.88	-4.61	6.45	-8.39	9.87
	CostP	-5.84	1.83	-5.71	3.45	-2.63	9.56	-7.10	3.75	-6.09	4.79	-6.91	11.65
	Relat	-5.74	1.93	-5.18	4.03	-3.03	9.10	-7.33	3.49	-5.82	5.09	-7.92	10.43
12	Rand	-6.09	1.55	-6.15	2.96	-3.62	8.43	-7.79	2.98	-6.53	4.30	-8.17	10.14
	RolH	-5.13	2.59	-4.79	4.46	-3.63	8.43	-4.83	6.29	-4.61	6.45	-8.26	10.03
	CostP	-6.00	1.65	-6.48	2.61	-2.81	9.35	-7.26	3.57	-6.83	3.96	-7.87	10.49
	Relat	-5.76	1.91	-5.70	3.46	-4.39	7.57	-7.05	3.80	-6.18	4.69	-8.52	9.72

Table 6: MHeu results with a run time of 1, 5, and 12 hours, with and without stretches, expressed in Gap* and Gap

hours computation time is only between 0.13% and 1.5%, and therefore one could also opt for a shorter computation time than 12 hours. However, a computation time of one hour seems insufficient especially for larger instances. This is especially evidenced by the poor performance of the RolH strategy at one hour computation times. Since the integrated timetabling and vehicle scheduling problem is a tactical level problem, computation times of several hours are non-prohibitive for using this algorithm in practice. Furthermore, for the sake of comparison, all experiments in this paper were run using only one computation core. Increasing the computation cores would most likely reduce the computational times.

As the size of the instance grows, the lower bound Gap increases from around 2% and 4% for the 3-line instance to around 10% and higher for the 8-line instance. Thus the quality of the lower bounds seems to decrease when the instance size increases. The lower bound Gap is somewhat higher when allowing stretches, which is intuitively explained by the wider solution space when allowing stretches.

The gap differences between selection strategies are small for a minimal computation time of five hours: the difference in upper bound Gap* ranges between 0.40% and 1.69% when not allowing stretches, and between 0.65% and 2.97% when allowing stretches. Moreover, there isn't one strategy that consistently returns better results than the others, and all strategies return the best result in at least one of the different combinations of runtime, instances, and allowing or disallowing stretches for a minimum of five hours computation time. The performance over the five runs is relatively stable with standard deviation below 0.82% for these instances.

7.2.2. Convergence of the Different Selection Strategies

The convergence of each selection strategy for the 8-line instance, with stretches, and with a total running time of 5 hours, is depicted in Figure 7. The x-axis reflects the running time and the y-axis represents the total costs, in terms of transfer and operational costs expressed in Danish crowns (DKK). Each curve represents the best run of the algorithm for a specific selection strategy. Markers indicate when the strategy finds an improved feasible solution.

All selection strategies start with a steep decline, which slows down generally after around 100 minutes of computation time. After 3 hours most strategies have converged, the Relat being the only exception in this specific case, however similar small improvements have been observed for Rand after this amount of computation time. Only minor improvements are obtained extending the computation time further than 5 hours, as discussed in Section 7.2.1.

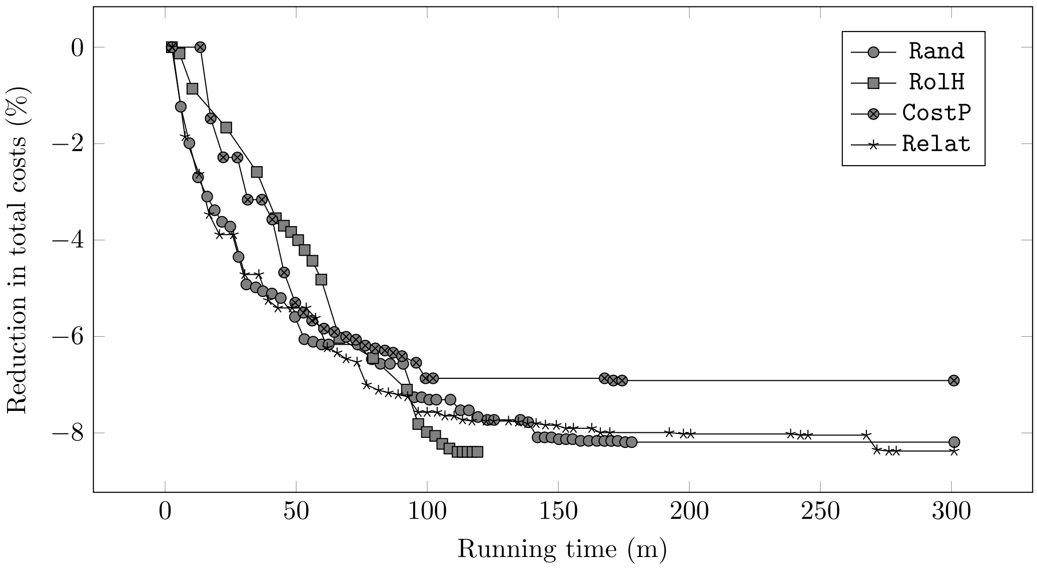

Figure 7: Total costs convergence analysis with $\kappa = 400$ or $|\Omega| = 8$, for the 8-line instance with stretches and a total running time of 5 hours

The `RolH` finds the solution with the lowest total costs for this instance, and does so within half of the total running time limit. The `RolH` runs shorter when at any of the iterations, the optimal solution for the IT-VSP (T') is found within the running time limit of the iteration.

7.3. Transfers and operational costs

This section expresses the performance of the MHeu in comparison to the base MDVSP solution to discuss the benefits in terms of passenger service and operational cost in comparison to current practice. Thus, it focusses on the value that the algorithm may provide public transport agencies, with a specific focus on the allowance to add dwell time to timetabled trips (stretches).

7.3.1. Value of Stretches

Table 7 contains the transfer cost `TrC`, operational cost `OpC`, and total costs `TC` for the 3, 5 and 8 lines instances, with and without stretches, for all four selection strategies and a total running time of 5 hours. The values in bold represent which selection strategy performed best in each category (`TrC`, `OpC`, and `TC`) per instance, for allowing and disallowing stretches independently.

	Instance	3 Lines						5 Lines						8 Lines					
	Stretches	No			Yes			No			Yes			No			Yes		
Running time (h)	Strategy	`TrC`	`OpC`	`TC`	`TrC`	`OpC`	`TC`	`TrC`	`OpC`	`TC`	`TrC`	`OpC`	`TC`	`TrC`	`OpC`	`TC`	`TrC`	`OpC`	`TC`
5	`Rand`	**-31.08**	-7.15	-11.30	-34.86	**-6.91**	**-11.76**	-35.09	**-6.78**	**-10.95**	-39.36	-5.85	-10.77	-34.67	-3.40	-7.69	-37.23	-3.49	-8.12
	`RolH`	-30.27	-5.98	-10.19	-34.22	-5.61	-10.57	-32.89	-5.92	-9.89	-36.78	-4.90	-9.58	-29.46	**-4.18**	-7.64	-36.23	-3.97	**-8.39**
	`CostP`	-30.28	**-7.42**	**-11.38**	**-36.15**	-6.37	-11.53	-35.32	-6.53	-10.76	-38.88	**-6.19**	**-10.99**	**-35.25**	-3.04	-7.46	-35.97	-2.30	-6.91
	`Relat`	-31.06	-7.15	-11.29	-35.41	-6.79	-11.75	**-36.37**	-5.76	-10.26	**-39.45**	-5.79	-10.74	-35.06	-3.52	**-7.84**	**-37.61**	-3.21	-7.92

Table 7: MHeu results with a run time of 5 hours, with and without stretches, expressed in `TrC`, `OpC`, and `TC`

The integration of timetabling and vehicle scheduling can reduce the total costs between 10.19% and 11.76% for the 3-line instance, between 9.58% and 10.99% for the 5-line instance, and between 6.91% and 8.39% for the 8-line instance. The inclusion of stretches in the timetable modifications achieves solutions with total costs comparable to the solutions with only shifts. In a few cases, the addition of stretches shows a small improvement in the total costs, ranging from approximately 0.08% to 0.75% points.

The results obtained in terms of transfer costs are comparable for all selection strategies, over all 6 cases (3, 5 and 8 lines instances, with and without stretches). The `Relat` strategy may be especially suitable to reduce transfer costs: it provides lowest transfer costs in 3 out of 6 cases, and a close second-least transfer costs in two more instances, however differences are too small to draw any definitive conclusions.

To illustrate the value of allowing stretches to decrease transfer times, we define a slightly changed IT-VSP^B that minimizes transfer costs within a budget for operational costs, rather than the weighted sum of both. Specifically, the IT-VSP (1) - (20) is changed by replacing objective (1) with (21) and adding budget constraint 22. Thus IT-VSP^B is defined as

$$\min \quad c^{TR} \sum_{r \in R} f_r \gamma_r \tag{21}$$

$$\sum_{(i,j,k) \in Q} c_{ijk} x_{ijk} + c_i^{DW} \sum_{i \in T} \delta_i \leq \Delta \tag{22}$$

$$(2) - (20)$$

The objective function (21) considers only transfer cost minimization. The operational cost components of (1) are removed from the objective and an additional constraint is added to the model, (22), where the operational cost components removed from (1) are kept below or equal to a certain budget Δ. In these experiments, Δ is defined as the operational costs in the optimal solution to the IT-VSP($\emptyset$) for the respective instance.

Table 8 contains transfer costs, operational costs, and total costs for the 3, 5 and 8 lines instances, with and without allowing stretches. Experiments were run for 5 hours total computation time and the `Rand` selection strategy, as there was no strategy that consistently performed best. Moreover, the increased complexity introduced by the budget constraint and changed objective in IT-VSP^B required to increase run time per iteration of the 8-line instance to 20 minutes.

	Instance	3 Lines			5 Lines			8 Lines		
		TrC	OpC	TC	TrC	OpC	TC	TrC	OpC	TC
Stretches	No	-35.75	0.00	-6.20	-37.81	-0.77	-6.22	-30.61	-0.02	-4.21
	Yes	-47.19	-0.06	-8.23	-48.81	-0.18	-7.33	-42.03	0.00	-5.76

Table 8: Budget results with a run time of 5 hours, with and without stretches, expressed in `TrC`, `OpC`, and `TC`

The results in Table 8 indicate that indeed adding stretches can reduce transfer costs by approximately 11% in all instances, without increasing the operational costs. With stretches, transfer costs reductions are increased from between 30.61% and 37.81% to between 42.03% and 48.81%.

7.3.2. Schedules and Excess Transfer Time

In this section, we compare the vehicle schedules and the excess transfer times in the original timetable MDVSP solution and in the best solution obtained using our MHeu approach. Results refer to the 8-line instance, allowing stretches, and with a total computation time of 5 hours. Figure 8 analyzes the quality of transfers in terms of excess transfer time, which is represented in the x-axis. The y-axis shows the percentage of passengers that experience each value of excess transfer time.

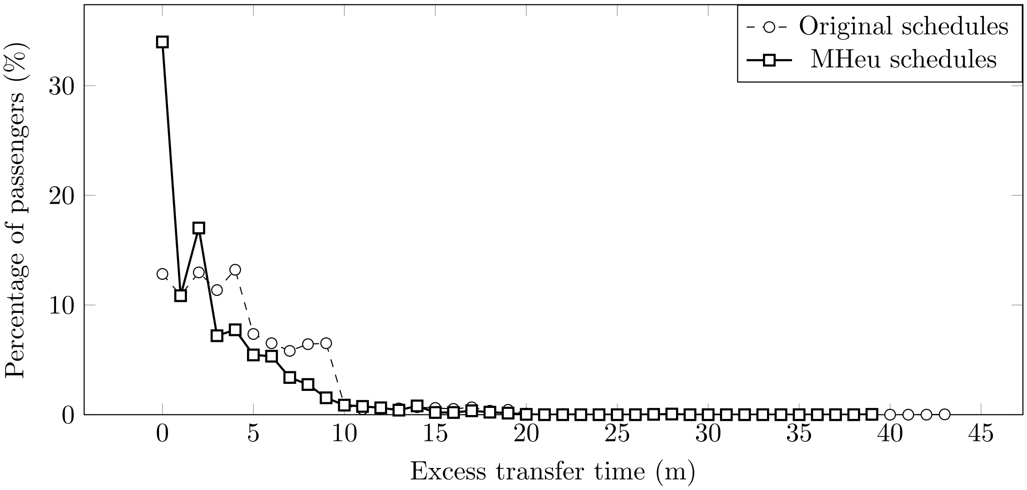

Figure 8: Comparison of transfers in the original solution and in the `Relat` best solution

In the MHeu solution, less passengers experience high excess transfer times. It can be observed that the number of passengers with ideal transfer time (experiencing zero minutes excess transfer time) increased by approximately 175%. In the MHeu solution a total of 34% of all passengers experience 0 minutes of excess transfer time, while only 12.8% of passengers experience 0 minutes excess transfer time in the original timetables. Furthermore, the average excess transfer time decreased from 4.5 minutes to 2.8 minutes in the MHeu solution, and the worst case excess transfer time was also reduced.

Figure 9 shows the number and duration of vehicle schedules for the best solution of the MHeu and for the MDVSP solution resulting from the original timetable. The x-axis contains the duration of schedules in minutes, while the y-axis indicates the number of schedules with a duration up to x minutes. The schedule durations are discretized in intervals of 50 minutes, so a value of 2 in the 600 minute duration means that there are two schedules with duration between 550 and 600 minutes.

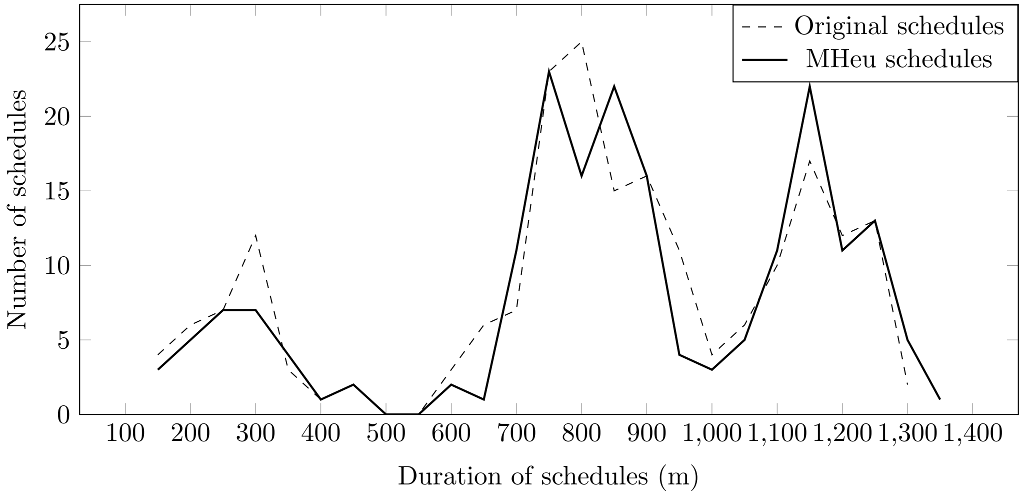

Figure 9: Comparison of schedules in the original solution and in the `Relat` best solution

The MHeu solution reduces the number of schedules from 205 to 195, which corresponds to a decrease of approximately 5%. Furthermore, the average duration of schedules increased by approximately 30 minutes. The percentage of modified trips in the original timetable, by either shifts, stretches, or both, is approximately 74%. A total of 78 trips are stretched. On average, circa 1.5 minutes of dwell time are added per stretched trip. This indicates that the increase in in-vehicle time for on-board passengers will be very limited.

We present further information on the vehicle schedules in Figure 10 and Tables 9 and 10. Figure 10 shows the number of trips per schedule in the original timetable MDVSP solution and in the best solution obtained with the MHeu approach. It can be observed that the MHeu generates schedules with a higher number of trips, being the average number of trips per schedule 8.13, while in the MDVSP solution this average amounts to 7.73.

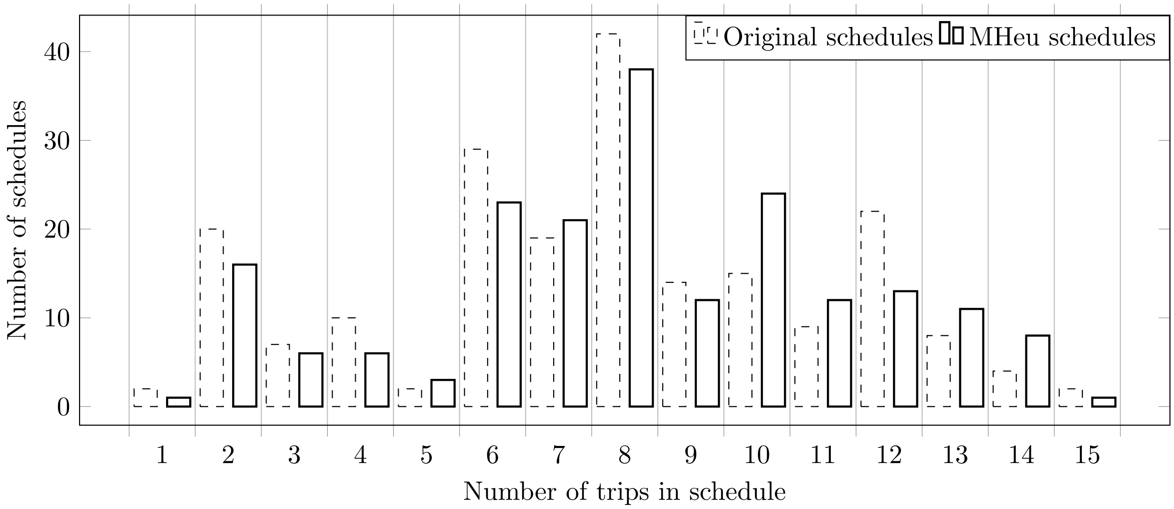

Figure 10: Number of trips in schedules in the original solution and in the Relat best solution

	Original schedules		MHeu schedules	
Depot	Trips	Schedules	Trips	Schedules
1	616	75	630	75
2	461	55	456	52
3	376	55	356	49
4	132	20	143	19

Table 9: Number of trips and number of schedules assigned to each of the four depots in the original and in the MHeu schedules

		MHeu schedules			
	Depot	1	2	3	4
Original Schedules	1	383 (24.2%)	126 (7.9%)	70 (4.4%)	37 (2.3%)
	2	134 (8.5%)	252 (15.9%)	60 (3.8%)	15 (0.9%)
	3	93 (5.9%)	62 (3.9%)	201 (12.7%)	20 (1.3%)
	4	20 (1.3%)	16 (1.0%)	25 (1.6%)	71 (4.5%)

Table 10: Distribution of trips per depot in the original and in the MHeu schedules

Table 9 shows the number of trips and number of schedules assigned to each depot. In both solutions, depot 1 is at its maximum capacity, from which we conclude it is the depot with most convenient location given the set of trips to be serviced. Furthermore, the number of trips assigned to this depot in the improved solution increases, reinforcing the importance of depot 1. Depot 4, although being the depot with lowest number of schedules, also has an increased number of trips in the MHeu solution. Depots 2 and 3 see a decrease in both number of trips and number of schedules when comparing to the MDVSP solution.

Table 10 provides information on how trips are assigned to depots, both in absolute value and in percentage of total number of trips. For example, trips in entry (1,1) are assigned to depot 1 in both the original and in the MHeu solutions, while trips in entry (1,2) are assigned to depot 1 in the original solution and to depot 2 in the MHeu solution. While 57.2% of all trips (907 trips, sum of the diagonal values) are assigned to the same depot in both solutions, 42.8% of all trips shift depots in different solutions, confirming the significant impact that timetable modifications have on vehicle schedules.

7.3.3. Trade-off between Operational Costs and Transfer Costs

In this section, we analyze the trade-off between operational costs and transfer costs by varying the value of time (VOT) for passengers. The VOT considered in the previous experiments was 100 DKK/hour, which corresponds to approximately 14 USD/hour. We considered the additional VOTs= $\{50, 200, 400\}$ DKK/hour, with 50 DKK/hour representing a very low value of time, 100 DKK/hour a standard value of time for commuters, 200 DKK/hour representing business travelers, and 400 DKK/hour representing an extreme high value of time. These values were inspired by value of time studies conducted at the transport modelling center at the Technical University of Denmark, and are available at their website[1]. The experiments were run for 5 hours of total running time, for the 8-line instance, allowing stretches, and the `Rand` strategy. The plots in Figure 11 have different VOTs represented in the x-axis and the percentage differences in the y-axis.

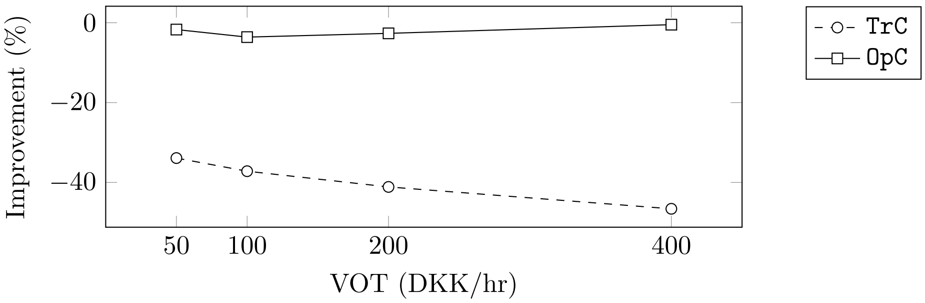

Figure 11: Analysis of the influence of Value of Time

The plots show that increasing the VOT leads to lower transfer costs and higher operational costs. However, the sensitivity of this relation appears to be low. For a value of time of 400 DKK/hour instead of 100 DKK/hour, the operational costs increase by 3.12% (-0.47 - -3.59) for a decrease of -9.42% (-46.61 - -37.19) in transfer costs.

8. Conclusion

This paper proposed a new model for the integrated timetabling and vehicle scheduling problem. Provided an initial timetable, it defines a set of timetable modifications and a set of vehicle schedules with the objective of minimizing passenger excess transfer times and operational costs. Modifications consist of changes in the start time of a trip (shifts), and addition of dwell time at intermediate stops (stretches). The new idea to include the addition of stretches could represent the redistribution of buffer time over trips to create better-timed transfers. Results for a realistic case study for the Greater Copenhagen area indicate that the integration of timetabling and vehicle scheduling may lead to a potential reduction in both transfer costs and operational costs. Moreover, our findings suggest that allowing a wider set of timetable modifications in the form or stretches creates a potential for further reducing transfer costs by up to 10%, without increasing operational costs.

We propose a matheuristic that in each iteration solves the MIP formulation for a sub-problem of the IT-VSP. The sub-problems restrict timetable modifications to a subset of all timetabled trips, while it solves the full integrated vehicle scheduling problem. Several methods for constructing sub-problems are proposed and compared. Results indicate that the matheuristic is able to produce better solutions in terms of transfer time and operational costs than a general purpose solver in 7 days, and does so faster in 1 to 5 hours of computation time. Solutions

[1] `http://www.modelcenter.transport.dtu.dk/Noegletal/Transportoekonomiske-Enhedspriser`

reduce average excess transfer time in comparison to the current timetable from 4.5 to 2.8 minutes, while increasing the number of passengers with ideal transfer times by almost 175% and decreasing worst case excess transfer times. In addition, reductions in operational costs are found in comparison to optimal vehicle schedules for the current timetable. Results for our case study are therefore promising that also for larger networks gains could be obtained from the integrated approach.

Several opportunities for future research exist. First, one could aim to include a dynamic passenger route choice component into the optimization. Indeed, if more favourable transfer connections are provided some passengers may change their route, thus leading to a change in passenger flows. This would also allow a better analysis of the trade-off between added dwell time and increase in travel costs for on-board passengers. The modelling of accurate passenger route choice is a non-trivial task, and therefore left for future research. One could also aim to include crew scheduling constraints into the model, to ensure that there exist feasible crew schedules for the resulting vehicle schedules. Finally, it would be interesting to consider the integration of timetabling and vehicle scheduling in a real-time setting. This would require a focus on increasing computation speed, include new practical constraints for en-route vehicles and drivers, and evaluation of results in a dynamic setting.

Acknowledgements

The authors would like to thank the two anonymous reviewers for the suggestions and comments, which improved the final version of the paper. The authors would also like to thank Stefan Røpke for his support in defining the project in its initial stage and to Movia for providing the data necessary to conduct the case study. This study was conducted with the financial support of InnovationsFonden, to whom the authors would also like to thank.

References

Bookbinder, J. H. and A. Desilets (1992). Transfer Optimization in a Transit Network. *Transportation Science 26*, 106–118.

Carpaneto, G., M. Dellamico, M. Fischetti, and P. Toth (1989). A branch and bound algorithm for the multiple depot vehicle scheduling problem. *Networks 19*(5), 531–548.

Ceder, A. (2001). *Efficient Timetabling and Vehicle Scheduling for Public Transport*, pp. 37–52. Berlin, Heidelberg: Springer Berlin Heidelberg.

Ceder, A. (2011). Optimal Multi-Vehicle Type Transit Timetabling and Vehicle Scheduling. *Procedia - Social and Behavioral Sciences 20*, 19–30.

Ceder, A., B. Golany, and O. Tal (2001). Creating bus timetables with maximal synchronization. *Transportation Research Part A: Policy and Practice 35*(10), 913–928.

Chakroborty, P., K. Deb, and R. K. Sharma (2001). Optimal fleet size distribution and scheduling of transit systems using genetic algorithms. *Transportation Planning and Technology 24*(3), 209–225.

Daduna, J. R. and S. Voß (1995). Practical Experiences In Schedule Synchronization. *Computer-Aided Transit Scheduling*, 39–55.

Desaulniers, G. and M. D. Hickman (2007). Chapter 2 - Public Transit. *Handbooks in Operations Research and Management Science 14*(C), 69–127.

Desrosiers, J., Y. Dumas, M. M. Solomon, and F. Soumis (1995). Time Constrained Routing and Scheduling. *Handbooks in Operations Research and Management Science 8*, 35–139.

Domschke, W. (1989). Schedule synchronization for public transit networks. *OR Spektrum 11*, 17–24.

Guihaire, V. and J.-K. Hao (2008a). Transit network design and scheduling: A global review. *Transportation Research Part A: Policy and Practice 42*(10), 1251–1273.

Guihaire, V. and J.-K. Hao (2008b). Transit Network Re-Timetabling and Vehicle Scheduling. *Modelling, Computation and Optimization in Information Systems and Management Sciences 14*(1), 135–144.

Guihaire, V. and J.-K. Hao (2010). Improving Timetable Quality in Scheduled Transit Networks. *Trends in Applied Intelligent Systems 6096*, 21–30.

Hadjar, A., O. Marcotte, and F. Soumis (2006). A branch-and-cut algorithm for the multiple depot vehicle scheduling problem. *Operations Research 54*(1), 130–149.

Hassold, S. and A. Ceder (2012). Multiobjective Approach to Creating Bus Timetables with Multiple Vehicle Types. *Transportation Research Record: Journal of the Transportation Research Board 2276*, 56–62.

Ibarra-Rojas, O. J., F. Delgado, R. Giesen, and J. C. Munoz (2015). Planning, operation, and control of bus transport systems: A literature review. *Transportation Research Part B-methodological 77*, 38–75.

Ibarra-Rojas, O. J., R. Giesen, and Y. A. Rios-Solis (2014). An integrated approach for timetabling and vehicle scheduling problems to analyze the trade-off between level of service and operating costs of transit networks. *Transportation Research Part B: Methodological 70*, 35–46.

Ibarra-Rojas, O. J. and Y. A. Rios-Solis (2012). Synchronization of bus timetabling. *Transportation Research Part B: Methodological 46*(5), 599–614.

Klemt, W.-D. and W. Stemme (1988). *Schedule Synchronization for Public Transit Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Kliewer, N., T. Mellouli, and L. Suhl (2006). A time-space network based exact optimization model for multi-depot bus scheduling. *European Journal of Operational Research 175*(3), 1616–1627.

Kümmling, M., P. Großmann, K. Nachtigall, J. Opitz, and R. Weiß (2015). A state-of-the-art realization of cyclic railway timetable computation. *Public Transport 7*(3), 281–293.

Kroon, L., D. Huisman, and G. Maróti (2007). Railway Timetabling from an Operations Research Perspective. Technical report, Econometric Institute Report EI2007-22, Erasmus University Rotterdam, The Netherlands.

Kwan, C. M. and C. S. Chang (2008). Timetable synchronization of mass rapid transit system using multiobjective evolutionary approach. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews 38*(5), 636–648.

Laporte, G., F. A. Ortega, M. A. Pozo, and J. Puerto (2017). Multi-objective integration of timetables, vehicle schedules and user routings in a transit network. *Transportation Research Part B: Methodological 98*, 94–112.

Liebchen, C. and R. H. Möhring (2002). A Case Study in Periodic Timetabling. *Electronic Notes in Theoretical Computer Science 66*(6), 18–31.

Liebchen, C. and R. H. Möhring (2007). The modeling power of the periodic event scheduling problem: Railway timetables - and beyond. In *Algorithmic Methods for Railway Optimization, Lecture Notes in Computer Science*, Volume 4359, pp. 3–40.

Liu, T. and A. Ceder (2017). Integrated public transport timetable synchronization and vehicle scheduling with demand assignment: A bi-objective bi-level model using deficit function approach. *Transportation Research Part B: Methodological*.

Liu, T. and A. A. Ceder (2016). Synchronization of public transport timetabling with multiple vehicle types. *Transportation Research Record 2539*(2539), 84–102.

Liu, Z. and J. Shen (2007). Regional Bus Operation Bi-level Programming Model Integrating Timetabling and Vehicle Scheduling. *Systems Engineering - Theory & Practice 27*(11), 135–141.

Liu, Z., J. Shen, H. Wang, and W. Yang (2007). Regional Bus Timetabling Model with Synchronization. *Journal of Transportation Systems Engineering and Information Technology 7*(C), 109–112.

Nachtigall, K. (1999). *Periodic network optimization and fixed interval timetables.* Ph. D. thesis, Deutsches Zentrum für Luft- und Raumfahrt, Braunschweig, Germany.

Peeters, L. (2003). *Cyclic Railway Timetable Optimization.* Ph. D. thesis, Erasmus University of Rotterdam, Rotterdam, the Netherlands.

Pepin, A.-S., G. Desaulniers, A. Hertz, and D. Huisman (2009). A comparison of five heuristics for the multiple depot vehicle scheduling problem. *Journal of Scheduling 12*(1), 17–30.

Petersen, H. L., A. Larsen, O. B. G. Madsen, B. Petersen, and S. Ropke (2013). The simultaneous vehicle scheduling and passenger service problem. *Transportation Science 47*(4), 603 – 616.

Schröder, M. and I. Solchenbach (2006). Optimization of Transfer Quality in Regional Public Transit. Technical report, Berichte des Fraunhofer Instituts for Techno-und Wirtschafts.

Serafini, P. and W. Ukovich (1989). A mathematical model for periodic scheduling problems. *Siam Journal on Discrete Mathematics 2*(4), 550–581.

Shafahi, Y. and A. Khani (2010). A practical model for transfer optimization in a transit network: Model formulations and solutions. *Transportation Research Part A-policy and Practice 44*(6), 377–389.

Van den Heuvel, a. P. R. ., J. M. van den Akker, and M. E. van Kooten Niekerk (2008). Integrating timetabling and vehicle scheduling in public bus transportation. Technical report, Erasmus University Rotterdam, The Netherlands.

Wong, R. C. W., T. W. Y. Yuen, K. W. Fung, and J. M. Y. Leung (2008). Optimizing Timetable Synchronization for Rail Mass Transit. *Transportation Science 42*(1), 57–69.

Wu, Y., J. Tang, Y. Yu, and Z. Pan (2015). A stochastic optimization model for transit network timetable design to mitigate the randomness of traveling time by adding slack time. *Transportation Research Part C-emerging Technologies 52*, 15–31.

Wu, Y., H. Yang, J. Tang, and Y. Yu (2016). Multi-objective re-synchronizing of bus timetable: Model, complexity and solution. *Transportation Research Part C-emerging Technologies 67*, 149–168.