TEL AVIV UNIVERSITY

The Iby and Aladar Fleischman Faculty of Engineering
The Zandman-Slaner School of Graduate Studies

The Mixed Transit Fleet
Bus Scheduling Problem

A thesis submitted toward the degree of
Master of Science in Industrial Engineering

by

Yuval Elbar
January 2016
The Mixed Transit Fleet
Bus Scheduling Problem

A thesis submitted toward the degree of
Master of Science in Industrial Engineering

by

Yuval Elbar

This research was carried out in the Department of
Industrial Engineering under the supervision of
Dr. Tal Raviv

January 2016
Acknowledgments

My deepest gratitude to my advisor, Tal, who guided me through this labyrinth of research and never hesitated to share his extensive knowledge; who willingly sat with me long hours for joint work and had endless patience for my inquiries; who was very considerate and understanding during tough times; and who has taught me so much the last two years. Thank you Tal!

To my best friends, the Amigos, thank you for helping me see the funny side of tough times and many moments of laughter, and for your uncompromising support. You guys rock.

Also a warm thanks to the scholarship program of the Salti Foundation which awarded this research with financial support.

Last but not least, a huge thank you to my cheeky monkey, that created the loving, cozy, stable environment that made the ripening of this research possible, and whom I'm so grateful for.
Abstract

Diesel buses are the most prevailing type of buses in public transit fleets around the world. There are several negative side effects that accompany their use: noise nuisance, black soot, and toxic and greenhouse gas emissions. These toxic gases play a significant role in increased mortality rates due to air pollution. During the last decade, new types of buses were presented, the latest of which is a pure electric, battery-run bus with zero emissions and much lower operational costs compared to diesel buses. In cities all around the world, this type of bus is gradually being incorporated into existing fleets. Due to its high purchase price, the long life cycle of diesel buses, and conservatism and risk aversion, the transition into electric fleets will be gradual and will take years if not decades. As a result, public transit operators will have to handle a mixed fleet of both diesel and electric buses. With the current battery technology, electric buses still have an effective range limitation of up to a few hundred kilometers, depending on the model. This needs to be taken into consideration when planning the daily schedule for electric buses. In this study we present a new vehicle scheduling problem that incorporates mixed-fleet, multi-depot, range limitation and cyclic-schedule characteristics. A mixed integer linear programming model is formulated for this problem; its objective is to minimize the total cost. The costs in the model may include externalities, meaning the quantification of the negative effect on public health and the environment, caused by the diesel buses. As a result, a good solution to the problem is one that maximizes the benefit, or the saving, obtained by employing the electric buses, thus maximizing their utilization.

An iterative 2-step math heuristic method is presented for solving real-life large-scale instances of the problem such as the ones transit operators in major cities around the world deal with. Our heuristic decomposes the problem into smaller, easier to solve sub-problems, one for each bus type that has range limitation. The first step of the algorithm delivers a tight lower bound to the scheduling problem. The results of a numerical study that we conducted, using real data from the cities Be’er Sheva and Tel Aviv in Israel, are presented. Our heuristic obtained very low optimality gaps and high saving rates for mixed fleets with up to 25-30 electric buses. These results are compared to a greedy heuristic that imitates a possible strategy of a human scheduler. In all the compared instances that were not beyond the solvable size for our algorithm, our algorithm significantly outperformed the greedy heuristic.
## Contents

1. Introduction ....................................................................................................................... 1
2. Literature Review .............................................................................................................. 3
3. Problem Description ......................................................................................................... 9
4. Methodology ...................................................................................................................... 14
   4.1. The Iterative 2-Step Heuristic .................................................................................. 14
5. Numerical Experiments .................................................................................................... 19
   5.1. Extracting Instance Data ......................................................................................... 19
   5.2. Implementation .......................................................................................................... 21
   5.3. Evaluating Results .................................................................................................... 21
   5.4. Results ....................................................................................................................... 22
   5.5. Comparison to a “manual scheduler” greedy heuristic ............................................. 28
6. Conclusion ........................................................................................................................ 31
Bibliography ........................................................................................................................ 32
Table of Figures

Figure 1: the underlying network (step 1) .................................................................................. 10
Figure 2: the underlying network (step 2) .................................................................................. 10
Figure 3: complete underlying network .................................................................................... 11
Figure 4: a possible schedule on the network .......................................................................... 11
Figure 5: flow chart for the 2-step heuristic ............................................................................. 15
Figure 6: total cost vs. number of electric buses in the Metrodan system .......................... 27
Figure 7: total cost vs. number of electric buses in the Dan system ........................................ 27
Tables of tables

Table 1: VSP literature review summary................................................................. 8
Table 2: a summary of the 2-step heuristic............................................................. 18
Table 3: characteristics of the two case-study systems.......................................... 23
Table 4: numerical results for Metrodan operator .................................................. 24
Table 5: numerical results for Dan operator ............................................................ 25
Table 6: the 2-step algorithm vs. the greedy heuristic, for Metrodan operator .......... 29
Table 7: the 2-step algorithm vs. the greedy heuristic, for Dan operator .................. 29
1. Introduction

Today, in most cities, buses run on diesel engines which emit toxic gases and soot and create noise nuisance. These negative side-effects have a significant impact on densely populated areas, including a substantial rate of illness and mortality due to respiratory diseases each year. In 2010, the World Health Organization estimated the total number of deaths globally associated with air pollution to be 7 million – one in eight of the total global deaths (World Health Organization, 2014). Moreover, the emission of greenhouse gases from public transit contributes to global climate changes. The increasing awareness of these shortfalls pushed policy makers and transit operators over the last decade to introduce buses that run on other types of energy – natural gas, bio-diesel, hybrid-electric, etc. Recently, due to the technological evolution of high-capacity batteries for electric vehicles, manufacturers have come up with pure electric battery-powered buses with range limitation that is nearly sufficient for the entire working day of an urban bus. These buses are energetically efficient and produce no emissions. Public transit operators around the world have started purchasing and combining these buses in their existing fleets. For example, Dan, the largest bus operator in the greater Tel Aviv area, has declared its plan to acquire 5 electric buses this year after a trial period of 2 years with an electric bus it bought from BYD, a Chinese based electric bus manufacturer (Dan, 2015).

Electric buses are characterized by greater (about twice) purchase cost compared to the current diesel-powered ones, but have lower operating costs, thus making their purchase potentially profitable in the long run. However, there is a high risk involved in adapting new technologies and it requires investment in new facilities and equipment. Moreover, the operational lifespan of diesel buses is 12-15 years. Therefore, the transition to electric buses is expected to occur gradually. For these reasons, many operators will own mixed bus fleets, consisting of both legacy diesel buses and new electric buses. Electric and diesel buses have significantly different operational cost structures and range limitations. In particular, the operational range of electric buses is limited and thus, charging or battery-swapping operations should be included in their schedule. Therefore, for an extended period, there will be a need to plan the transit system operation of fleets that contain at least two types of buses – one of them with a limited range of total distance traveled.

Given this state of a mixed fleet, with both high emissions and zero-emissions buses, a question arises as to how to integrate the electric buses into the system so as to maximize their benefits. That is, which journeys should be served by electric buses in order to minimize the total pollution and noise impact on the population of the city? This impact is affected both by the amount of emissions as well as the number of people exposed to it, so location in the urban area and time of day are essential factors.
A public transit operational planning process includes four basic components usually performed in sequence (Ceder 2002): (1) **Network route design** – line planning. A line is a sequence of fixed location bus-stops at which the bus traveling that line lets passengers off and on the bus. Every line starts and ends at a terminal, which are usually large stations that function as an origin or destination for multiple bus lines. (2) **Setting timetables** – determining the departure times of each line in each direction. Each departure is a single journey that ends upon arrival to the ending terminal. This timetable must be adhered to without missing a single journey. (3) **Scheduling vehicles to journeys** – assigning vehicles to each journey in the timetable. The sequence of journeys throughout the day of a single vehicle is called a "chain" and may include "deadheading" journeys. These journeys without passengers on board may occur when a bus is assigned to a journey that starts in a different terminal than the end terminal of the previous journey. For this kind of assignment to be feasible, the starting time of the later journey must be greater than the ending time of the earlier journey plus the deadheading time between the two relevant terminals. (4) **Assignment of drivers** – arranging the workforce according to the timetable. While the first steps are typically the responsibility of the authorities (transportation ministry and transportation department of the city), the later steps are carried out by the operator.

The assignment of vehicles to tasks under some objective function is related to step (3) above, and is known as the Vehicle Scheduling Problem (VSP) or, sometimes, the Bus Scheduling Problem. In this study, we introduce an extension of this problem that considers a mixed fleet of buses with different range limitations and cost structures. We also allow several depots in the bus network and enforce cyclicity of the schedule. We refer to our extended problem as the Mixed-Fleet, Multi-Depot Cyclic Vehicle Scheduling Problem with Route Distance Constraints (MMC-VSP-RDC).
2. Literature Review

The Vehicle Scheduling Problem (VSP) originated in the 1960's as a special case of the Vehicle Routing Problem (VRP) with fixed delivery times. In 1981 an international workshop concerning the VSP was held, where it was well defined and discussed extensively. The outcome of this discussion is summarized in Bodin et al. (1981). They outline the differing and shared characteristics of these two problems and define the VSP as a special case of the VRP. The simplest version of the VSP is defined as follows:

Given (1) a set of journeys characterized by departure and arrival terminals, starting and ending times, and (2) the deadheading time between each pair of terminals, find the minimal number of vehicles to serve all journeys.

A slightly different version adds the costs of deadheading and minimizes the total cost, with a possible combination of first minimizing fleet size and then secondly the cost. These versions, called the Single-Depot VSP (SD-VSP), can be solved as a minimum cost flow problem on a network in which each journey is a node, and an arc connects two nodes if the same vehicle can perform them sequentially; This means that the starting time of the second journey has to be greater than the ending time of the first plus the deadheading time from the arrival terminal of the first journey to the departure terminal of the second. This network flow problem can be solved in polynomial time.

A more complex version is the Multi-Depot VSP (MD-VSP), in which the vehicles can be dispatched from numerous depots and each vehicle must return to its starting depot at the end of the planning horizon. This adds a new dimension of assigning vehicles to depots, which makes the problem NP-hard. Two basic heuristic approaches are suggested: (1) the first, "cluster first, route second", clusters journeys together and assigns each cluster to a depot. Then for each depot, a single depot VSP is solved for obtaining the minimum number of vehicles to serve these journeys from that specific depot. (2) The second approach, "schedule first, cluster second", firstly solves the scheduling problem disregarding the depots completely, and obtains the minimum number of vehicles to serve all journeys. Secondly, the vehicles are assigned to depots as to minimize the deadheading cost. The second phase can be seen as a simple transportation problem.

Two more NP-hard extensions of VSP are mentioned: (1) one is the VSP with length of path restriction (VSP-LPR) in which the maximal time a vehicle can spend out of the depot or the total distance traveled by a vehicle is limited. It can also be found under the name VSP-RDC, standing for "route distance constraint". (2) The second includes multiple vehicle types and named VSP-MVT. Commonly, it refers to vehicles with different capacities for passengers, which means the predicted demand for each journey has to be taken into account.
Freling et al. (2001) present a few new algorithms and improvements for existing algorithms for solving the SD-VSP. Recall that this problem can be solved in polynomial time using a network flow algorithm, so the goal of this study is improving computational times. The authors point out the importance of short computation time especially when solving this problem as part of a wider problem, such as combined vehicle and crew scheduling. Their algorithms include: a combined backward-forward auction algorithm, the "schedule first cluster second" approach, and a problem-size reduction technique that deletes journey-connecting arcs of the underlying network when existent arcs that represent passage through the depot can be used instead. Numerical experiments on both generated and real data of up to 1500 trips and about 370 vehicles show that no specific method dominates the others, but there is a consistent improvement from past performance regarding the computation time.

Ceder (2010) used a deficit-function based heuristic to minimize the number of vehicles in the SD-VSP-MVT. Vehicle types differ in cost and the set of journeys compatible with them, in terms of vehicle capacity. An important feature of this study is the flexibility of departure times, instead of fixed, unchangeable times; departures may be early or late a few minutes. The author points out that the graphic nature of this method is an important feature that contributes to the understanding of the scheduling process by the human operators, thus enabling them to make changes if needed.

While studies dealing with the polynomial time solvable SD-VSP focus on improving efficiency and reducing computational time, the NP-hard MD-VSP studies' focus is obtaining a good solution. First for small-scale instances and later, for real-life sized instances.

Carpaneto et al. (1989) devised a Branch & Bound algorithm for the MD-VSP. Several lower bounds are suggested, e.g. one based on a relaxation that allows a bus to return to a different depot than the one it started at. Their algorithm solves randomly generated instances of up to 3 depots and 60 journeys, and serves as a baseline for later exact solution methods for solving MD-VSP.

Forbes et al. (1994) present a new multi-commodity network flow formulation to MD-VSP, and an exact algorithm that utilizes it. In the first stage, the problem is relaxed into a quasi-assignment problem, by allowing a vehicle to finish its duty in a different depot than the one it departed from. This is an easy network-flow problem. In the second stage, dual simplex is used to solve the linear relaxation of MD-VSP, with the solution from stage one as an initial solution. Finally, a branch & bound procedure finds an integer solution, in case some variables were assigned non-integer values at the previous step.
Numerical experiments on both randomly generated and real problems with 3 depots and 600 trips are solved to optimality.

Dell’Amico et al. (1993) present a new heuristic for MD-VSP and test it on self-generated instances of up to 10 depots and 1000 trips. Their heuristic creates a chain of trips at each step by finding a shortest path on a network in which nodes represent journeys, and forbidding arcs that would lead to a solution with more than the minimal number of vehicles. It performs better than previous known heuristics and produces small optimality gaps with short computation times, guaranteeing minimal fleet size.

Lobel (1998, 1999) uses a column-generation technique called "Lagrangian pricing" to solve instances of up to 25,000 trips and 50 depots with optimality gaps of less than 0.5%. His method is based on two Lagrangian relaxations: The first deals with flow conservation and the second with flow condition – allowing uncovered trips. Another attribute of his model is a type of mixed fleet, in the sense that not all vehicles are allowed to serve all trips.

Haghai and Banihashemi (2002) present a new formulation of MD-VSP, based on the Forbes et al. (1994) formulation. They add a set of "transshipment" depot nodes that enable buses to wait at the depot between consequent journeys. They assume that the cost of waiting at a depot is zero, unlike waiting at the street (layover). The number of variables in this new formulation is smaller by 40%, compared with the original formulation. Using a commercial solver they solve instances of up to 800 journeys.

Into this formulation they introduce route time constraints to present a compact formulation for the MD-VSP-RTC. One exact and two heuristic procedures are presented to solve the problem: (1) the exact method iteratively solves the MD-VSP and adds constraints to eliminate duties that exceed the time limitation. It solved randomly generated instances of up to 400 trips and showed how adding a few route time constraints makes the problem much harder to solve. (2) The first heuristic method has similar structure to the exact method, but at each iteration it cuts journeys from the end of illegal blocks until they satisfy the route time constraint. Then these blocks are added to the solution as constraints. With this method they solve instances of up to 400 trips with small optimality gaps of up to 0.5%. (3) The second heuristic is similar but uses the LP solver and not the integer programming (IP) module. This solves up to 500 trips in 900 minutes with 0.3% optimality gap.

For dealing with larger instances, they offer two ways for reducing the size of problems:
1. Connecting journeys that do not require deadheading when the layover time is less than a certain constant.
2. Eliminating the 80% most expensive deadheading options.
In a real case study they conducted on the Baltimore network (4 depots, 5650 trips) their heuristic obtained Improvement of 5.77% in total cost.

Kliewer et al. (2006) present a new time-space network formulation for MD-VSP that reduces the amount of variables compared with earlier formulations. The reduction in the number of variables is obtained by aggregating connection arcs together by the use of waiting arcs. Higher waiting cost is given for a layover (waiting at a station between trips) than for returning to a depot and waiting there. Their graph contains a layer for each vehicle type in order to control capacity. The largest instance, consisting of some 7000 trips and 5 depots, is solved using CPLEX 8.0 in around 3 hours.

Wang and Shen (2007) is the first study motivated by the emergence of electric buses. They consider MD-VSP with limited route distance and recharging time. They formulate an M.I.P model and solve the problem heuristically in two stages: first an ant colony algorithm is applied to construct feasible blocks and then a maximum matching problem on a bi-partite graph is used to assign the blocks to the minimum number of vehicles. The heuristic is tested on a small real-world instance of 3 depots and 276 journeys.

In Li and Head (2009), a multi depot bus scheduling model is used to show how purchasing different types of buses can reduce emissions with a slight increase in operation costs. Evaluation of emissions is by total length traveled. A time-space network is applied and real instances from Tucson, Arizona with up to 200 journeys are solved in several hours and have optimality gaps of less than 10%. The model enables the purchasing of new buses with a dedicated budget and limits the total emissions of the fleet. An elastic approach is taken towards the emissions and budget constraints, with penalties for exceeding the original limitation.

Wei et al. (2010) also use ant colony optimization to solve MD-VSP-RTC; however, they incorporate a mixed fleet feature in their model. Vehicles differ in capital costs, their route time limitation, and refueling time. The algorithm is applied on a small instance of up to 220 journeys with 3 depots and 25 vehicles of 2 types.

Li (2013) addresses two variations of the SD-VSP in the context of an electric bus fleet: (1) enabling battery swapping and (2) bus range limitation. A linear programming model based on a journey-indexed network is presented and solved, using CPLEX, for up to 500-journey instances. For larger instances the author presents a branch and price heuristic based on the Dantzig-Wolfe decomposition and column generation (C.G). It solves large instances of 947 journeys with optimality gaps of up to 1%. Optimality gaps refer to a Lagrangian lower bound used in the C.G based heuristic. Results were obtained using real-life instances or variations of them. Deadheading distances and times were estimated by using Dijkstra's algorithm for finding the shortest
path between junctions on the area's road geometry network provided by NavTeq, and an estimated average speed of 40 km/h. Operational costs are estimated based on energy consumption data. Electric bus fleet size is 10 to 140 vehicles, and the range limit per vehicle used is 120-150 km. The largest instance took 86 hours to solve.

In this work we study the Mixed-Fleet, Multi-Depot Cyclic Vehicle Scheduling Problem with Route Distance Constraints (MMC-VSP-RDC). The input for the extended problem also includes the information about the bus fleet, i.e., how many vehicles of each type are available with their operational costs and driving ranges. The objective is to find a feasible daily schedule that covers all the journeys at minimal overall cost while satisfying route distance and cyclicity constraints. The model also decides upon the number and types of buses that are located initially at each depot subject to the fleet size constraint. Cyclicity implies that the numbers and types of buses located at each depot at the end of the day are equal to their numbers and types at the beginning of the day. The main contribution of this study is: (1) presenting a rich model that captures all the aspects of the problem faced by operators who use a mixed fleet of buses with range-limited vehicles; (2) presenting a heuristic solution method that is capable of solving large real-life instances of this problem.

In Table 1, the features of the models in the above-mentioned studies are summarized and compared to those of the current model. Each row in the table refers to a certain study, the author(s) of which and the year of publishing are written in the first column. The second and third columns state whether the study deals with a single-depot or multi-depot problem, and if depots are capacitated or not. The fourth column states whether the fleet is homogenous or mixed. If the study includes route distance or route time constraints, is written in the fifth column. Method of solving is stated in the sixth column. The seventh column states if the study uses real data instances in their experiments, and written in the eighth column is whether the instances tested are large-scale or not. The type of the underlying network of the model is referred to in the ninth column, and in the tenth column any special features of the model are written.
Table 1: VSP literature review summary

<table>
<thead>
<tr>
<th>Study</th>
<th>Depots</th>
<th>Depot capacity</th>
<th>Fleet</th>
<th>RTDC</th>
<th>Method</th>
<th>Real data</th>
<th>Large scale</th>
<th>Network</th>
<th>Special features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wang &amp; Shen 2007</td>
<td>multiple</td>
<td>yes</td>
<td>Homogenous</td>
<td>distance</td>
<td>Ant Colony, Matching</td>
<td>yes</td>
<td>no</td>
<td>Bi-partite</td>
<td>Refueling</td>
</tr>
<tr>
<td>Wei et al. 2010</td>
<td>multiple</td>
<td>yes</td>
<td>mixed</td>
<td>time</td>
<td>Ant Colony</td>
<td>no</td>
<td>no</td>
<td>-</td>
<td>Refueling</td>
</tr>
<tr>
<td>Li 2013</td>
<td>single</td>
<td>yes</td>
<td>homogenous</td>
<td>distance</td>
<td>Column Generation, local search</td>
<td>yes</td>
<td>no</td>
<td>Journey-based</td>
<td>Refueling</td>
</tr>
<tr>
<td>Kiewer et al. 2006</td>
<td>multiple</td>
<td>yes</td>
<td>mixed</td>
<td>no</td>
<td>IP</td>
<td>yes</td>
<td>yes</td>
<td>Multi-layer time-space</td>
<td>-</td>
</tr>
<tr>
<td>Freling et al. 2009</td>
<td>single</td>
<td>no</td>
<td>homogenous</td>
<td>no</td>
<td>assignment</td>
<td>yes</td>
<td>no</td>
<td>Journey-based</td>
<td>-</td>
</tr>
<tr>
<td>Ceder</td>
<td>single</td>
<td>no</td>
<td>mixed</td>
<td>no</td>
<td>Deficit function</td>
<td>yes</td>
<td>no</td>
<td>Journey-based</td>
<td>Flexible schedule</td>
</tr>
<tr>
<td>Carpaneto et al. 1989</td>
<td>multiple</td>
<td>yes</td>
<td>homogenous</td>
<td>no</td>
<td>Branch &amp; Bound</td>
<td>yes</td>
<td>no</td>
<td>Journey-based</td>
<td>-</td>
</tr>
<tr>
<td>Forbes et al. 1994</td>
<td>multiple</td>
<td>yes</td>
<td>homogenous</td>
<td>no</td>
<td>QAP, MCNF, B&amp;B</td>
<td>yes</td>
<td>no</td>
<td>Journey-based</td>
<td>-</td>
</tr>
<tr>
<td>Dell’Amico et al. 1993</td>
<td>multiple</td>
<td>yes</td>
<td>homogenous</td>
<td>no</td>
<td>Shortest path based</td>
<td>no</td>
<td>no</td>
<td>Journey-based</td>
<td>-</td>
</tr>
<tr>
<td>Lobel 1998</td>
<td>multiple</td>
<td>yes</td>
<td>mixed</td>
<td>no</td>
<td>Column Generation</td>
<td>yes</td>
<td>yes</td>
<td>Journey-based</td>
<td>-</td>
</tr>
<tr>
<td>Haghani &amp; Banihashemi 2002</td>
<td>multiple</td>
<td>yes</td>
<td>homogenous</td>
<td>time</td>
<td>Constraint generation</td>
<td>yes</td>
<td>yes</td>
<td>Journey-based</td>
<td>Problem reducing techniques</td>
</tr>
<tr>
<td>Current study</td>
<td>multiple</td>
<td>Aggregative</td>
<td>mixed</td>
<td>distance</td>
<td>3-step heuristic</td>
<td>yes</td>
<td>yes</td>
<td>Multi-layer time-space</td>
<td>Aggregative cyclicity</td>
</tr>
</tbody>
</table>
3. Problem Description

In this chapter we provide a formal model for the MMC-VSP-RDC and introduce the required notation. Recall that VSPs are solved after the bus line planning and timetabling are done, and are part of the input for the VSP. The input for the MMC-VSP-RDC also includes the information about the bus fleet, i.e., how many vehicles of each type are available with their operational costs and driving ranges. The objective is to find a feasible daily schedule that covers all the journeys at minimal overall cost while satisfying route distance and cyclicity constraints. The model also decides upon the number and types of buses that are located initially at each depot subject to the fleet size constraint. Cyclicity implies that the numbers and types of buses located at each depot at the end of the day are equal to their numbers and types at the beginning of the day.

The problem can be summarized as follows.

Input
1. Set of bus types, for each:
   - quantity available in the fleet
   - range limitation
2. Set of terminals and depots:
   - capacity limitation (for buses of all types)
   - deadheading time for each pair
   - deadheading distance for each pair
3. Set of journeys, for each:
   - origin and destination terminals
   - departure and arrival times
   - distance
4. Tuples of journeys and bus types:
   - cost, including externalities for each tuple
5. Tuples of terminals and bus types:
   - capacity of the terminal for each bus type (this bus type particular limitation may occur due to availability and compatibility of charging facilities)
6. Tuples of terminal pairs and bus types
   - deadheading cost for each tuple

Output: assignment of vehicles to journeys and deadheading tasks

Objective: minimize total cost
Graphic representation of the problem
Next we introduce a useful graphical representation by a network that is based on a time-space diagram. The horizontal axis is time units and the vertical represents the different terminals (space). On this diagram we first plot the journeys by placing nodes at their start and finish coordinates and connect each pair by an arc. See Figure 1 for an illustration. In this illustration, departure nodes are colored green while arrival nodes are colored red. All together we will call them "event nodes", event being a departure or arrival.

![Figure 1: the underlying network (step 1)](image1)

Next, on top of the journey arcs, we add deadheading arcs which enable a bus that finished a journey to take another journey that begins in a different terminal or depot than the one it is currently at (depicted as yellow arcs in Figure 2). This is possible only if the arrival time of the first journey plus the deadheading time to the next journey's departure terminal is smaller or equal to the departure time of the second journey. In addition, we want to enable deadheading as the last ride of the day, as a preparation for the next day. For this cause we add to each terminal one start node and one end nodes (depicted as grey squares in Figure 2). In this example we assume that driving from terminal 1 to terminal 2 at this hour of the day takes more than 1 unit of time, therefore no deadheading arc is stretched from (2,1) to (3,2).

![Figure 2: the underlying network (step 2)](image2)
Another type of arcs we add is waiting arcs. A bus may wait in its current location and take a later journey that departs from there (a layover). One last arc type is the circulation arcs that are used for connecting the end and start nodes of each terminal. See Figure 3. Note that waiting arcs can make some deadheading arcs redundant. In our illustration it occurs with the deadheading arc that connects (4,3) to Terminal 2 end node. Since this trip can be done earlier via the arc \{(4,3),(5,2)\} and then the arcs that connects (5,2) to the end node of Terminal 2. Therefore we can remove this deadheading arc without affecting the value of the optimal solution.

A possible schedule is highlighted on this network as illustrated in Figure 4. The orange path describes the itinerary of one bus and the green of another. In this example the paths are disjointed, but it may happen that more than one bus uses the same deadheading or waiting arcs.
This network with appropriate lower and upper bounds on the flows can be used to solve the underlying VSP (assuming single vehicle type and cyclicity as defined above) as a min-cost flow circulation problem, which can be solved by a known polynomial time algorithm. The flow on the journey arcs must be one and there are no flow constraints on all the other arcs. A cost equal to the deadheading distance is associated with each of the deadheading arcs. The net demand of all the nodes is zero. Any feasible flow in this network can be decomposed into paths of single unit flow which represent a schedule of a single bus.

Note that in the classical MD-VSP each bus must return to its initial depot. This requirement is slightly stronger than our cyclicity requirement and result in an NP-Hard problem, as proven by Lenstra and Kan (1981). Moreover, introduction of a route distance constraint is also enough to make the problem NP-hard (Bodin 1981).

In order to represent the extended MMC-VSP-RDC we define a new graph which is created by duplicating the original graph, one copy (layer) for each bus. With this representation it is possible to introduce side constraints on the various layers that enforce the route distance limitation. The flow costs in each layer may represent different types of vehicles with different cost structures.

Referring the above network, we can formulate an Integer Programming model of our problem. Here we first present the notation, then the model.

**Notation:**

- $A$: Set of all possible trips (timetabled journeys and deadheading)
- $J$: Set of journeys. $J \subset A$
- $T$: Set of types of vehicles (e.g., diesel, electric, etc.)
- $B$: Set of individual vehicles (buses)
- $B_t$: Set of all individual vehicles from type $t$. $B_t \subset B$
- $N$: Set of events (arrival / departure)
- $R$: Set of events that represent start and end terminals
- $R_{beg}$: Set of events that represent start terminals
- $\delta_{in}(j,b)$: Set of incoming arcs to the node that represent event $j$ at the layer of bus $b$
- $\delta_{out}(j,b)$: Set of outgoing arcs from the node that represent event $j$ at the layer of bus $b$
\( d_a \)  Distance of trip \( a \)  \\
\( r_b \)  Range limitation for bus \( b \)  \\
\( c_{ab} \)  Cost of trip \( a \) for bus \( b \)  \\
\( x_{ab} \)  Decision variable that denotes the flow value on the arc of trip \( a \) by bus \( b \)  

Model:

\[
\min \sum_{a \in A} \sum_{b \in B} c_{ab} x_{ab} \quad (1)
\]

\[
\sum_{b \in B} x_{ab} = 1 \quad \forall a \in J \quad (2)
\]

\[
\sum_{a \in \delta_{\text{in}}(j,b)} x_{ab} = \sum_{a \in \delta_{\text{out}}(j,b)} x_{ab} \quad \forall j \in N, b \in B \quad (3)
\]

\[
\sum_{b \in B} \sum_{a \in \delta_{\text{in}}(j,b)} x_{ab} = \sum_{b \in B} \sum_{a \in \delta_{\text{out}}(j,b)} x_{ab} \quad \forall j \in R, t \in T \quad (4)
\]

\[
\sum_{j \in R_{\text{beg}}} \sum_{a \in \delta_{\text{out}}(j,b)} x_{ab} \leq 1 \quad \forall b \in B \quad (5)
\]

\[
\sum_{a \in A} d_a x_{ab} \leq r_b \quad \forall b \in B \quad (6)
\]

\[
x_{ab} \geq 0, \quad \text{integer} \quad \forall a \in A, b \in B \quad (7)
\]

Objective function (1) minimizes the overall cost of the arcs traveled by all of the buses. Constraints (2) state that each journey is assigned to exactly one vehicle. Constraints (3) are flow conservation at the event nodes for each bus (graph layer), and (4) stipulate cyclicity of the schedule – the number of buses that depart from each terminal in the beginning of the day is equal to the number arriving at the end of the day – for each bus type. Constraints (5) assure that no bus is used more than once, and (6) are route distance constraint for each vehicle. In (7) we define the decision variables as integer non-negatives.
4. Methodology

As explained in the previous section, the MMC-VSP-RDC is NP-hard, thus even by using a state of the art solver, the above integer programming (IP) model can be solved only for small instances and not for real-life sized problems of urban transit systems. To deal with the latter, we present in this section an iterative 2-step math heuristic that decompose the problem into sub problems that are easier to solve. The first step of this heuristic also provides a lower bound to the original problem that allows us to evaluate the quality of the obtained solution.

Our solution method is based on the assumption that the bus type which is most expensive to operate (including externalities) has no effective range limitation. This is indeed the case as long as some diesel buses are still used.

4.1. The Iterative 2-Step Heuristic

This heuristic is composed of two different IP models performed iteratively, each model taking input from the previous one. As described in the previous section, in order to enforce the route distance constraint, there is a need for a layer of the underlying graph for each individual bus. This significantly grows the number of variables and makes the problem hard to solve. Our heuristic uses a relaxation of this constraint in order to split the problem into separated sub-problems, one for each vehicle type, which reduces the size of the problem into a much more solvable size.

In the first step we solve a relaxation of the problem, in which each of the remaining bus types is represented by a single layer of the network and the route distance constraint is enforced aggregately on all the vehicles of each such layer. In the second step the schedule of the buses from the type with the lowest operational and external costs of all the remaining types is constructed. The schedule covers a subset of the journeys that were allocated to it in the first step, ideally all of them. This division into sub-problems reduces the dimension of the problem significantly. The scheduled bus type and journeys are removed from the problem and the process is repeated until all the types are scheduled. When all the remaining bus types are ones without an effective range limitation (e.g., diesel buses) the schedule can be directly derived from the optimal solution of the first step’s model and the algorithm is terminated. Recall that we assume that at least the last bus type that is considered indeed has no effective range limitation.

Note that the objective function value of the first step model in the first iteration is a lower bound to the optimal solution of the actual problem.
A flow chart that summarizes and illustrates the 2-step heuristic is shown in Figure 5.

**Figure 5: flow chart for the 2-step heuristic**

**Step 1 – relaxation of the route distance constraint**
The range limitation per vehicle is replaced with a total range limitation for all the vehicles of the same type combined. The new limitation is the sum of individual ranges of the vehicles of that certain type. That is, the range for that type, multiplied by the number of vehicles available of that type. Therefore, we formulate the problem using a network with a single layer for each vehicle type instead of a layer for each individual bus. This significantly reduces the dimension of the problem and results in an IP model that can be solved to optimality even for large networks. The relaxed problem is solved by the following model.

**Additional Notation**

- $c_{at}$: Cost of trip $a$ for bus type $t$
- $n_t$: Number of available buses of type $t$
- $r_t$: Range limitation for bus of type $t$
δ_{in}(j,t)  Set of incoming arcs to the node that represent event \( j \) at the layer of bus type \( t \)

δ_{out}(j,t)  Set of outgoing arcs from the node that represent event \( j \) at the layer of bus type \( t \)

\( x_{at} \)  Decision variable that denotes the total flow on the arc of trip \( a \) by all buses of type \( t \)

The model solved in Step 1 is:

\[
\begin{align*}
\min & \sum_{a \in A} \sum_{t \in T} c_{at} x_{at} \\
\text{s.t.} & \sum_{t \in T} x_{at} = 1 \quad \forall a \in J \tag{9} \\
& \sum_{a \in \delta_{in}(j,t)} x_{at} = \sum_{a \in \delta_{out}(j,t)} x_{at} \quad \forall j \in N \cup R, t \in T \tag{10} \\
& \sum_{j \in R_{beg}} \sum_{a \in \delta_{out}(j,t)} x_{at} \leq n_{t} \quad \forall t \in T \tag{11} \\
& \sum_{a \in A} d_{at} x_{at} \leq n_{t} r_{t} \quad \forall t \in T \tag{12} \\
& x_{at} \geq 0, \quad \text{Integer} \quad \forall a \in A, t \in T \tag{13}
\end{align*}
\]

The objective function (8) minimizes the total cost over all the arcs traveled by all the bus types. Constraints (9) state that each journey is assigned to exactly one vehicle (of one type, of course). Flow conservation for each node is assured by constraints (10), this stipulate consistency of the schedule and cyclicity at the same time. In (11), the sum of vehicles used of each type is limited by this type’s availability. Constraints (12) limit the total distance traveled by all the vehicles together for each bus type. Constraints (13) define the decision variable as non-negative integers.

Any feasible solution of the original IP model (1)-(7) can be mapped to a feasible solution of the relaxed model (8)-(13) but not vice versa. Thus, the solution of this model (assuming all bus-types and all journeys are used) is a lower bound for the MMC-VSP-RDC.

**Step 2 – feasible assignment for a single bus type**

In this step we solve the scheduling problem for a single bus type by selecting a subset of the journeys that were allocated to this type at Step 1. Covering all of these journeys with the same bus type may not be feasible due to the re-enforcement of the range constraint in
Step 2, which was relaxed in Step 1. Therefore instead of trying to minimize the costs while covering all these journeys, we maximize the operational saving obtained by using the current bus type. The saving of each journey is calculated by subtracting the operational and external cost of the currently considered type from the costs of the most expensive type. Model (14)-(20) solves this problem. We denote the current vehicle type by $t$. The set of buses $B$ is redefined to include only the buses of type $t$ and the set of journeys $J$ includes now only the ones allocated to this type at Step 1.

The model solved in Step 2 is:

$$\text{max} \sum_{b \in B} \sum_{a \in J} (c_{a,t} - c_{at})x_{ab}$$

(14)  

$$\sum_{b \in B} x_{ab} \leq 1 \quad \forall a \in J$$

(15)  

$$\sum_{a \in \delta_{in}(j,b)} x_{ab} = \sum_{a \in \delta_{out}(j,b)} x_{ab} \quad \forall j \in N, b \in B$$

(16)  

$$\sum_{b \in B} \sum_{a \in \delta_{in}(j,b)} x_{ab} = \sum_{b \in B} \sum_{a \in \delta_{out}(j,b)} x_{ab} \quad \forall j \in R$$

(17)  

$$\sum_{a \in A} d_{a} x_{ab} \leq r_{t} \quad \forall b \in B$$

(18)  

$$\sum_{j \in R_{beg}} \sum_{a \in \delta_{out}(j,b)} x_{ab} \leq 1 \quad \forall a \in A, b \in B$$

(19)  

$$x_{ab} \geq 0, \quad \text{Integer} \quad \forall a \in A, b \in B$$

(20)

Objective function (14) maximizes the total cost of the journeys assigned. Constraints (15) make sure that each journey is assigned at most once. Constraints (16) are flow conservation equations that enforce consistency of the schedule, and (17) stipulate its cyclicity – the number of buses that depart from each terminal in the beginning of the day equals the number arriving at the end of the day. Constraints (18) limit the distance traveled by each bus, and (19) assure that no bus is used more than once. In (20) we demand non-negativity and integrality of the decision variables.

The outcome of this step is a feasible assignment for (possibly all of the) buses of type $t$ in the fleet.
A summary of the 2-step heuristic is presented in Table 2.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Journey set</th>
<th>Fleet</th>
<th>Network layers</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Relaxation of range constraint</td>
<td>All remaining</td>
<td>All remaining</td>
<td>Remaining number of bus types</td>
<td>Journey set for Step 2. Lower bound at the first iteration</td>
</tr>
<tr>
<td>2</td>
<td>Maximize utilization / saving</td>
<td>Allocated in Step 1</td>
<td>Buses of type $t$</td>
<td>Number of buses of type $t$</td>
<td>Feasible assignment of bus type $t$</td>
</tr>
</tbody>
</table>
5. Numerical Experiments

In this section we present the results of a numerical study that is based on applying our method for two study cases of large bus operators in Israel. We examined the performance of the algorithm and the implications of introducing electric buses into the currently pure diesel fleets. Thus, in our experiments we have two types of buses, legacy diesel buses with unlimited range and electric buses with range limits of 200-300 km.

5.1. Extracting Instance Data

We created instances of our problem using timetable data that was made available by the Israeli Ministry of Transport and Road Safety in standard format called “General Transit Feed Specification” (GTFS). The files and a description of the structure of this format can be downloaded from Ministry of Transport and Road Safety (2015). These files include information about the public transportation in the entire country, including the operators, stop locations, timetables, distances etc. In addition, we use google map API to obtain time and distance of potential deadheading trips. We have run the algorithm on two real life sets of data from two cities in Israel: Be'er Sheva (operated by Metrodan Public Transport Ltd.), and Tel Aviv (operated by Dan Public Transportation Ltd.). These are the actual timetables and routes as of March 2015.

Data extraction was done in the following way.

1. **List of journeys:**
   The complete journey set from the GTFS files is filtered by a few criteria:
   - a specific public transport operator
   - a specific day of the week – Monday, which represents a routine working day
   - no school buses or other transit jobs not available for all the citizens, or seems irregular

   The attributes each journey is characterized by are:
   - starting and ending terminals
   - departure and arrival times
   - total distance

2. **List of terminals:** stations that journeys begin or terminate at.

   This initial list of terminals is further processed: nearby terminals are clustered together, with radius of about 200 meters. Terminals are characterized by their location, in terms of latitude and longitude.
3. Deadheading distances and times:
To determine the driving distance and time between any pair of terminals, we used google maps queries. For each pair of terminals, we used the latitude and longitude coordinates from the GTFS files to calculate the driving distance and time, for each direction. With this data we created deadheading distance matrix and deadheading time matrix.

4. Operational and external costs:
One of the purposes of this work is to provide an evaluation or decision tool regarding the economic and environmental benefits of introducing electric buses into an existing fleet. This is done by including externalities in the costs, on top of the operational costs. Both cost types can differ greatly from one urban transit system to another – depending on local regulations, labor costs, vehicle types and condition, population density, traffic congestion and other factors. In our experiments, we approximate the costs to be proportional to the duration of the ride, following the logic that the longer the ride the more pollution is emitted. Thus, rides that are short in distance but long in duration are likely to benefit the most from zero emission vehicles. Indeed, these rides are more likely to take place in densely populated areas where more people are exposed to the emissions.

In our experiments, the cost of operating a diesel bus per time unit is set to be double than the cost of operating an electric bus.

We fixed the cost per engine hour of the diesel buses to 120 NIS (about 30 US$) and of the electric buses to 60 NIS (about 15 US$). These costs include direct operational cost (e.g., energy, payment to the drivers, etc.) and externalities (e.g., the negative effect of air pollution, etc.). The actual costs may vary depending on the economic conditions in each market but a cost ratio of 2:1 between diesel and electric buses is a very conservative estimation in favor of the diesel. In fact, some electric bus manufacturers (Proterra, 2015) claim that this ratio holds for the operational cost only, even without consideration of the externalities.

In addition, we set the fixed cost of using a bus at a certain day to 100 NIS, which represents the effort needed to prepare the bus and clean it at the end of the day. Our model does not consider the capital cost of the buses but such cost can easily be introduced into it.

When applying to model in real life setting we recommend performing a more accurate estimation of the externalities based on GIS data about population density and human activity in the different regions of the city. This is out of the scope of this research. Our model assumes obtaining the total cost of each journey as an input.

The processed data used in our experiments is available for download from ADD.
5.2. Implementation

The framework of the 2-step heuristic is implemented in Python, and the IP models were formulated in OPL and solved by CPLEX 12.6.1. The workstation used for the experiment was Intel i7-2600 3.4 GHz with 16 GB Ram under Windows 7, 64 bit. The Python framework automates several components:

1. **Preprocessing**: extracting instance data from GTFS files and Google Map API.
2. **Heuristic frame**: creating input files for the OPL models and running the CPLEX model, iteratively.
3. **Post processing**: creating schedules from the results of the models and calculating their objective value.

5.3. Evaluating Results

In order to evaluate the quality of the solution obtained from our algorithm we calculated several measures for each of the solved instances, as follows.

**Lower bound** (**LB**): the total cost of the first iteration of Step 1. Recall that the problem solved in this iteration includes the entire fleet, covers all journeys but relaxes the range constraint.

**Solution value** (**V**): The total cost of our algorithm's solution. Note that the value is not obtained directly from the solution of the mathematical models but is re-calculated from the schedule delivered by the algorithm at the various iterations of Step 2 and the last one of Step 1.

**Optimality gap** \(\frac{V-LB}{LB}\): A gap larger than 0% may occur for two reasons, or the combination of both:

1. The assignment of journeys to vehicles at Step 1 is infeasible when the route distance constraint is re-enforced. Therefore the optimal solution of MMC-VSP-RDC is necessarily greater than \(LB\).
2. The optimization models at some iteration are not solved to optimality due to computational resources limitations. We note that in our experiments, Step 1 could always be easily solved in a few minutes while in Step 2 we had to compromise on suboptimal solutions due to time limitation of 12 hours that we decided upon.

**Upper bound** (**UB**): The total cost when not using electric buses at all, no saving possible. This is calculated using the same model of Step 1, including all journeys and only diesel buses available, without any electric buses. This serves as a reference value.
that represents the minimal possible cost in the current state, before introducing the electric buses into the fleet.

**Actual saving** \((UB - V)\): The costs saved by using the electric buses, using the schedule obtained from our algorithm. This equals the difference between the upper bound to our solution's cost.

**Potential saving** \((UB - LB)\): The gap between the total cost without electric buses at all, and the lower bound on the solution with electric buses. This represents an upper bound on the potential saving due to the introduction of electric buses.

**Relative saving** \(\left(\frac{UB - V}{UB - LB}\right)\): The actual saving divided by the potential saving. Note that this is a normalized quality measure for the solution obtained by the algorithm, and represents the quality of the solution better. Indeed, when the electric buses constitute a small fraction of the fleet, the relative difference between the solution value and the lower bound (i.e., optimality gap) must be small regardless of the quality of the solution. The Relative saving represents the share of the saving out of an upper bound on the potential saving. Thus, it is not sensitive to the number of electric buses in the fleet. The closer the relative saving is to 100% the better the solution is.

### 5.4. Results

For each of the two systems we created several instances by using different values for the following two parameters:

1. **Electric bus range**: the range values we used represent the current technology according to electric buses manufacturers. We tested the values 200km and 300km.
2. **Number of electric buses**: here too, the values we used represent real life situations that transit operators are dealing with at present, due to the gradual transition to electric buses, as mentioned in the introduction section. We used the values 5, 10, 15, 20, 25 and 30 buses.

Using the combinations of these parameters, we created 12 instances for each of the two systems. That is, 24 problem instances in total. In Table 3, we present the characteristics of current state in the two systems based on the timetable data that we extracted. These include the name of the operator and the city it is operating at, in the first two columns. The number of timetabled journeys and the number of terminals appear on the next two columns. The size of the diesel fleet, in the fifth column, was deduced from the solution and it is consistent with information that is available from the operators. In the last column, the total operational and external costs are calculated using our Step 1 model with diesel buses only. For the instances with electric buses, we did not change the
number of diesel buses and thus the diesel fleet size is never a binding constraint for these instances.

Table 3: characteristics of the two case-study systems

<table>
<thead>
<tr>
<th>Operator</th>
<th>City</th>
<th>Journeys</th>
<th>Terminals</th>
<th>Diesel Fleet</th>
<th>Cost (UB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metrodan</td>
<td>Be'er Sheva</td>
<td>2,444</td>
<td>17</td>
<td>100</td>
<td>142,662</td>
</tr>
<tr>
<td>Dan</td>
<td>Tel Aviv</td>
<td>12,627</td>
<td>50</td>
<td>950</td>
<td>1,371,080</td>
</tr>
</tbody>
</table>

The results of the numerical study are presented in Table 4 for Metrodan operator and Table 5 for Dan operator. In both tables, each row presents the results for a different instance. The instances differ in the number of electric buses (column A) and in their range limitation (column B), as appears in the first two columns of each row. It took our algorithm two iterations to solve each one of the instances, since they include only two types of buses: diesel buses, and electric buses with an effective range limitation. Columns C to H include the results of the first iteration, columns I to J the results of the second iterations (which includes only the first step since the diesel bus has no range limitation). Columns K to M present some overall statistics.

The first two columns for iteration 1 describe Step 1. Column C is the objective value, which is also a lower bound to the problem as explained in section 5.3. In column D the number of journeys allocated to electric buses is presented. Recall that this allocation is done by a model that relaxes the route distance constraint.

Columns E to H belong to Step 2 of iteration 1. Column E is the IP optimality gap as stated by the solver at the end of the 12-hour run of this model, and column F is the objective value obtained. Column G is the number of journeys that were not assigned to electric buses out of the subset allocated to electric buses in Step 1. Column H is the utilization of the electric buses, which is the total distance traveled by electric buses divided by the total distance made available by them.

The next two columns present the second iteration, in which only the diesel buses remain, therefore only Step 1 is performed. Column I is the objective function value and column J is the number of diesel buses used in the solution.

In the last three columns, we present and measure the solution for the entire scheduling problem that includes both types of buses. Column K is the total cost obtained by our
algorithm, in column L the relative saving is presented, and in column M the overall optimality gap of the algorithm. The calculation methods of the values in the last two columns are explained in Section 5.3 above.
Table 4: numerical results for Metrodan operator

<table>
<thead>
<tr>
<th>Metrodan</th>
<th>Iteration 1</th>
<th>Iteration 2</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance parameters</td>
<td>Step 1</td>
<td>Step 2</td>
<td>Step 1</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>Num. of e-buses</td>
<td>e-bus range</td>
<td>LB</td>
<td>Num. of allocated journeys</td>
</tr>
<tr>
<td>5</td>
<td>200</td>
<td>139,363</td>
<td>160</td>
</tr>
<tr>
<td>10</td>
<td>200</td>
<td>136,054</td>
<td>305</td>
</tr>
<tr>
<td>15</td>
<td>200</td>
<td>132,766</td>
<td>445</td>
</tr>
<tr>
<td>20</td>
<td>200</td>
<td>129,551</td>
<td>571</td>
</tr>
<tr>
<td>25</td>
<td>200</td>
<td>126,370</td>
<td>708</td>
</tr>
<tr>
<td>30</td>
<td>200</td>
<td>123,219</td>
<td>839</td>
</tr>
<tr>
<td>5</td>
<td>300</td>
<td>137,780</td>
<td>209</td>
</tr>
<tr>
<td>10</td>
<td>300</td>
<td>132,934</td>
<td>407</td>
</tr>
<tr>
<td>15</td>
<td>300</td>
<td>128,151</td>
<td>590</td>
</tr>
<tr>
<td>20</td>
<td>300</td>
<td>123,427</td>
<td>780</td>
</tr>
<tr>
<td>25</td>
<td>300</td>
<td>118,779</td>
<td>969</td>
</tr>
<tr>
<td>30</td>
<td>300</td>
<td>114,205</td>
<td>1153</td>
</tr>
</tbody>
</table>
Table 5: numerical results for Dan operator

<table>
<thead>
<tr>
<th>Num. of e-buses</th>
<th>e-bus range</th>
<th>LB</th>
<th>Num. of allocated journeys</th>
<th>IP gap</th>
<th>Num. of unassigned journeys</th>
<th>Range utilization</th>
<th>Objective function value</th>
<th>Num. of buses</th>
<th>Solution total cost</th>
<th>Relative saving</th>
<th>Optimality gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>200</td>
<td>1,367,336</td>
<td>136</td>
<td>0.07%</td>
<td>4,261</td>
<td>2</td>
<td>1,363,182</td>
<td>931</td>
<td>1,367,443</td>
<td>97.1%</td>
<td>0.01%</td>
</tr>
<tr>
<td>10</td>
<td>200</td>
<td>1,363,739</td>
<td>264</td>
<td>1.38%</td>
<td>8,333</td>
<td>4</td>
<td>1,355,616</td>
<td>927</td>
<td>1,363,949</td>
<td>97.1%</td>
<td>0.02%</td>
</tr>
<tr>
<td>15</td>
<td>200</td>
<td>1,360,039</td>
<td>340</td>
<td>2.44%</td>
<td>12,486</td>
<td>7</td>
<td>1,348,224</td>
<td>923</td>
<td>1,360,710</td>
<td>93.9%</td>
<td>0.05%</td>
</tr>
<tr>
<td>20</td>
<td>200</td>
<td>1,356,576</td>
<td>446</td>
<td>2.61%</td>
<td>16,339</td>
<td>11</td>
<td>1,340,914</td>
<td>917</td>
<td>1,357,253</td>
<td>95.3%</td>
<td>0.17%</td>
</tr>
<tr>
<td>25</td>
<td>200</td>
<td>1,352,946</td>
<td>556</td>
<td>11.34%</td>
<td>19,107</td>
<td>61</td>
<td>1,336,422</td>
<td>915</td>
<td>1,355,529</td>
<td>85.8%</td>
<td>0.19%</td>
</tr>
<tr>
<td>30</td>
<td>200</td>
<td>1,349,380</td>
<td>660</td>
<td>135.13%</td>
<td>11,072</td>
<td>396</td>
<td>1,351,494</td>
<td>925</td>
<td>1,362,566</td>
<td>39.2%</td>
<td>0.98%</td>
</tr>
<tr>
<td>5</td>
<td>300</td>
<td>1,365,720</td>
<td>140</td>
<td>0.42%</td>
<td>5,883</td>
<td>1</td>
<td>1,359,894</td>
<td>931</td>
<td>1,365,777</td>
<td>98.9%</td>
<td>0.00%</td>
</tr>
<tr>
<td>10</td>
<td>300</td>
<td>1,360,355</td>
<td>295</td>
<td>0.92%</td>
<td>11,692</td>
<td>3</td>
<td>1,348,854</td>
<td>926</td>
<td>1,360,546</td>
<td>98.2%</td>
<td>0.01%</td>
</tr>
<tr>
<td>15</td>
<td>300</td>
<td>1,354,974</td>
<td>457</td>
<td>2.24%</td>
<td>17,400</td>
<td>9</td>
<td>1,338,196</td>
<td>921</td>
<td>1,355,596</td>
<td>96.1%</td>
<td>0.05%</td>
</tr>
<tr>
<td>20</td>
<td>300</td>
<td>1,349,683</td>
<td>607</td>
<td>7.52%</td>
<td>22,449</td>
<td>38</td>
<td>1,329,642</td>
<td>917</td>
<td>1,352,091</td>
<td>88.7%</td>
<td>0.18%</td>
</tr>
<tr>
<td>25</td>
<td>300</td>
<td>1,344,416</td>
<td>751</td>
<td>9.35%</td>
<td>27,731</td>
<td>61</td>
<td>1,320,022</td>
<td>912</td>
<td>1,347,753</td>
<td>87.5%</td>
<td>0.25%</td>
</tr>
<tr>
<td>30</td>
<td>300</td>
<td>1,339,123</td>
<td>886</td>
<td>53%</td>
<td>23,734</td>
<td>300</td>
<td>1,327,252</td>
<td>916</td>
<td>1,350,986</td>
<td>62.9%</td>
<td>0.89%</td>
</tr>
</tbody>
</table>
The experiment demonstrates that our algorithm produces near optimal solutions when applied to a large diesel-based system, extended with some 20-30 range-limited electric buses. Beyond 25 or 30 buses, a near optimal solution of Step 2 is impossible to obtain using our integer programming formulation, given the hardware and the time budget in our experiment. For such larger instances, the schedule of the electric buses constructed in Step 2 is inferior and thus the overall solution exhibits relatively large optimality gaps and small relative saving.

The range of the electric buses has a mixed effect on the computational effort that is required to solve Step 2. On one hand, the larger the range, the larger the subset of journeys allocated to electric buses on step 1. This increases the dimension, thus the complexity, of Step 2. On the other hand, the complexity of the vehicle-scheduling problem stems from an effective range limitation, therefore a large range will be less binding and the complexity of Step 2 will decrease. In practice, increased range limitation tended to decrease the optimality gap at Step 2 for the smaller instances (with less buses or journeys), but not so for the larger ones.

The effect of the network's size on the solution process is not straightforward. Note that the Dan system is more than five times larger in terms of journeys and three times larger in terms terminals compared to the Metrodan system. However, the intricate part of our algorithm is Step 2, where the model is solved only for a subset of journeys allocated for range-limited buses at Step 1, along with the corresponding possible deadheading trips. Since the size of this set is mainly affected by the number of electric buses and their range, the effort required for solving the model of Step 2 for the Dan instances is not significantly greater than the effort required for the Metrodan instances. Therefore, the relative saving obtained for the Dan instances is only slightly smaller in most of the instances. Interestingly, the overall optimality gap for the Dan instances is not consistently larger than the one for the Metrodan but this only demonstrates the need for our relative saving measure. Indeed, since in Dan the relative effect of a few tens of electric buses on a 950 bus fleet is much minor than its effect on the 100 bus fleet of Metrodan.

Based on the results of our experiment we can cautiously state that replacing some 25-30% of the buses in metropolitan transit systems to electric buses may result in high economic and environmental benefits. However, this conclusion should be verified by applying our model to more accurate estimations of its inputs.

In Figure 6 and Figure 7, we present the correlation between the total cost and the number of electric buses introduced into the fleet, for both Metrodan and Dan systems, respectively. Only the instances solved to near-optimality in the 12-hour time budget are presented.
We can see in both systems that within this range of electric fleet size, the decrease in total cost is roughly linear with the increase in number of electric buses, as long as Step 2 is solved to near-optimality. Also, the decrease is steeper when the range of electric buses is larger (300 km). This is because each bus can perform more journeys per day and save more costs.

It may be that for a greater number of buses, the marginal contribution of each unit will be reduced. That is, the slope of the graph will become less steep since electric buses will already be assigned to the journeys with the highest saving potential.
5.5. Comparison to a "manual scheduler" greedy heuristic

When observing the optimal solutions of the pure diesel fleet cases of our test instances, we noted that many bus schedules (chains) already satisfy the range limitation of the electric buses – even those with 200km a day range limit. This brought up the idea to develop a very fast heuristic that imitates what a manual scheduler might have done. In this heuristic method, we start with the optimal (or existing) schedule of a pure diesel fleet. Electric buses are assigned to replace the chains of individual diesel buses that satisfy the range limitation. The replaced chains are greedily selected according to the potential saving that may be obtained from replacing them with electric buses. In this section, we compare the results of our 2-step algorithm with the outcome of such a fast heuristic.

Note that the solution of Step 1 may be broken into chains in more than one way, meaning that different schedules with different chain lengths may be constructed, but all of them share the same total cost. In the greedy heuristic, we used the simplest rule for breaking down the solution into journey chains: whenever there are multiple options for selecting the arc for the next step in the chain, we choose the arc with the lowest index.

In Table 6, we compare the results of our algorithm with the greedy heuristic for the system of Metrodan, and in Table 7 for the system of Dan. The first two columns define the instance – the number of electric buses and the range limitation respectively. In the third and fourth columns, the absolute savings obtained by our 2-step algorithm and by the greedy heuristic are presented. In fifth and sixth columns, the corresponding relative savings are presented (these are the same figures presented in Table 4 and Table 5). Recall that the definition of relative saving can be found in section 5.3. The rightmost column presents the improvements of relative savings gained by the 2-step algorithm over the greedy heuristic. That is, the difference between the fifth and sixth columns.
Table 6: the 2-step algorithm vs. the greedy heuristic, for Metrodan operator

<table>
<thead>
<tr>
<th>Number of electric buses</th>
<th>Electric bus range</th>
<th>2-step heuristic saving</th>
<th>Greedy heuristic saving</th>
<th>2-step heuristic relative saving</th>
<th>Greedy heuristic relative saving</th>
<th>2-step excess relative saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>200</td>
<td>3,239</td>
<td>2,830</td>
<td>98.2%</td>
<td>85.78%</td>
<td>12.4%</td>
</tr>
<tr>
<td>10</td>
<td>200</td>
<td>6,452</td>
<td>5,471</td>
<td>97.6%</td>
<td>82.79%</td>
<td>14.8%</td>
</tr>
<tr>
<td>15</td>
<td>200</td>
<td>9,768</td>
<td>7,542</td>
<td>98.7%</td>
<td>76.21%</td>
<td>22.5%</td>
</tr>
<tr>
<td>20</td>
<td>200</td>
<td>12,622</td>
<td>8,894</td>
<td>96.3%</td>
<td>67.84%</td>
<td>28.5%</td>
</tr>
<tr>
<td>25</td>
<td>200</td>
<td>15,592</td>
<td>9,653</td>
<td>95.7%</td>
<td>59.25%</td>
<td>36.5%</td>
</tr>
<tr>
<td>30</td>
<td>200</td>
<td>18,756</td>
<td>9,653</td>
<td>96.5%</td>
<td>49.65%</td>
<td>46.9%</td>
</tr>
<tr>
<td>5</td>
<td>300</td>
<td>4,871</td>
<td>4,522</td>
<td>99.8%</td>
<td>92.63%</td>
<td>7.2%</td>
</tr>
<tr>
<td>10</td>
<td>300</td>
<td>9,709</td>
<td>8,921</td>
<td>99.8%</td>
<td>91.70%</td>
<td>8.1%</td>
</tr>
<tr>
<td>15</td>
<td>300</td>
<td>14,292</td>
<td>13,102</td>
<td>98.5%</td>
<td>90.29%</td>
<td>8.2%</td>
</tr>
<tr>
<td>20</td>
<td>300</td>
<td>18,977</td>
<td>17,101</td>
<td>98.7%</td>
<td>88.91%</td>
<td>9.8%</td>
</tr>
<tr>
<td>25</td>
<td>300</td>
<td>23,510</td>
<td>20,952</td>
<td>98.4%</td>
<td>87.73%</td>
<td>10.7%</td>
</tr>
<tr>
<td>30</td>
<td>300</td>
<td>11,897</td>
<td>24,639</td>
<td>41.8%</td>
<td>86.58%</td>
<td>-44.8%</td>
</tr>
</tbody>
</table>

Table 7: the 2-step algorithm vs. the greedy heuristic, for Dan operator

<table>
<thead>
<tr>
<th>Number of electric buses</th>
<th>Electric bus range</th>
<th>2-step heuristic saving</th>
<th>Greedy heuristic saving</th>
<th>2-step heuristic relative saving</th>
<th>Greedy heuristic relative saving</th>
<th>2-step excess relative saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>200</td>
<td>3,637</td>
<td>3,248</td>
<td>97.1%</td>
<td>86.75%</td>
<td>10.3%</td>
</tr>
<tr>
<td>10</td>
<td>200</td>
<td>7,131</td>
<td>6,434</td>
<td>97.1%</td>
<td>87.64%</td>
<td>9.5%</td>
</tr>
<tr>
<td>15</td>
<td>200</td>
<td>10,370</td>
<td>9,535</td>
<td>93.9%</td>
<td>86.36%</td>
<td>7.5%</td>
</tr>
<tr>
<td>20</td>
<td>200</td>
<td>12,168</td>
<td>12,581</td>
<td>83.9%</td>
<td>86.74%</td>
<td>-2.8%</td>
</tr>
<tr>
<td>25</td>
<td>200</td>
<td>15,551</td>
<td>15,572</td>
<td>85.8%</td>
<td>85.87%</td>
<td>-0.1%</td>
</tr>
<tr>
<td>5</td>
<td>300</td>
<td>5,303</td>
<td>4,895</td>
<td>98.9%</td>
<td>91.32%</td>
<td>7.6%</td>
</tr>
<tr>
<td>10</td>
<td>300</td>
<td>10,534</td>
<td>9,717</td>
<td>98.2%</td>
<td>90.60%</td>
<td>7.6%</td>
</tr>
<tr>
<td>15</td>
<td>300</td>
<td>15,484</td>
<td>14,471</td>
<td>96.1%</td>
<td>89.85%</td>
<td>6.3%</td>
</tr>
<tr>
<td>20</td>
<td>300</td>
<td>18,989</td>
<td>19,178</td>
<td>88.7%</td>
<td>89.63%</td>
<td>-0.9%</td>
</tr>
<tr>
<td>25</td>
<td>300</td>
<td>23,327</td>
<td>23,865</td>
<td>87.5%</td>
<td>89.50%</td>
<td>-2.0%</td>
</tr>
</tbody>
</table>
The first conclusion from the results presented in Table 6 and Table 7 is that in all cases where we are able to solve Step 2 to near optimality, our algorithm delivers significantly better solutions than the greedy heuristic. This justifies the additional computational effort of the algorithm presented here. It also calls for devising better solution methods for the sub problem of Step 2, which can be either exact methods or heuristic ones.

As expected, in the solutions obtained by the greedy heuristic method, the absolute savings increase with the number of electric buses but the marginal contribution of each additional bus is decreasing. The same phenomenon is observed in the results of the 2-step algorithm when applied to the smaller Metrodan instances. However, the marginal contribution of the electric buses does not consistently decrease in the solutions of the larger Dan instances. This can be explained by the fact that we are unable to solve these instances close enough to optimality. The rate of decrease in the marginal contribution of electric buses to the savings is much smaller in the Dan instances. This can be explained by the fact that the number of journeys (and thus possible chains) from which the ones assigned to electric buses are selected is much larger in the Dan instances.
6. Conclusion

We have presented a new model for the Mixed Fleet, Multi-Depot, Cyclic Vehicle Scheduling Problem with Route Distance Constraint, named MMC-VSP-RDC. This combination of characteristics has not been dealt with previously. An iterative 2-step math heuristic is presented for solving real-life large-scale instances of this problem. We have conducted numerical experiments based on timetables of two real urban transit systems, one with about 100 buses and the other with about 950 buses. The algorithm performed well and achieved optimality gaps of no more than 0.3% when the number of range-limited buses is up to 25-30 buses.

By incorporating external costs into the model, our algorithm may serve as a decision tool that demonstrates the benefit of introducing some electric buses into existing fleets and as a scheduling tool, once such buses are introduced.

The first step of our algorithm establishes a lower bound on the total cost of operating a mixed fleet. At the second step, the actual schedule (chains) of the range-limited buses is constructed. It is shown numerically that the total cost of the optimal schedule is very close to the lower bound. This implies that for strategic planning ends, the model solved in the first step provides sufficiently accurate insights. This observation is useful because the first model can be solved in a very short time independently of the number of range-limited buses.

The challenge for future research is to devise better solution methods for Step 2, in order to allow scheduling of fleets with greater number of range-limited buses. One possible direction can be to divide these buses into several subsets of say, 25 buses each, and then apply our algorithm with a single iteration for each subset of buses. That is, the range-limited buses can be artificially divided into several types. In fact, this division is not necessarily artificial since the fleet may have several types of range-limited buses with different cost structures and ranges. Other heuristic methods, not necessarily based on mathematical programming, may be used to solve the model of Step 2.
Bibliography


The most common propulsion technology for buses is diesel. The operation of these buses involves a number of unwanted byproducts: they emit harmful gases, greenhouse gases, are a source of noise pollution, and leave a mark on the environment. Atmospheric pollution is a leading cause of death, accounting for one in every eight deaths worldwide, and public transportation contributes significantly to this pollution.

In recent decades, several types of buses have been developed to address these issues: natural gas, bio-diesel, and electrically powered buses. All of these improve but do not eliminate the drawbacks of diesel.

As a result of the highlighted disadvantages of diesel buses and due to technological advances in recent years, an electric bus has been developed entirely on a battery-only system and is marketed by various manufacturers worldwide.

These electric buses are not polluting, do not irritate, and have lower operating costs than diesel buses. However, the initial cost of purchase is significantly higher than that of diesel buses and requires an investment in infrastructure and training of the crew.

Additionally, in existing technology, electric buses have a restricted range of up to 200-300 km between charges.

Due to the high initial cost, the long life of diesel buses, their safety and prudence, the transition to an electric fleet is expected to be gradual and take years.

During this period, public transportation operators will have to manage a mixed fleet, some of which have a limited range. Efficient use of this fleet will maximize the use of electric buses to reduce pollution in crowded urban centers.

In this work, we define a new scheduling problem of multi-modal hybrid electric buses, taking into account a mixed fleet, limited operating range and the schedules of the buses. The objective function of this problem is to minimize costs, where these costs include pollution-causing losses. We define a linear programming model for this problem, but this model can only be used for small cases of the problem.

To solve the real cases of this problem, which result from managing a large fleet of public transportation in a large city, we developed a mathematical heuristic with two stages that are iteratively executed.

This algorithm breaks down the original problem into smaller, easier-to-solve sub-problems, one for each type of bus with a limited range.

The first stage of this algorithm provides a lower bound for the solution of the entire problem. We tested this algorithm on various cases of this problem, which we built from real-world schedules of the Danon Corporation in Tel Aviv and the Mota-Don Corporation in Be'er Sheva. For each of these schedules, we ran the algorithm on various cases of the number of electric buses and their limited range. In these experiments, we can see that for 25-30 electric buses, our algorithm achieves nearly optimal results, building schedules that achieve the greatest possible savings.

To check the computational effort of 12 hours we gave the algorithm to run, we compared this algorithm to a heuristic human-like approach. This heuristic does not build a new schedule but simply switches a contaminated bus to a clean bus for the schedules of the most contaminated buses. We demonstrated that for all cases that are not too large, our algorithm produces results that are significantly better than the heuristic approach.

The results of these experiments show that the conversion of public transportation to electric buses can significantly reduce pollution in urban areas, and that the cost-benefit ratio is in favor of electric buses.

A comprehensive report on this research and its implications is available.
structuring vehicles in a mixed fleet

This document was submitted as a thesis towards the degree of "MSc in Engineering" in
the Department of Industry Engineering, under the supervision of Dr. Tal Ribeh,
under the guidance of Mr. Yohai Elber.

Yohai Elber

The work was conducted in the Research Department of the Faculty of Engineering.
בעיית שיבוץ האוטובוסים בטח מזויף

הויבר והויבר נכובים גמר לקראת התואר "מוכתר אוניברסיטאי".

עינת-די

יובל אלבר

שנת החשון"א