Mathematical Models and Solution Algorithms for the Vehicle Routing Problem with Environmental Considerations

by

Ramin Raeesi

THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Centre for Transport and Logistics (CENTRAL)

Department of Management Science

Lancaster University Management School

Lancaster, UK

April 2019

This thesis is dedicated to Elham.

For her patience, unconditional love and support.

DECLARATION

This thesis is my original work and has not been submitted, in whole or in part, for a degree at this or any other university. Nor does it contain, to the best of my knowledge and belief, any material published or written by another person, except as acknowledged in the text.

Ramin Raeesi

June 2019

ABSTRACT

Urban freight distribution is essential for the functioning of urban economies. However, it is contributing significantly to problems such as traffic congestion and environmental pollution. The main goal of this research is to contribute to greening urban freight distribution by developing new mathematical models and solution algorithms pertaining to the two major steams in *Vehicle Routing Problems (VRPs)* with environmental considerations: (i) VRPs with an explicit fuel consumption estimation component as a proxy for emissions, and (ii) VRPs with vehicles in the fleet that run on a cleaner alternative fuel such as electricity.

In the first stream, this thesis develops and solves a new realistic multi-objective variant of the pollution-routing problem, referred to as the *Steiner Pollution-Routing Problem (SPRP)*, that is studied directly on the original urban roadway network. The proposed variant is capable of incorporating the real operating conditions of urban freight distribution, and striking a balance between traditional business and environmental objectives, while integrating all factors that have a major impact on fuel consumption, including the time-varying congestion speed, vehicle load, vehicle's physical and mechanical characteristics, and acceleration and deceleration rates. The thesis develops new combinatorial results that facilitate problem solution on the original roadway network and also introduces new mathematical models for synthesizing the expected second-by-second driving cycle of a vehicle over a given road-link at a given time of the day. New efficient multi-objective optimisation heuristics are also developed for addressing realistic instances of the SPRP.

On the other hand, in the latter stream discussed above, to tackle the significantly impeding problem of range anxiety in the face of goods distribution using *Electric Commercial Vehicles (ECVs)*, a paradigm shift in the routing of

ECVs is proposed by introducing the *Electric Vehicle Routing Problem with Synchronised Ambulant Battery Swapping/Recharging (EVRP-SABS)*. The proposed problem exploits new technological developments corresponding to the possibility of mobile battery swapping (or recharging) of ECVs using a *Battery Swapping Van (BSV)*. In the EVRP-SABS, routing takes place in two levels for the ECVs that carry out delivery tasks, and for the BSVs that provide the running ECVs with fully charged batteries on their route. There is, therefore, a need to establish temporal and spatial synchronisations between the vehicles in the two levels and to do so new combinatorial results and a new solution algorithm is proposed.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest appreciation to Prof. Konstantinos Zografos who provided me the opportunity to join Lancaster to work with him and never stopped supporting me. His amazing eye for detail, insightful comments and hard questions always contributed to the improvement of my work and I benefited immensely from his extensive experience and expertise. I have been so lucky to be one of his students and words are powerless to express my gratitude.

I should also thank several people within the Management Science department and CENTRAL, including Prof. Adam Letchford and Prof. Matthias Ehrgott with whom I had insightful discussions. I would also like to thank Ms. Gay Bentinck for being a great supporter of doctoral students. I also gratefully acknowledge the funding support I received from the Lancaster University Management School which made my PhD possible.

I had a great network of friends at Lancaster and to all I am greatly thankful; special mention goes to Fotis Katsigiannis, Sebastian Stauß, Ghassan Al Koureiti, Seyma Bekli, Anis Zebiri, Esra Paca, Amer Reza, Anh Vu, Sasan Barak and Masume Maghsudi.

I should give a dedicated mention to my best friend Tabish Umar Ansari, a great thinker and scientist with whom I shared my journey the most and had the best moments of my years in Lancaster. The sleepless nights of working together and discussing our research problems and our walks and hikes will be very much missed. I will be thankful to you eternally for your endless support.

Last but not the least, I would like to thank my family for supporting me throughout my life.

Contents

1. INTRODUCTION AND BACKGROUND

1.1 Motivation

Urban Freight Distribution (UFD) plays an indispensable role in transporting and delivering the consumer goods required to sustain more than half of the world's population that now live in urban areas (Ritchie and Roser, 2018). UFD is vital for economic and societal growth of cities, and is a key enabler of wider businesses. It is presently growing at a faster pace than ever, as e-commerce and small-package delivery by logistics giants such as DHL, UPS, FedEX, and DPD is becoming more and more widespread.

Alongside with the pivotal role that UFD is playing in the functioning of urban economies, however, it is generating significant externalities such as traffic congestion and environmental pollution. Freight vehicles typically represent 8% to 15% of total traffic flow in urban areas (MDS Transmodal, 2012) and are responsible for 25% of urban transport related CO_2 emissions and 30% to 50% of other transport related pollutants (e.g. Particulate Matter and Nitrogen Oxide) (Alice/Ertrac, 2015). In London in 2006, for example, of the 9.6 million tonnes of CO_2 emitted by all forms of transport, around 23% was from freight vehicles (Transport for London, 2007). UFD is also considered more polluting than long-distance freight transportation, as fuel consumption increases sharply due to the stop-and-go driving patterns in congested urban centres (MDS Transmodal, 2012). The problem is getting more compound as new trends suggest that the high competition in the parcel courier sector between multiple operators has led to higher numbers of part-loaded vehicles (generally medium-sized goods vehicles) that enter residential areas; in many cases if residents are out at work, deliveries by these vehicles cannot be completed and the goods have to be returned to the depot of the parcel courier and re-delivery must be arranged (MDS Transmodal, 2012).

This large and ever-increasing level of emissions from urban freight transport activities has attracted the attention of policy makers and national governments. The European Commission has, for instance, set a target for "essentially CO_2-free city logistics in urban centres by 2030" (European Commission, 2011). A recent survey that is aimed at assessing this target (Allen et al., 2017) reviews freight initiatives that are expected to reduce *Heavy Goods Vehicles (HGVs)* kilometres and CO_2 emissions in European urban areas and ranks vehicle routing and scheduling tools among the top 10 impactful initiatives, which can help achieving around 23% reduction in HGV vehicle kilometres by 2030. The same survey suggests also that the uptake of vehicles with zero local emissions (e.g. electric vehicles) is a top-ranking initiative that is expected to bring in a reduction of over 60% in emissions in urban centres by 2030.

It is now well-realised by academics and practitioners that the *Vehicle Routing Problem (VRP)* can play a significant role in greening the UFD while satisfying business objectives. The 2015 survey of the UK Department for Transport to capture data on current levels of uptake of fuel efficient technologies among HGV operators shows that 41% of the respondents are now using telematics to optimise their vehicle routing (Department for Transport, 2017). On par with the industry side, a considerable wave of academic work has appeared in the area of VRPs with environmental considerations in recent years. These comprise the seminal papers on the *Pollution Routing Problem (PRP)* (Bektaş & Laporte, 2011), the *Green Vehicle Routing Problem (G-VRP)* (Erdoğan & Miller-Hooks, 2012) and the *Electric Vehicle Routing Problem (EVRP)* (Schneider et al., 2014).

Despite the large number of research work that has appeared in a rather short time in the area of VRPs with environmental considerations, research in the field is still lagging behind in terms of providing a realistic exposition of the real operating conditions of urban freight distribution, and in incorporating the rapid transport related technological advancements into new ways of designing environment-friendly distribution routes. These technological developments are particularly relevant to the wide availability of real-time and historical traffic data from all across the roadway network that could be used in data-informed decision making, and the rapid developments that are taking place in the area of alternative fuel vehicles running on cleaner fuel types such as electricity, natural gas and bio-diesel.

In the current doctoral thesis, I have focused on the development of new mathematical models and solution algorithms that are pertinent to the two major streams leading to the green UFD via VRPs with environmental considerations: (i) minimising vehicle emissions by optimising factors that affect the overall amount of fuel consumed by a delivery route, and (ii) facilitating the conversion of the fleet to electric commercial vehicles that are characterised by zero local emissions through the exploitation of new technological developments.

1.2 The Vehicle Routing Problem

The VRP is one of the most studied operational research problems that is celebrating its 60th anniversary since its introduction in 1959. The seminal paper of Dantzig and Ramser (1959) on the "Truck Dispatching Problem" was concerned with a real-life application of designing delivery routes with minimum total distance for a homogenous fleet of trucks that deliver gasoline to service stations. In the basic variants of the VRP we are given a central depot, a set of customers with geographically dispersed locations and known demand sizes, a set of homogenous vehicles, and cost (e.g. distance or travel time) of travelling between every pair of customers and every customer and depot. The aim of the VRP is to determine a set of vehicle routes that begin and terminate at the depot while visiting every customer exactly once, such that the total cost is minimised.

Since its introduction by Dantzig and Ramser (1959), the VRP has attracted a lot of attention in the academic literature, due to its many applications in practice and its various theoretical challenges. Many different variants of the VRP have been introduced through years to incorporate real-life complexities, and many models and solution algorithms have been proposed for the problem. There are now several review papers and books available on the VRP (Braekers et al., 2016; Eksioglu et al., 2009; Golden et al., 2008; Toth & Vigo, 2014), and there are also many specialised review papers on its different variants such as the capacitated VRP (Laporte, 2009), VRP with time windows (Bräysy & Gendreau, 2005a; Bräysy & Gendreau, 2005b; Gendreau & Tarantilis, 2010), VRP with pickup and delivery (Berbeglia et al., 2007), VRP with split deliveries (Archetti & Speranza, 2012), the periodic VRP (Campbell & Wilson, 2014), dynamic VRP (Pillac et al., 2013), VRP with multiple depots (Montoya-Torres et al., 2015), green VRP (Bektaş et al., 2019; Demir et al., 2014; Lin et al., 2014), time-dependent VRP (Gendreau et al., 2015), multi-objective VRP (Jozefowiez et al., 2008), heterogeneous VRP (Koç et al., 2016), VRP on road networks (Ben Ticha et al., 2018) and synchronization aspects in VRP (Drexl, 2012).

In the rest of this section, a brief overview on the variants that are more pertinent to the current study is presented.

1.2.1 Emissions minimising VRP

Following the realisation of the significant role of the VRP in greening UFD, a considerable deal of research work has appeared in the literature in a rather short time that try to introduce pollution related objectives into traditional VRPs. In this class of the VRP, that could be categorised as *Emissions Minimising VRP (EMVRP)*, fuel consumption is usually used as a proxy for pollutants emissions, particularly CO_2 emissions which are proportional to the amount of fuel consumed.

Fuel consumption approximation, on the other hand, is a very complicated task as it is dependent on a number of factors including travel related factors (such as speed, acceleration rates, and driving pattern), vehicle related factors (such as engine size, fuel type, payload, and age of the vehicle), road related factors (such as gradients, roundabouts, and traffic lights), and meteorological conditions (such as ambient temperature, wind speed and direction) (Palmer, 2007). Existing emission models could be broadly categorised into two main groups of (i) macroscopic (average speed) models, which make estimations based on the trip-based average speed, and (ii) microscopic (instantaneous) models, which are based on instantaneous vehicle kinematic variables, such as speed and acceleration (Demir, Bektaş, & Laporte, 2014b). It is well-known that compared with macroscopic models, microscopic emission and fuel consumption models provide much more accurate estimations (Zegeye et al., 2013; Ahn & Rakha, 2008; Boulter et al., 2006; Wang et al., 2011). This is because macroscopic models are unable to consider technical and vehicle-specific parameters such as vehicle shape (e.g. frontal area), and road conditions (e.g., gradient, surface resistance) (Demir, Bektaş, & Laporte, 2012). Therefore, microscopic models seem more robust, reliable and more applicable in the area of optimization (Demir, Bektaş, & Laporte, 2014b). Among all the available models, the *Comprehensive Modal Emissions Model (CMEM)* of Barth et al. (2004) can be viewed as a state-of-the-art microscopic emission model that has received far more attention in the EMVRP literature due to its ease of application (Demir, Bektaş, & Laporte, 2014) and its capability to embrace time-varying traffic conditions, vehicle payload, and certain vehicle's physical and mechanical characteristics.

Existing emissions minimising vehicle routing models have so far considered several of the major factors that have an impact on the fuel consumption of a truck, such as vehicle type, speed, and load. Some of these models only incorporate the effect of the load carried by the vehicle on the fuel consumption level of routes

(Kara et al., 2007; Ubeda et al., 2011; Xiao et al., 2012). The main bearing on the emissions level in this category of models is due to the sequence that customers are visited which affects the payload between consecutive visits. More sophisticated models recognise the major role of the vehicle speed over each road-link in addition to the vehicle load. In the PRP (Bektaş & Laporte, 2011) and several related papers (Demir et al., 2012; Kramer et al., 2015b) fuel consumption is assumed a nonlinear convex function of the vehicle speed, and hence speed optimisation in a time-independent setting, where non-congested traffic conditions are assumed throughout the day throughout the network, is attempted. However, the acknowledgement of the fact that travel speed could not be freely chosen in congested urban areas, as it fully depends on the expected time-varying traffic conditions, has led to the consideration of time-dependent variants of the VRP for a more accurate estimation model of fuel and other relevant decisions with temporal dependencies (Androutsopoulos & Zografos, 2017; Ehmke et al., 2016; M. Figliozzi, 2010; M. A. Figliozzi, 2011; Franceschetti et al., 2013). Some of these studies also consider the possible benefits of waiting at the depot and/or the customers (Androutsopoulos & Zografos, 2017; Franceschetti et al., 2013; Xiao et al., 2012).

The effect of the type and the number of the trucks that are included in the fleet to execute the routes on fuel consumption was previously considered in the context of the EMVRPs by Koç et al. (2014) and Xiao and Konak (2016). The studies of Demir, Bektaş, and Laporte (2014) and Androutsopoulos and Zografos (2017) are the only available studies that identify the objectives of fuel consumption minimisation and driving time as two conflicting objectives and study the problem as a bi-objective optimisation problem. Some recent research work has also acknowledged the problem of fuel-consumption minimising path identification in a time-dependent setting, and new emissions minimising vehicle routing models considering alternative road-paths between the consecutive truck stops have been published recently (Qian & Eglese, 2016; Ehmke et al., 2016; Huang et al., 2017; Androutsopoulos & Zografos, 2017).

This last category of the models constitutes the state-of-the-art in the EMVRPs as they offer a more realistic exposition of real life complications associated with UFD.

1.2.2 Green and Electric VRP

As a second major stream of research in greening UFD through VRPs, the use of vehicles that run on cleaner alternative fuels, such as electricity, hydrogen-gas and biofuel in the design of delivery routes has gained an increasing popularity in the last few years. Despite the significant role that these *Alternative Fuel Vehicles (AFVs)* can play in the development of environment-friendly freight distribution solutions, their application is still considerably hindered by their limited driving range, due to which visits to *Refuelling Stations (RSs)* during the course of the delivery might be required. This, however, brings in extra challenges owing to the fact that RSs are still very much scarce and unevenly scattered over the road network. In the case of electric vehicles this situation is still much more difficult as the time required to recharge their battery is significantly higher than conventional internal combustion engine vehicles.

The *Green vehicle routing problem (G-VRP)* which was first introduced by Erdoğan and Miller-Hooks (2012) concentrates on routing a fleet of AFVs considering the availability of RSs in the network. The main challenge in the G-VRP is hence to plan visits to the available RSs in the network as many times as it is required, as long as AFV routes are energy feasible and minimal in terms of the total distance travelled.

The *Electric Vehicle Routing Problem with Time-Windows (EVRPTW)* (Schneider et al., 2014), on the other hand, can be viewed as a special case of the G-VRP (Erdoğan & Miller-Hooks, 2012) where capacity constraints and time-windows are added to the problem, and significantly larger refuelling (recharging) time is assumed. In the variant considered by Schneider et al. (2014) a minimum number of ECVs must be assigned to energy-feasible delivery routes (potentially visiting one or several charging stations) that visit each customer exactly once during their pre-defined time-windows, such that the total capacity constraint of the ECV is not violated and the total distance travelled is minimised. Due to the limited driving range of ECVs, the core complication in the EVRPTW is related to the introduction of minimal detours in the vehicle routes to visit available *Charging Stations (CSs)* on the working graph to fully recharge their battery and carry on the delivery task.

To allow more flexibility in the design of the ECV delivery routes, Keskin and Çatay (2016) relax the full recharging restriction and allow partial recharging at a CS. Other variants of the EVRPTW considering different recharging strategies, recharging functions and fleet composition have been also explored in the literature. Felipe et al. (2014) solve a variant in which in addition to the decision on the charging level at a CS, the technology used for recharging e.g. regular or fast recharging is considered. Montoya et al. (2017) argue that the recharging level of the battery is a non-linear function of the recharging time and study the EVRP (without time windows) with a nonlinear recharging function. Hiermann et al. (2016) consider the fleet size and mix in the EVRPTWs where the available vehicle types in the fleet differ in terms of their capacity, battery size and acquisition cost. Goeke and Schneider (2015) study the EVRPTW with a mixed fleet of ECVs and conventional internal combustion commercial vehicles. A distinctive feature of their study is that instead of simply assuming energy consumption is a linear function of the distance travelled, they utilise an energy consumption model that takes speed, road slope and vehicle payload into account. In the same vein, Basso et al. (2019) incorporate into the routing decision an improved and more accurate energy consumption estimation model comprising detailed topography and speed profiles.

To compensate for the large recharging time of ECVs at CSs, research has also turned attention towards recent technological developments that allow swapping the ECV's depleted battery with a fully charged spare battery at a battery swapping station. In the *Battery Swap Station Location-Routing Problem (BSS-LRP)* studied by Yang and Sun (2015) and Hof et al. (2017) the locations of battery swap stations and the vehicle routes considering the limited range of the ECVs must be determined.

1.2.3 Time-dependent VRP

Traffic congestion could be broadly attributed to two different sources of congestion known as "recurrent" and "non-recurrent" congestion. Recurrent congestion refers to high volume of traffic seen during peak commuting hours, and non-recurrent congestion is due to incidents, such as accidents, vehicle breakdowns, bad weather, work zones, lane closures, special events, etc. (Güner et al., 2012). Time-dependent vehicle routing models are good tools for coping with the effects of recurrent congestion.

While conventional VRPs are often conducted under the assumption that all the information necessary to formulate the problems is time-invariant, in the *Time-Dependent VRP (TD-VRP)* arc traversal times are assumed varying exogenously due to traffic congestion (Gendreau & Tarantilis, 2010). Therefore, given the availability of historical data on traffic congestion, travel time of any given road link or arc in the network is assumed a function of the start time from the origin node of the arc.

Early TDVRP models (Hill & Benton, 1992; Malandraki & Daskin, 1992; Malandraki & Dial, 1996) contained the shortcoming of the undesired effect of passing, stemmed from the way travel times were modelled. The non-passing property, well-known as the FIFO property, is a logical requirement that ensures a later start time cannot lead to an earlier arrival time. To adhere to the FIFO principle, Ichoua et al. (2003) suggested adopting a step function for the speed, using which a piecewise linear function for travel time could be deduced that satisfies the FIFO property (Figure 1-1). They proposed an algorithmic procedure for the calculation of the time-dependent travel time, on which most of the existing time-dependent vehicle routing literature relies.

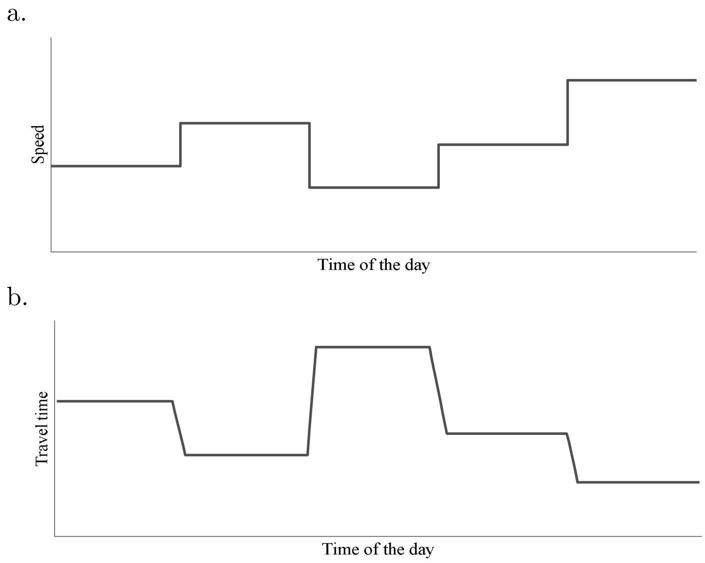


Figure 1-1 (a.) a step function for speed and (b.) the resulting piecewise linear function for travel time for a given arc using the model proposed by Ichoua et al. (2003)

To address a shortcoming of the model proposed by Ichoua et al. (2003), that corresponds to ignoring the time required for acceleration/deceleration from one speed level to the next, the more realistic FIFO-consistent model of Horn (2000) has also attracted some recent attention (Androutsopoulos & Zografos, 2012; Androutsopoulos & Zografos, 2017). This model uses directly in its input the speed data as they are archived by traffic agencies, i.e. time-series of speed observations. Connecting each speed observation at each given time instant from such databases results in a continuous piece-wise linear function of the time for speed (including acceleration/deceleration rates), using which a FIFO-consistent non-linear travel-time function could be deduced.

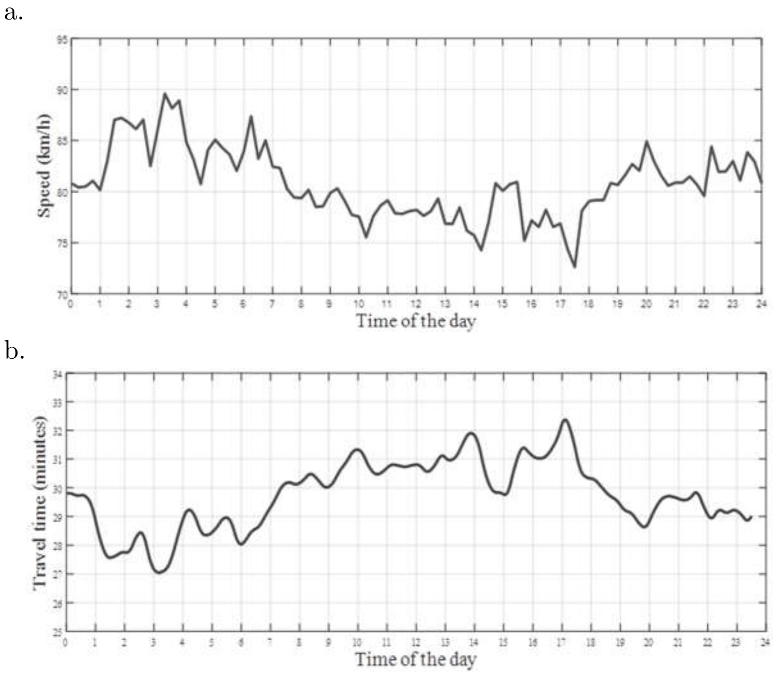


Figure 1-2 (a.) speed as a continuous piece-wise linear function of the time (b.) the resulting non-linear travel time function using the model proposed by Horn (2000)

In Figure 1-2.a fifteen-minute data series of average speed provided by data.gov.uk (Highways England, 2018) for March 2015 on a given 'A' road in the UK is illustrated. Figure 1-2.b. shows the corresponding non-linear travel time function estimated for 40 kilometres travelled under this speed profile using the model proposed by Horn (2000).

1.2.4 VRPs on road networks

It is a common practice in most of the VRP literature to transform the original road network into a complete graph of only the required nodes (i.e. the depot and customers), through the calculation of the shortest (or cheapest) path between every given pair of customers, and customer and depot, in a pre-processing stage (Figure 1-3). This way the routing decision on the transformed graph is only concerned with sequencing the customers' visits, ignoring the intermediate road-path finding problem between them. However, in many situations this representation results in missing important information contained in the original road network and hence can have a negative impact on the quality of the solutions obtained.

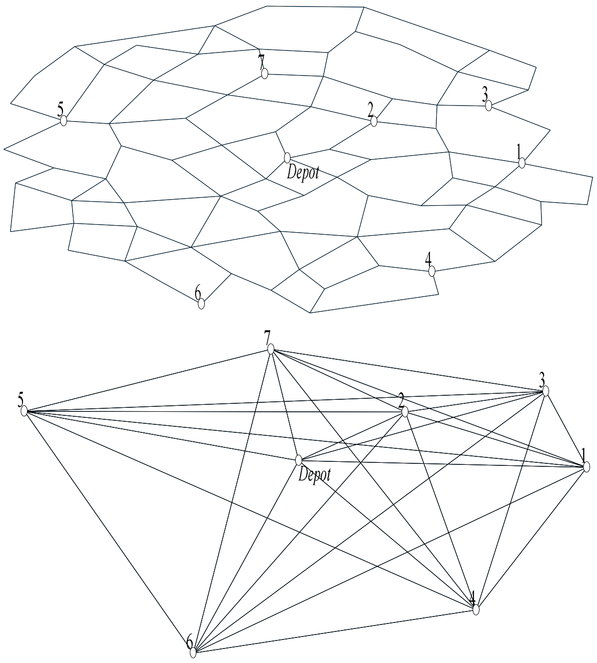


Figure 1-3 Transformation of the original road network into a complete graph of the required nodes

One example of such situation arises when routing on the congested urban road network for fuel consumption minimisation. Due to the time-varying congestion in urban areas, depending on the time of the day at which a customer is to be left towards a next customer on the vehicle's schedule, the optimal path can vary. Combine with this the lack of knowledge about the load on the vehicle on its departure which is an undecided factor until the entire routing plan is revealed.

This issue has been recently acknowledged in the literature and there are few papers that have tried to address it by studying the problem on the road network. Qian and Eglese (2016) and Huang et al. (2017) propose to use a multi-graph of the time-dependent shortest paths between the required nodes, where a set of such paths as candidate paths are precomputed between every pair of required nodes and kept. Ehmke et al. (2016) propose a new result that identifies a condition under which a time-dependent path between two customers is load invariant. This allows them to reduce the computational challenge of finding the time and load-dependent paths between some customers at some time instants by making it possible to precompute expected time-dependent fuel consumption minimising paths between them. Androutsopoulos and Zografos (2017) propose a network reduction approach that is based on the use of the k-shortest distance road paths. They show that when k is small (e.g. k=2) eligible paths might be excluded from the reduced network, and if a higher value for k is selected (e.g. k=5), while the number of excluded eligible paths is reduced, the computational time increases, accordingly.

Besides the cases when the nature of the objective function of concern necessitates the study of the problem on the original road network, when multiple attributes are associated with each arc in the network, the problem of finding the best path between two points of interest becomes multi-criteria and efficient paths with different compromises between the different attributes, are missed as a result of network abstraction to a complete graph. Moreover, it has also been argued that from a methodological point of view it is not always beneficial to work on a transformed graph instead of the original road network (Ben Ticha et al., 2018).

1.2.5 Other pertinent variants of the VRP

There are other variants of the VRP that are closely related to the problems studied in this thesis. Alongside with the well-known VRP with time windows, the fleet size and mix VRP, and the multi-objective VRP, a relevant variant of the VRP that has a role to play in greening UFD is the multi-trip VRP. Unlike conventional VRPs, in multi-trip VRP vehicles are allowed to make several journeys during a day. In VRPs with environmental considerations, this has important implications regarding multiple uses of energy-efficient resources multiple times during the planning horizon. Moreover, multi-trip VRPs are particularly useful for planning UFD, as in urban areas vehicle trips are rather short and it is possible to reload vehicles and dispatch them for an extra round of delivery.

Finally, VRPs with synchronisation requirements are also pertinent particularly to the problem studied in the fourth chapter of this thesis. Due to the existence of specific spatiotemporal synchronisation requirement in the proposed problem in chapter 4 of the thesis, it comprises some similarities with a class of VRPs known as the two-Echelon VRP with Satellite Synchronisation (2E-VRP-SS). In 2E-VRPSSs (Anderluh et al., 2017; Crainic et al., 2009; Grangier et al., 2016) there is a need to establish temporal synchronisation between the vehicles in the first echelon with the vehicles of the second echelon at an intermediate site, called satellite. The main complication that arises in establishing such synchronisation is due to the fact that unlike in the standard VRPs where vehicles are independent of one another, in VRPs with temporal synchronisation constraints, a change in one route may have effects on other routes, and in the worst case, a change in one route may render all other routes infeasible.

1.2.6 Exact and heuristic solution algorithms

VRP is an NP-hard problem and exact algorithms can address relatively small instances. Hence, in practice heuristics and metaheuristics that are capable of producing high-quality solutions in limited time are often used. Typical exact algorithms for the VRP include dynamic programming (Christofides et al., 1981b), branch-and-bound (Christofides et al., 1981a), branch-and-cut (Laporte et al., 1985), and branch-and-cut-and-price (Fukasawa et al., 2006), with the latter demonstrating noticeable success recently in solving richer variants of the VRP, too (e.g. Dabia et al., 2016 and Desaulniers et al., 2016).

On the other hand, a lot of heuristic and metaheuristics have been proposed for the VRP. Traditional heuristic approaches could be classified into: (i) route construction and (ii) route improvement (local search) methods. In route construction heuristics customers are selected sequentially based on some cost minimization criterion, subject to the problem constraints, until a feasible solution is created (Bräysy & Gendreau, 2005a). This can be done either sequentially (i.e. constructing one route at a time) or in parallel (i.e. building several routes simultaneously). The saving algorithm of Clark and Wright (1964), the sweep algorithm of Gillet and Miller (1974), and the parallel-route building algorithm of Potvin and Rousseau (1993) are the well-known classics of the route construction heuristics. Route improvement methods, on the other hand, are based on the concept of iteratively improving a solution by exploring neighbouring ones. The key concept here is to select the appropriate move-generation mechanism that creates the neighbouring solutions by changing one or several attributes (e.g. arcs connecting a pair of customers) of the current solution. A generated neighbouring solution is then compared against the current solution and if it is better, it replaces the current solution, and the search continues (Bräysy & Gendreau, 2005a). Most iterative improvement methods for the VRP are edge-exchange algorithms such as the well-known classical 2-opt and 3-opt edge exchange procedures of Lin (1965) and the Or-opt operator of Or (1977).

The local optimum produced by local-search algorithms can be very far from the optimal solution, as local-search methods perform a myopic search by only sequentially accepting solutions that provide improvements in the objective function value (Bräysy & Gendreau, 2005a). To avoid being trapped in a low quality local optimum, metaheuristics are devised to explore the wider solution space by accepting non-improving solutions also in a systematic way during the search. Successful examples of metaheuristics that have been applied on important variants of the VRP include Tabu search (Glover, 1989), genetic algorithm (Holland, 1992), simulated annealing (Kirkpatrick et al., 1983), ant colony systems

(Dorigo & Di Caro, 1999) and large neighbourhood search (Pisinger & Ropke, 2010).

In recent years, matheuristic solution algorithms, which are based on the hybridisations of heuristic and exact solution techniques, have also been gaining popularity, especially in solving richer variants of the VRP (Archetti & Speranza, 2014; Doerner & Schmid, 2010; Grangier et al., 2017; Kramer et al., 2015a; Villegas et al., 2013). While any solution methodology that uses mathematical modelling or exact solution algorithms like dynamic programming within the structure of a metaheuristic can be identified as a matheuristic, most of the existing matheuristics are based on decomposition approaches and column generation-based methods. In decomposition approaches, the problem is usually divided into smaller and simpler sub-problems and a specific exact solution method is applied to some or all of these sub-problems. Column generation-based approaches, on the other hand, preserve the structure of the branch-and-price methods but use either restricted master heuristics, heuristic branching approaches and/or relaxation-based approaches (Archetti & Speranza, 2014).

As regards algorithmic developments for realistic variants of the VRP with environmental considerations, contrary to the modelling developments in the field, the literature is significantly lagging behind. Incorporation of factors that have a major impact on fuel consumption, such as vehicle speed, acceleration/deceleration rates, load, and vehicle type, requires the unification of several hard variants of the VRP such as time-and-load dependent VRPs that are studied directly on the road network. Moreover, the need to consider environmental criteria alongside traditional business objectives leads to the necessity of multi-objective optimisation of these models that adds extra challenges. Algorithms that are able to address these rich variants of the VRP with environmental considerations are still very rare.

1.3 Contribution

Despite the rapid developments in the field of VRPs with environmental consideration, modelling and algorithmic developments relevant to realistic variants that can incorporate the real operating conditions of UFD are still very rare. Incorporation of technological developments pertinent to alternative fuel vehicles (electric commercial vehicles in particular) into routing problems dedicated for these vehicles has been also advancing rather slowly. This thesis is trying to close some of these gaps through the development of new mathematical models and solution algorithms for the VRP with environmental considerations.

The contribution of the thesis can be summarised as follows:

- The thesis introduces a new realistic variant of the pollution-routing problem, referred to as the *Steiner Pollution-Routing Problem (SPRP)* that is studied directly on the original urban roadway network. The proposed variant is a multi-objective, time and load dependent, fleet size and mix PRP with multiple trips, time windows, and flexible departure times on congested urban road networks. In particular, the added value of the proposed model is in integrating all previously studied attributes contributing to fuel consumption, and other new important decisions such as multiple trips, into a single modelling and solution scheme.
- To overcome difficulties in solving the proposed SPRP problem directly on the original roadway network, the thesis introduces new combinatorial results that are used in a new exact path elimination procedure to reduce the network size to a significant extent in a fast pre-processing stage by discarding all proven to be redundant paths from the network.
- To address a shortcoming of previous research in estimating fuel consumption accurately, the thesis proposes to use a microscopic fuel estimation formula that incorporates instantaneous truck kinematic

variables including the time-dependent second-by-second speed and *Acceleration/Deceleration (A/D)* rates in the proposed models; moreover, to supply the models with the possible lack of the instantaneous truck A/D data at the planning stage, a new mixed integer linear programming model for the construction of realistic spatiotemporal driving cycles from macroscopic traffic speed data is proposed.

- Three new multi-objective optimisation heuristics that are tailored to solve the SPRP instances with realistic sizes within a reasonable computational time are developed; the proposed heuristics output a pool of Pareto optimal solutions, representing the trade-off between business and environmental objectives, and their main added value is in that the considered problem unifies a number of existing and new complex EMVRP features and difficult variants of the VRP.
- The thesis introduces a new paradigm shift in electric vehicle routing models by exploiting new technological developments in the *Electric Vehicle Routing Problem with Synchronised Ambulant Battery Swapping (EVRP-SABS).*
- To solve the proposed EVRP-SABS, the thesis develops new combinatorial results that are used in an exact eligible path identification procedure for the identification of the set of the paths that must be retained between a pair of customers or a customer and the depot, a priori. These results are hence used in a strengthened formulation for the problem that can tackle some large instances, and in a proposed 2-stage matheuristic solution algorithm for the problem.

1.4 Structure of the thesis

The next three chapters of this thesis consist of three research articles that are either submitted for publication or are in preparation for submission. The following is a brief description of each chapter:

Chapter 2: The Multi-objective Steiner Pollution-Routing Problem on Congested Urban Road Networks. This chapter introduces the SPRP and proposes a new exact path elimination procedure for alleviating the difficulty of solving the problem directly on the roadway network. The chapter also proposes a new model for the construction of reliable synthetic spatiotemporal driving cycles from available macroscopic traffic speed data.

Chapter 3: Multi-objective Optimisation Heuristics for the Steiner Pollution-Routing problem. This chapter develops three new multi-objective optimisation heuristics to approximate the true Pareto optimal frontier of realistic instances of the SPRP.

Chapter 4: The Electric Vehicle Routing Problem with Synchronised Ambulant Battery Swapping/Recharging. This chapter introduces the EVRP-SABS as a class of EVRPs that exploits new technological developments pertaining to mobile battery swapping/recharging of electric vehicles. The chapter also proposes new results that are used in solving the newly proposed problem.

Finally, *chapter 5* is the discussion and concluding remarks chapter of the thesis.

2. THE MULTI-OBJECTIVE STEINER POLLUTION-ROUTING PROBLEM ON CONGESTED URBAN ROAD NETWORKS

--AN EXCERPT OF THIS CHAPTER IS PUBLISHED AT TRANSPORTATION RESEARCH PART B: METHODOLOGICAL--

2.1 Introduction

Urban Freight Distribution (UFD) is essential to the functioning of urban economies; however, it generates significant externalities such as traffic congestion and environmental pollution. European surveys indicate that the share of emissions of freight vehicles is between 20% and 30% of the total urban traffic emissions. For instance, in London in 2006 around 23% of *Carbon Dioxide (CO_2)* emitted by all forms of transport was due to freight vehicles (MDS Transmodal, 2012). The European Commission has therefore set a target for "essentially CO_2-free city logistics in urban centres by 2030" (European Commission, 2011).

A recent survey that is aimed at assessing the target set by the European Commission (Allen et al., 2017) reviews freight initiatives that are expected to reduce *Heavy Goods Vehicles (HGVs)* kilometres and CO_2 emissions in European urban areas, and ranks vehicle routing and scheduling tools among the top 10 impactful initiatives, which can help achieving around 23% reduction in HGV vehicle kilometres by 2030. Therefore, introducing pollution related objectives into traditional *Vehicle Routing Problems (VRPs)* can be viewed as a major approach to combat *Greenhouse Gas (GHG)* emissions, and can assist decision makers to strike a balance between business and environmental objectives. This need has led to the development of a significant body of the literature related to the *Emissions Minimising Vehicle Routing Problems (EMVRPs)*, comprising Green VRPs and the *Pollution-Routing Problem (PRP)* (Bektaş & Laporte, 2011). In the EMVRPs a fuel consumption estimation model, which is dependent on several vehicle and roadway network characteristics, is explicitly incorporated into the routing decision. Hence, unlike the traditional VRP that is predominantly concerned with the allocation of customers to feasible truck routes, realistic emissions minimising routing decisions on congested urban road networks must address a much more complicated decision, mainly due to the effect that the time-varying traffic conditions, the vehicle payload, and certain vehicle's physical and mechanical characteristics have on the fuel consumption level of a truck. A very first implication of this, which has not been sufficiently addressed by the previous related work in the field, is that the consideration of a priori determined single road-path for travelling between consecutive truck visits is not possible, as in practice several alternative paths can become optimal in terms of the fuel consumption between a given origin-destination pair on the underlying roadway graph depending on the departure time from the origin node, the load on the truck, and the type of the truck that is to be dispatched; none of which are known prior to realising the full route plan and schedule. Despite very recent efforts in addressing this situation(Androutsopoulos & Zografos, 2017; Ehmke et al., 2016; Y. Huang et al., 2017; Qian & Eglese, 2016), existing approaches can only identify a limited subset of the eligible paths for the time and load dependent emissions minimisation, and it is still an open research issue to identify optimally all road-paths that must be retained.

This paper aims to close this gap by studying a new variant of the PRP, called the *Steiner PRP (SPRP)*, directly on the original urban roadway network, and proposing new combinatorial results to develop an exact path elimination approach for the identification of the full set of the eligible road-paths (i.e. paths that might appear in a fuel consumption minimising route) in a fast pre-processing stage. It is worth mentioning that in calling the proposed variant the SPRP, we are following Cornuéjols, Fonlupt, and Naddef (1985) and Letchford, Nasiri, and Theis (2013) in calling a relevant variant of the *Travelling Salesman Problem (TSP)* on road networks as the Steiner TSP.

In addition to the aforementioned issue, the first step in constructing fuel-efficient truck routes involves an accurate estimation of the amount of fuel consumed at each route. Existing PRPs assign this task to a simplified average-speed version of the *Comprehensive Modal Emissions Model (CMEM)* formula (Barth et al., 2004), in which all model parameters are assumed to remain constant during a truck haul, except for load and speed which might vary from one road-link to another. However, as it has been recently argued by Turkensteen (2017), CMEM is a microscopic fuel consumption and emission model and relies on instantaneous vehicle kinematic variables, such as second-by-second speed and *Acceleration/Deceleration (A/D)* rates. Lack of this information at the planning stage, especially the A/D rates, which are assumed zero over a truck haul in average-speed CMEM, can lead to an inaccurate estimation of fuel consumption and hence might lead to unreliable and misleading routing decisions. While the proposed model in this paper is developed to work directly with the instantaneous CMEM formula, to address the lack of truck A/D data we propose a simple and reliable model for the construction of synthetic driving cycles from the available macroscopic traffic speed data.

The paper also acknowledges the fact that in urban areas travel times are rather small and it is often possible that after performing short routes trucks are reloaded and used again (Olivera & Viera, 2007). Therefore, the proposed model in the paper incorporates for the first time in the area of EMVRPs the decision regarding multiple uses of the cost and energy efficient resources during the planning horizon through the multi-trip decision-making in a multi-objective setting, where both business and environmental objectives are considered. In the SPRP three objective functions pertaining to (i) vehicle hiring cost, (ii) total amount of fuel consumed, and (iii) total makespan (duration) of the routes are considered, and to solve the problem to multi-objective optimality the paper develops a multi-phase solution framework, underpinned by the proposed exact network reduction technique, for the identification of the full set of the *Non-Dominated (ND)* points.

The contribution of this paper is multi-fold: (i) the SPRP is introduced as a multi-objective, time and load dependent, fleet size and mix PRP with multiple trips, time windows, and flexible departure times on congested urban road networks. In particular, the added value of the proposed model is in integrating all previously studied attributes contributing to fuel consumption, and other new important decisions such as multiple trips, into a single modelling and solution scheme, (ii) to overcome difficulties in solving the problem directly on the original roadway network, and eliminate the computational burden of the intermediate problem of finding the emissions minimising paths between consecutive visits on-the-fly, new combinatorial results are developed and used in proposing a new exact *Path Elimination Procedure (PEP)* that reduces the network size to a significant extent in a fast pre-processing stage by discarding all proven to be redundant paths from the network without eliminating ad-hoc ND points, (iii) to address a shortcoming of previous research in estimating fuel consumption accurately, the microscopic CMEM formula incorporating instantaneous truck kinematic variables including the time-dependent second-by-second speed and A/D rates is used in the models proposed in the paper, and (iv) a new *Mixed Integer Linear Programming (MILP)* model for the construction of realistic road-and-time-dependent driving cycles from macroscopic traffic speed data is proposed, to supply the model with the possible lack of the instantaneous truck A/D data at the planning stage.

The remainder of the chapter is structured as follows: section 2.2 discusses a background on the most relevant literature. Section 2.3 develops the SPRP model. Section 2.4 elaborates on the proposed path elimination approach and the model based on it. Section 2.5 discusses the methodology used for the identification of ND points to the SPRP. Section 2.6 discusses the proposed approach for generating driving cycles. Computational experiments are presented in Section 2.7; and finally, section 2.8 concludes the chapter.

2.2 Previous related work

The problem considered in this paper encompasses several attributes frequently encountered in real world urban freight distribution settings, including the time-varying road congestion, time and load dependent path selection, multiple use of the vehicles in the fleet, decisions on hiring a heterogeneous fleet of vehicles, and inclusion of both business and environmental objectives in decision making. There is research work focusing on each independent aspect of the proposed problem; however, the intention of this section is to discuss a selected review of the key studies in the general area of emissions minimising vehicle routing, and the more specific area of emissions minimisation on congested urban road networks. For an inclusive and up-to-date review on the state-of-the-art literature on the role of operational research in green freight transportation, the reader is referred to the recent study of Bektaş, Ehmke, Psaraftis, and Puchinger (2019b). We may also refer to Ben Ticha, Absi, Feillet, and Quilliot (2018) for a review on VRPs that are studied on road networks.

In the VRPs with explicit consideration of environmental performance of the planned routes, fuel consumption is usually used as a proxy for pollutants emissions, particularly CO_2 emissions which are proportional to the amount of fuel consumed. Fuel consumption is in turn dependent on many factors, and several of these factors such as vehicle type, speed, and load have already been considered in emissions minimising vehicle routing models. Some of these models only incorporate the effect of the load carried by the vehicle on the fuel consumption level of routes (Kara et al., 2007; Ubeda et al., 2011; Xiao et al., 2012). The main bearing on the emissions level in this category of models is due to the sequence that customers are visited which affects the payload between consecutive visits. More sophisticated models recognise the major role of the vehicle speed over each road-link in addition to the vehicle load. In the PRP (Bektaş & Laporte, 2011) and several related papers (Demir et al., 2012; Kramer et al., 2015) fuel consumption is assumed a nonlinear convex function of the vehicle speed, and hence speed optimisation in a time-independent setting, where non-congested traffic conditions are assumed throughout the day throughout the network, is attempted. However, the acknowledgement of the fact that travel speed could not be freely chosen in congested urban areas, as it fully depends on the expected time-varying traffic conditions, has led to the consideration of time-dependent variants of the VRP for a more accurate estimation model of fuel and other relevant decisions with temporal dependencies (Figliozzi, 2010; Figliozzi, 2011; Franceschetti, Honhon, Woensel, Bektaş, & Laporte, 2013; Ehmke et al., 2016; Androutsopoulos & Zografos, 2017; Çimen & Soysal, 2017; Ehmke et al., 2018). Some of these studies also consider the possible benefits of waiting at the depot and/or the customers (Xiao et al., 2012; Franceschetti et al., 2013; Androutsopoulos & Zografos, 2017).

Very recent research (Kancharla & Ramadurai, 2018; Turkensteen, 2017) has shed light on the inaccuracy of the fuel consumption estimation model used within EMVRPs due to ignoring truck A/D rates. Using numerical experiments from available chassis dynamometer driving schedules, Turkensteen (2017) shows that the magnitude of this error can be high. Kancharla and Ramadurai (2018), on the other hand, collect some on-road truck A/D data in a time-independent and static setting and randomly feed these data into their model for fuel consumption estimation. A major shortcoming of their proposed approach, however, lies in the fact that the spatial and temporal characteristics of the road-links in the graph are completely ignored. This issue is addressed in the current work by using the microscopic CMEM formula that incorporates instantaneous time-dependent second-by-second speed and A/D rates.

The effect of the type and the number of the trucks that are included in the fleet to execute the routes on fuel consumption was previously considered in the context of the EMVRPs by Koç et al. (2014) and Xiao and Konak (2016). The studies of Demir, Bektaş, and Laporte (2014) and Androutsopoulos and Zografos (2017) are the only available studies that identify the objectives of fuel consumption minimisation and driving time as two conflicting objectives and study the problem as a bi-objective optimisation problem. Some very recent research work has also acknowledged the problem of fuel-consumption minimising path identification in a time-dependent setting, and new emissions minimising vehicle routing models considering alternative road-paths between the consecutive truck stops have been published very recently (Qian & Eglese, 2016; Ehmke et al., 2016; Huang et al., 2017; Androutsopoulos & Zografos, 2017). This last category of research is the most pertinent to the current study and will be discussed further in the sequel.

In Table 2-1 different attributes and features that are considered by the key literature in the area and the current work are indicated using tick marks. This table can highlight two major gaps in the field that the proposed work is trying to address: (i) despite its important implications with regard to multiple use of energy-efficient resources multiple times during the planning horizon, multi-trips decision making has not yet been incorporated into emissions minimising vehicle routing models, and (ii) all factors and attributes identified and addressed have not yet been unified into a realistic integrated modelling and solution framework. It is also worth mentioning that the proposed work in this paper is the first study in the area to consider vehicle cost as a major business objective next to makespan and fuel consumption objectives in a tri-objective setting.

Table 2-1
Overview of attributes covered by the previous related works

	Attributes covered									
Study	Time-dependency	Load-dependency	Time-windows	Fleet mix	Fleet size	Alternative paths	Departure time	Multi-trip	Multi-objective	Fuel estimation accuracy
Kara et al. (2007)	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Ubeda et al. (2011)	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Bektaş and Laporte (2011)	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗
Figliozzi (2011)	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗
Franceschetti et al. (2013)	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
Demir et al. (2014)	✗	✓	✓	✗	✗	✗	✗	✗	✓	✗
Koç et al. (2014)	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗
Xiao and Konak (2016)	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗
Qian and Eglese (2016)	✓	✗	✓	✗	✗	✓	✗	✗	✗	✗
Ehmke et al. (2016)	✓	✓	✗	✗	✗	✓	✗	✗	✗	✗
Huang et al. (2017)	✓	✓	✓	✗	✗	✓	✓	✗	✗	✗
Androutsopoulos and Zografos (2017)	✓	✓	✓	✗	✗	✓	✓	✗	✓	✗
Kancharla and Ramadurai (2018)	✗	✓	✗	✓	✗	✗	✗	✗	✗	✓
Proposed work	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Except for the last category of the models discussed above (i.e. models that consider alternative road-paths), a major limitation of most of the existing research work in the area of EMVRPs lies in the fact that they consider an a priori determined single road-path for travelling between each pair of customers. There are at least two main reasons why this is not possible when routing on a congested urban road network for fuel consumption minimisation:

- Determining a minimum fuel consuming path between a given pair of origin/destination on urban road networks with time-varying traffic conditions requires a knowledge of the time the origin node is to be departed, the type of the truck to be dispatched to traverse the path (in case of a heterogeneous fleet), and the load to be carried by the dispatched truck over the path. All these variables are unknown until the routing plan and schedule is fully realised, and therefore identifying a path (or a set of paths) between every pair of the required nodes (customers and the depot) in order to transform the roadway network into a complete graph seems impossible.

- For a given sequence of visits starting and terminating at the depot (a vehicle route), and a given departure time from the depot, it is not guaranteed that merely fuel consumption minimising paths are taken by the truck between every pair of consecutive stops in order to minimise the total amount of fuel required by the vehicle route. In other words, inferior paths in terms of the fuel consumption might appear in the optimal fuel consuming vehicle route (see examples in Androutsopoulos and Zografos, 2017).

We are aware of 4 papers that acknowledge one or both of these issues and try to address them. Qian and Eglese (2016) and Huang et al. (2017) propose to use a multi-graph of the *Time-Dependent Shortest Paths (TDSPs)* between the required nodes, where a set of such paths as candidate paths are precomputed between every pair of required nodes and kept. However, this approach is not sufficient as it does not take load (and vehicle type) dependency into account. In the hope of partially tackling the effect of load-dependency, Huang et al. (2017) also include the distance-minimising path to the set of the time-minimising paths. While this might be partially helpful, there is no guarantee that all eligible road-paths are included. Ehmke et al. (2016) identify the first issue mentioned above and state that the identification of a set of all eligible paths a priori is not possible. Instead, to take both time and load-dependency into consideration, they propose a new result that identifies a condition under which a time-dependent path between two customers is load invariant (in case of a homogenous fleet). This allows them to reduce the computational challenge of finding the time and load-dependent paths between some customers at some time instants by making it possible to precompute expected time-dependent fuel consumption minimising paths between them. However, still for the rest of the customer pairs where the condition they check is not satisfied, they need to carry out shortest path computation on-the-fly in their Tabu Search algorithm, which is a costly requirement that prohibits solving problems with larger than 30 customers, even heuristically. Androutsopoulos and Zografos (2017) acknowledge both of the stated issues and propose a network reduction approach that is based on the use of the k-shortest distance road paths. This approach is, however, sensitive to the selection of the value of k. They try to show that when k is small (e.g. k=2) eligible paths might be excluded from the reduced network, and if a higher value for k is selected (e.g. k=5), while the number of excluded eligible paths is reduced, the computational time increases, accordingly.

Based on this review, the exact identification of the full set of the eligible emissions minimising road-paths between the required nodes on a time-dependent graph is still an open research issue. To tackle this, an efficient exact *Path Elimination Procedure (PEP)* is proposed by this paper that advances the result found by Ehmke et al. (2016) and can identify and discard all proven to be redundant paths between the required nodes in a pre-processing stage and eliminates the need for the shortest-path calculation on-the-fly. Our results are generalised for the case of a heterogeneous fleet, with multiple objective functions to be minimised by the planned routes.

2.3 Model development: notation and definitions

The SPRP is defined on a directed graph $G = (N, A)$, representing a real roadway network, where N is the set of network nodes and A is the set of directed road-links. The set $N = \{N_0 \cup N_1 \cup N_2\}$ is comprised of the depot $N_0 = \{0\}$, customer nodes $N_1 = \{1,2, \dots, n\}$, and network junctions $N_2 = \{n + 1, \dots, n + m\}$. There is a fleet of heterogeneous vehicles set K, with $|K| = \vartheta$, located in the central depot, which is assumed to be composed of $\Bbbk$ different types of trucks. To each truck $k \in$

K a curb weight μ_k *(kg)*, a maximum payload Q_k *(kg)*, and a daily hiring fixed cost c_k *(£)*, among other vehicle-specific factors such as engine friction factor, engine speed, engine displacement, coefficient of aerodynamic drag, and frontal surface area is attributed.

Each customer $i \in N_1$ is associated with a certain demand $q_i \leq \max_{k \in K} Q_k$ to be delivered within its pre-determined hard time window denoted by $w_i = [e_i, l_i]$, with service time s_i. The depot working hours which is also considered as the planning horizon is denoted by $T = w_0 = [e_0, l_0]$, and while it is assumed that trucks are initially loaded, reloading them for operating a new route takes s_0 time at the depot. To each road-link $(i, j) \in A$, a distance d_{ij}, and a time-dependent travel time t_{ij}^{τ}, depending on the departure time from the origin node i, i.e. $\tau \in [e_0, l_0]$ is attributed. In this study we assume that the time-dependent travel times (t_{ij}^{τ}) are integer.

The aim of the SPRP is to determine an optimal composition of vehicles in the fleet to operate routes that start and finish at the depot and serve every customer exactly once within their pre-defined time-windows, without violating vehicle capacities and working day limits, such that the following objectives are minimized: (i) vehicle hiring cost, (ii) total amount of fuel consumed, and (iii) total makespan (duration) of the routes.

The following terms are used throughout this paper:

- *Required nodes:* required nodes (N_R) are the nodes on the roadway network representing the location of the depot and the customers; i.e. $N_R = N_0 \cup N_1$.
- *Road-link:* a road-link is any kind of road in the hierarchy of roads such as a freeway, an arterial, a collector, or a local road that connects two nodes on the roadway network; i.e. $(i, j) \in A$.
- *Road-path:* a road-path p_{ij}, or simply a path, is a sequence of road-links which connects a pair of required nodes $i, j \in N_R$, $i \neq j$ on the roadway

network; i.e. $p_{ij} = [(i,1),(1,2),\dots,(\ell,j)], 1..\ell \in N \setminus i,j$. By convention, let us assume that $\mathscr{p}_{ij}$ is the set of all paths between a pair of required nodes, i.e. $\mathscr{p}_{ij} = \{p_{ij,1}, p_{ij,2}, \dots, p_{ij,\ell}\}$ (identification of this set can be intractable). Whenever it is needed, we denote the time-dependent travel time of a road-path p_{ij} by $\mathscr{t}^{\tau}(p_{ij})$ for departure time τ from node i; moreover, the fuel consumption of a truck $k \in K$, carrying a load $f \in [0, Q_k]$ over a given road-path p_{ij} is denoted by $\mathscr{f}_f^{k\tau}(p_{ij})$. Note that since no waiting is allowed at intermediate nodes between the origin and the destination of the path, knowing the departure time from the origin node is sufficient for estimating the time-dependent attributes of the path.

- *Route (trip):* A route or a vehicle trip (r) is a sequence of visits starting at the depot, passing through at least one customer and terminating at the depot, i.e. $r = \{0, i, \dots, 0\}, i \in N_1$.
- *Route-path:* A route-path (r_p) of route (r) is a route enhanced by the road-paths connecting every pair of consecutive required nodes on the route; i.e. $r_p = \{(0, i, p_{0i,1}), \dots, (j, 0, p_{j0,1})\}, i, j \in N_1, p_{0i,1} \in \mathscr{p}_{0i}, p_{0j,1} \in \mathscr{p}_{j0}$.
- *Route-trajectory:* A route-trajectory $(r_p^{\tau_0})$ is basically a scheduled route-path detailing the departure time from the depot and hence each required node on the route-path, i.e. $r_p^{\tau_0} = \{(0, i, p_{0i,1}, \tau_0), \dots, (j, 0, p_{j0,1}, \tau_\ell)\}, i, j \in N_1, p_{0i,1} \in \mathscr{p}_{0i}, p_{0j,1} \in \mathscr{p}_{j0}, \tau_0, \tau_\ell \in T$. The total makespan and fuel consumption of a truck $k \in K$ over a route-trajectory $r_p^{\tau_0}$ are denoted by $\mathscr{t}(r_p^{\tau_0}\ )$ and $\mathscr{f}^k(r_p^{\tau_0}\ )$, respectively. Note that, since $\mathscr{f}^k(r_p^{\tau_0}\ )$ is deduced from the aggregation of the fuel consumption over each of the constituting road-paths with varying payloads, it is not indexed by the truck load.
- *SPRP solution:* A feasible SPRP solution is a set of feasible route-trajectories enhanced by the type of the truck with enough capacity to carry out each one of them. The solution moreover specifies the amount of load

that is carried by each truck over each road-path, and any need for reloading any of the vehicles for extra rounds of trip.

The SPRP is categorised as a *Multi-Objective MILP (MOMILP)* problem with three conflicting objectives. Unlike the single objective optimisation, where a global optimal solution is reachable, solution to an MOMILP is the set of the ND points, or efficient solutions called the *Pareto Optimal Set (POS).* The reader is referred to Coello, Lamont, and Van Veldhuizen (2007) for definitions relevant to Pareto optimality, Pareto dominance, weak/strict Pareto optimality and the Pareto front.

A list of the notation used throughout the paper is given in Table 2-2. Note that Table 2-2 presents only notations that are used in more than one section, and additional notations that are used within specific sections are explained when used. It is also worth mentioning that while we have attempted to keep unique meaning for each notation to the greatest extent possible, in those cases where an item has additional uses, we have tried to make it clear from context.

Table 2-2
List of notation

Notation	Definition	Notation	Definition
$\boldsymbol{G}$	A directed graph representing a real roadway network	h_{ij}	Number of available speed observations during the planning horizon for road-link $(i,j) \in A$
$\boldsymbol{N}$	The set of network nodes in G	v_t	Observed speed at time instant m_t
$\boldsymbol{A}$	The set of directed road-links in G	a_t	Slope of the line segment connecting v_t to v_{t+1}
$\boldsymbol{N_0}$	The set comprising the depot only	$Z_{ij}^{k\tau}$	UTM attribute of a road-link (or a road-path if it is explicitly mentioned) for truck type $k \in K$ at departure time τ
$\boldsymbol{N_1}$	The set of customer nodes	$\mathit{\Gamma}_{ij}^{\tau}$	RTM attribute of a road-link (or a road-path if it is explicitly mentioned) at departure time τ
$\boldsymbol{N_2}$	The set of network junctions	$\mathcal{D}_{ij}$	The set of all minimum fuel consuming paths between required nodes i and j
$\boldsymbol{n}$	Number of customers	$\mathcal{E}_{ij}$	The set of all paths p_{ij} with non-dominated vectors $[\mathscr{t}^{\tau}(p_{ij}), \mathscr{f}_f^{k\tau}(p_{ij})]$
$\boldsymbol{\mathscr{m}}$	Number of network junctions	$ATT_{\mathfrak{p}}$	Tuple containing the attributes of path $\mathfrak{p}$
$\boldsymbol{K}$	The fleet of heterogeneous vehicles	$\mathcal{R}_{ij}^{k\tau}$	The set of retained eligible paths between nodes $i,j \in N_R$ for vehicle type $k \in K$ at departure time τ
$\boldsymbol{\vartheta}$	The total number of vehicles in K	$\mathcal{ATT}_{ij}^{k\tau}$	The set of tuples containing the attributes of all paths retained in $\mathcal{R}_{ij}^{k\tau}$
$\Bbbk$	The number of vehicle types	$\mathcal{M}_{ij}^{\tau}$	The ordered set of the k fastest paths at time instant τ
$\boldsymbol{\mu_k}$	Curb weight of vehicle $k \in K$	$\mathscr{r}$	Maximum number trips a truck is allowed to make during the planning horizon
$\boldsymbol{Q_k}$	Maximum payload of vehicle $k \in K$	$\mathscr{t}_{ijp}^{\mathbb{t}}$	Travel time of road-path $(i,j,p) \in \hat{A}$ during time period $\mathbb{t}$
$\boldsymbol{c_k}$	Daily hiring fixed cost of vehicle $k \in K$	$Z_{ijp}^{k\mathbb{t}}$	UTM attribute of road-path $(i,j,p) \in \hat{A}$ during time period $\mathbb{t}$
$\boldsymbol{q_i}$	Demand requested by customer $i \in N_1$	$\mathit{\Gamma}_{ijp}^{\mathbb{t}}$	RTM attribute of road-path $(i,j,p) \in \hat{A}$ during time period $\mathbb{t}$
$\boldsymbol{w_i}$	Hard time window of customer $i \in N_1$	$\mathcal{T}_{ijp}$	Time horizon dedicated to road-path $(i,j,p) \in \hat{A}$ for customised discretisation
$\boldsymbol{e_i}$	Lower boundary of w_i	$\mathscr{h}_{ijp}$	Number of time periods during $\mathcal{T}_{ijp}$
$\boldsymbol{l_i}$	Upper boundary of w_i	$\mathscr{a}_{ijp}^{\mathbb{t}}$	The lower boundary of time period $\mathbb{t} \in \mathcal{T}$
$\boldsymbol{s_i}$	Service time at customer $i \in N_1$	$\mathscr{b}_{ijp}^{\mathbb{t}}$	The upper boundary of time period $\mathbb{t} \in \mathcal{T}$
$\boldsymbol{T}$	The planning horizon	$x_{ijp}^{k\mathbb{t}}$	Binary variable equal to 1 iff vehicle $k \in K$ departs node $i \in \widehat{N}$ during time period $\mathbb{t} \in \mathcal{T}$ to go to node $j \in \widehat{N}$, through road-path $(i,j,p) \in \hat{A}$
$\boldsymbol{d_{ij}}$	Distance of road-link $(i,j) \in A$	$f_{ijp}^{k\mathbb{t}}$	Continuous variable to represents the size of load carried by vehicle $k \in K$ over the road-path $(i,j,p) \in \hat{A}$ during time period $\mathbb{t}$

$\boldsymbol{t}_{\boldsymbol{ij}}^{\boldsymbol{\tau}}$	The time-dependent travel time of road-link (or a road-path if it is explicitly mentioned) for departure time τ	$y_{ijp}^{k\mathbb{t}}$	Integer variable indicating the exact departure time from the origin of path $(i,j,p) \in \hat{A}$ given that it is departed by vehicle $k \in K$ during time period $\mathbb{t}$
$\boldsymbol{N_R}$	The set of required nodes	$\mathcal{Y}_{\mathbb{N}}$	The set of non-dominated points
$\boldsymbol{p_{ij}}$	A road-path that connects a pair of required nodes $i,j \in N_R$.	$\mathcal{A}_{max}$	The maximum possible acceleration rate for a truck
$\boldsymbol{\wp_{ij}}$	The set of all paths between a pair of required nodes $i,j \in N_R$	$\mathcal{D}_{max}$	The maximum possible deceleration rate for a truck
$\boldsymbol{t^{\tau}(p_{ij})}$	The time-dependent travel time of road-path connecting nodes i to j for departure time τ	v_U	The maximum possible speed in the network
$\boldsymbol{f_f^{k\tau}(p_{ij})}$	Fuel consumption of a truck $k \in K$, carrying a load $f \in [0, Q_k]$ at departure time τ	$\mathscr{v}_t$	Speed level at each second t of a driving cycle
$\boldsymbol{r}$	A route or a vehicle trip	ACC_t	The acceleration rate during second $t-1$ until t
$\boldsymbol{r_p}$	A route-path	DEC_t	The deceleration rate during second $t-1$ until t
$\boldsymbol{r_p^{\tau}}$	A route-trajectory	$\varkappa_t$	Binary decision variable equal to 1 iff vehicle accelerates during second $t-1$ until t
$\boldsymbol{t(r_p^{\tau})}$	Travel time of a route trajectory r_p^{τ}	ϵ_1	QSM parameter
$\boldsymbol{f^k(r_p^{\tau_0})}$	Fuel consumption of a truck $k \in K$ over a route-trajectory r_p^{τ}	ϵ_2	QSM parameter

2.3.1 The time-dependent travel time estimation model

Most of the existing research work in the area of the *Time-Dependent VRP (TDVRP)* relies on the model proposed by Ichoua, Gendreau, and Potvin (2003) for the calculation of the time-dependent travel time of a road-link. Their algorithm uses a step function for speed to deduce a piecewise linear function for travel time that satisfies the FIFO property. However, despite its ease of use, the application of the model proposed by Ichoua et al. (2003)on real life congestion situations is hindered by their extra simplification in viewing congestion speed as a step function which implies that changes in travel speed occur instantly (i.e. A/D rates equal to infinity) with unjustified leaps from one level to the next, ignoring the time required for A/D. In reality, a lot of such A/Ds occur during the actual arc traversal, especially in congested urban areas, and this leads to lack of accuracy in estimating the expected travel times.

This shortcoming can be overcome by using the travel time model in Horn (2000) which allows using directly the archived point-based historical speed data. Connecting each speed observation at each given time instant results in a continuous piecewise linear function of the time for speed (including A/Ds) (Figure 2-1), using which a FIFO-consistent non-linear travel-time function could be deduced. In Horn (2000) the computation of these travel times is performed by counting time from the departure time up to the point in time that a distance equal to the length of the arc is traversed, which can be computationally intensive. Androutsopoulos and Zografos (2012) enhance this model by presenting a closed form formula for calculating the travel time on any road-link given a departure time from the origin node, based on the computation of the arrival time at the destination node. Here, we propose an alternative closed form approach that compared with that of Androutsopoulos and Zografos (2012) is less complicated to implement and use, and unlike their formula does not require to observe if departure and arrival time occur within the same time interval or not. Another added advantage of the proposed formula is that it could be simply used in backward travel time calculation (i.e. finding the departure time for an intended arrival time).

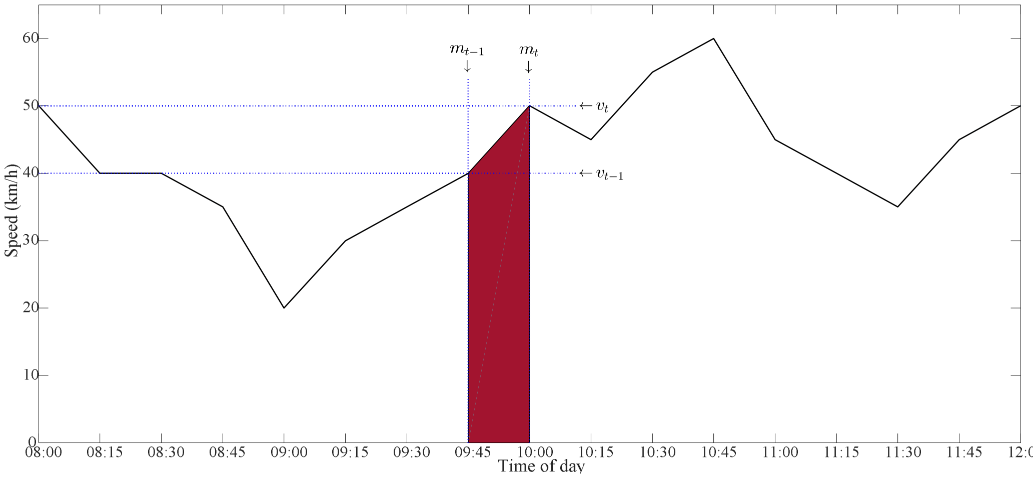


Figure 2-1 Speed as a piecewise linear function of time

To deduce the intended closed-form formula, suppose that there are h_{ij} speed observations during the planning horizon T for road-link $(i,j) \in A$. For notational simplicity, in the rest of this section we drop ij indices for all parameters. We denote by $v_t, t \in \{1, \dots, h\}$ the speed observation at time instant m_t. The line segment connecting v_t to v_{t+1} has a slope a_t (i.e. A/D rate) equal to $(v_{t+1} - v_t)/(m_{t+1} - m_t)$, and an intercept β_t equal to $(v_t - a_t m_t), \forall t \in \{1, \dots, h-1\}$ $(a_t = \beta_t = 0$ for $t = h)$. The distance that could be traversed from time instant $m_1 = e_0$ until time instant $m_t, t \in \{2, \dots, h\}$, is denoted by δ_t, and could be calculated as follows:

$$\delta_t = \delta_{t-1} + \left[\frac{1}{2}(m_t - m_{t-1})(v_t + v_{t-1})\right], \qquad \delta_1 = 0, t \in \{2, \dots, h\} \tag{2-1}$$

This equation is based on the premise that the area below the speed curve is equal to the physical distance that can be travelled. The equation hence calculates sequentially the area below the speed curve bounded by each pair of consecutive time instants (which is indeed a right trapezoid as highlighted in Figure 2-1) and adds this area to the entire area below the curve up until time instance m_{t-1}, i.e. δ_{t-1}. Based on the definition of δ_t, the time-definitive accumulated distance function ς, which is the building block of our closed form formula, is defined as follows:

Definition 1 *The time-definitive accumulated distance function $\varsigma(\tau)$ is defined as a function that calculates the distance possible to traverse from the beginning of the planning horizon until any given time instant $\tau \in T$:*

$$\varsigma(\tau) = \delta_{t+1} - \left[\frac{1}{2}(m_{t+1} - \tau)(v_\tau + v_{t+1})\right], \qquad \tau \epsilon [m_t, m_{t+1}), t \in \{1, \dots, h-1\} \tag{2-2}$$

where v_τ denotes the speed at time instant τ, which is equal to $v_t + [a_t(\tau - m_t)]$.

Then, for any given road-link in the network with distance d the following relationship always holds: $d = \varsigma(\tau_{\mathbb{a}}) - \varsigma(\tau_{\mathbb{d}})$, where $\tau_{\mathbb{d}}$ denotes the departure time from the origin of the given road-link, and $\tau_{\mathbb{a}}$ denotes the arrival time at the destination of the road-link. Hence, for any given departure time $\tau_{\mathbb{d}}$, the arrival time $\tau_{\mathbb{a}}$ could be found using the inverse of the ς function, and this implies the possibility of proposing a FIFO-consistent closed form formula for the time-dependent travel time calculation. To derive such formula, as $d = \varsigma(\tau_{\mathbb{a}}) - \varsigma(\tau_{\mathbb{d}})$, we have $\varsigma(\tau_{\mathbb{a}}) = d + \varsigma(\tau_{\mathbb{d}})$, which can be written as $\varsigma(\tau_{\mathbb{a}}) = E$, where $E = d + \varsigma(\tau_{\mathbb{d}})$, $E \in [\delta_t, \delta_{t+1}),\ t \in \{1, \dots, h-1\}$. Then, based on the definition of ς function we can write: $\delta_{t+1} - \left[\frac{1}{2}(m_{t+1} - \tau_{\mathbb{a}})(v_{\tau_{\mathbb{a}}} + v_{t+1})\right] = E$, and further based on the definitions of $v_{\tau_{\mathbb{a}}}$ and δ_{t+1}, we can write: $\delta_t + [\frac{1}{2}(m_{t+1} - m_t)(v_{t+1} + v_t)] - [\frac{1}{2}(m_{t+1} - \tau_{\mathbb{a}})(a_t\tau_{\mathbb{a}} + \beta_t + v_{t+1})] = E$, and writing this for $\tau_{\mathbb{a}}$ we will have the following expression:

$$\tau_{\mathbb{a}} = \begin{cases} \dfrac{[\beta_t^2 - 2a_t\delta_t + 2a_tE + a_t^2m_t^2 + 2a_tm_t\beta_t]^{1/2} - \beta_t}{a_t}, & a_t \neq 0 \\ m_t + \dfrac{E\ -\ \delta_t}{v_t}, & a_t = 0 \end{cases} \tag{2-3}$$

Note that the model by Ichoua et al. (2003)is a special case of expression (2-3), where $a_t = 0, \forall t \in T$. Also note that this formula can use microscopic traffic speed data (i.e. second-by-second speed variations) as well as macroscopic data (e.g. every 5, 10, or 15 minutes) as input. As will be discussed later in section 6 of the paper, when microscopic data are not available, the travel time estimated from macroscopic data using this formula provides a basis for the generation of synthetic driving cycles with instantaneous speed variations.

For the most efficient implementation of (2-3), all model parameters including a_t, β_t, and δ_t, and also $\varsigma(\tau)$ and v_τ for all possible departure times, could be pre-computed, which then make the application of expression (2-3) pretty simple and straightforward. A useful feature of (2-3) is that it is also possible to find the departure time τ_d for any given arrival time τ_a using the same formula with the only modification that $E = \varsigma(\tau_a) - d$.

It is worth mentioning that the time-dependent travel time of a given scheduled road-path could be simply estimated from its constituent road-links, and thus as the time-dependent travel times of the road-links preserve the FIFO property, any simple paths considered on the graph would be also FIFO-consistent (given that waiting is not allowed on intermediate nodes).

2.3.2 The instantaneous fuel consumption estimation model

Assume that the spatiotemporal instantaneous driving cycles DC_{ij}^{τ}, denoting the expected second-by-second speed variations, are available for each road-link $(i, j) \in A$ of the network, for all time instants $\tau \in T$. It is worth mentioning that with the current advancements in the *Global Positioning System (GPS)* devices, it is possible to create a historical archive of such data for the required road-links at different times of a day (Belliss, 2004; Byon et al., 2006; Lee et al., 2016); however, in the event that they are unavailable at the planning stage, they could be instead generated synthetically using the approach proposed later in the paper.

Given such cycles, the instantaneous time, load and truck-type dependent fuel consumption (in litres) over the given road link $(i,j) \in A$ for vehicle $k \in K$, i.e. $\mathcal{F}_{ij}^{k\tau}$, could be computed using the CMEM formula of Barth et al. (2004)as follows:

$$\begin{aligned} \mathcal{F}_{ij}^{k\tau} = \sum_{t=\tau}^{\tau+t_{ij}^{\tau}} \frac{\zeta}{\kappa\psi}\left(H_k \mathcal{N}_k V_k + \frac{0.5 C_k \rho A_k v_{ijt}^3}{1000\varepsilon\varpi}\right) + (\mu_k \\ + f_{ij}) \sum_{t=\tau}^{\tau+t_{ij}^{\tau}} \frac{\zeta}{\kappa\psi}\left(\frac{v_{ijt}(g\,sin\,\theta + gC_r\,cos\,\theta + a_t)}{1000\varepsilon\varpi}\right), \\ \forall (i,j) \in A, k \in K, \tau \in T, \end{aligned} \tag{2-4}$$

where ζ is fuel-to-air mass ratio, κ is the heating value of a typical diesel fuel (*kJ/g*), ψ is a conversion factor from grams to litres (from (*g/s*) to (*l/s*)), H_k is the engine friction factor (*kJ/rev/l*) for vehicle k, $\mathcal{N}_k$ is the engine speed (*rev/s*) for vehicle k, V_k is the engine displacement (*l*) for vehicle k, ρ is the air density (*kg/m³*), A_k is the frontal surface area (*m²*) for vehicle k, v_{ijt} is the vehicle speed (m/s) at the tth second of the cycle, μ_k is the vehicle curb weight (*kg*) for vehicle k, f_{ij} is the load (*kg*) carried over the given road link by the truck, g is the gravitational constant (equal to 9.81 *m/s²*), θ is the road angle, C_k and C_r are the coefficient of aerodynamic drag and rolling resistance, ε is vehicle drive train efficiency and ϖ is an efficiency parameter for diesel engines.

Expression (2-4) divides CMEM into a time-dependent term *Unrelated to Truck Mass* (called the *UTM* attribute and indicated by $Z_{ij}^{k\tau}$ hereafter), and a time-dependent term linearly *Related to the Truck Mass* (called the *RTM* attribute and indicated by Γ_{ij}^{τ} hereafter), and both of these could be precomputed and stored for all road-links (or road-paths) at all possible departure times based on the available DC_{ij}^{τ}s. Hence, this expression could be simply re-written as $\mathcal{F}_{ij}^{k\tau} = Z_{ij}^{k\tau} + \Gamma_{ij}^{\tau}(\mu_k + f_{ij}), \forall (i,j) \in A, \tau \in T$ (to see more detail on the derivation of this formula in a homogenous fleet case, the reader can refer to appendix A in Androutsopoulos and Zografos, 2017). As a note on the storage space requirement for storing all UTM and RTM attributes along with time-dependent travel times for all road-links at all possible departure times, it should be mentioned that the space complexity is $O((\Bbbk + 2)|A||T|)$, where $|A|$ is the number of network road-links, and $|T|$ is the number of all possible departure times. However, as will be explained later in section 2.4.1, this required storage space could be critically reduced by using 'time periods' instead of 'time instants'.

In this study, for experimental purposes, similar to the work of Koç et al. (2014) on the fleet size and mix PRP, we consider the fleet to be composed of light, medium and heavy duty trucks and use the same values they use for the common and vehicular specific parameters, which they obtain for the three main vehicle types of MAN Trucks (see Tables 1 and 2 in Koç et al. - 2014).

2.4 The Path Elimination Procedure (PEP)

The intention of this section is to deal with an important prerequisite to any subsequent exact/heuristic solution algorithm for the SPRP, which is to alleviate the difficulty of solving the problem directly on the real roadway network, without losing the essential information contained in the original graph. As discussed in section 2 of the paper, existing approaches in the literature can only identify a limited subset of eligible road-paths that must be preserved between the required nodes and cannot guarantee that all paths that might be used in the design of an optimal vehicle route are identified and preserved. In the case of the SPRP, a much more complicated situation must be coped with, since not only a time, load and truck type dependent fuel consumption objective is to be minimised, but also this objective is considered alongside two other conflicting objectives, and any set of the paths that are returned by any pre-processing algorithm should ensure that ad-hoc ND points will not be eliminated.

Since the main problematic objective function that causes complications is the time, load and truck type dependent fuel minimisation objective, we begin by focusing on this objective only, and then generalise all our results for the multi-objective case of the SPRP. The underlying idea of the proposed PEP in this section is hence to identify and retain all road-paths that might be used by at least one of the truck types in the fleet, for at least one time instant during the planning horizon, to carry some load levels in the range of the truck capacity, and then eliminate all other paths as redundant paths from the network. An "eligible" path can be therefore defined formally as follows:

Definition 2 *A road-path $p_{ij,a}$ between a pair of required nodes $i, j \in N_R$: $i \neq j$ is called an 'eligible' path, iff $\exists\, k \in K, f \in [0, Q_k], \tau \in T$: $\mathscr{f}_f^{k\tau}(p_{ij,a}) \leq \mathscr{f}_f^{k\tau}(p_{ij,\ell}), \forall p_{ij,\ell} \in \mathscr{P}_{ij}$.*

The elimination of an eligible road-path from the underlying roadway network can hence lead to a suboptimal vehicle route in terms of fuel consumption, and all such paths must be identified and retained.

In order to set the scene, the motivation of the PEP is reiterated through the following remarks:

Remark 1 *Determining a priori a (set of) minimum fuel consuming road-path(s) between a given pair of origin/destination on urban road networks with time-varying congestion seems impossible.*

Remark 2 *A minimum fuel consuming route-trajectory is not necessarily concatenated; i.e. its constituent scheduled road-paths are not necessarily optimal, and they can be road-paths which are inferior in terms of the fuel consumption.*

It is easy to acknowledge the first remark, which stems from our lack of knowledge about the departure time from the origin node, the type of the truck that is going to traverse the path and the amount of load that the truck is going to carry over the road-path, prior to realising the full truck route and schedule. However, the second remark is not as intuitive, because it might seem that once fuel consumption minimising road-paths between every consecutive visit, for every possible departure time, and any load on the trucks are known, these paths can be retained to minimise the overall amount of fuel required by the route, and the alternative inferior road-paths could be simply ignored. However, it is not difficult to show that it might be beneficial to take road-paths that are not optimal in terms of fuel consumption to gain improvements in the overall fuel consumption of the route (see example 1 in Androutsopoulos and Zografos, 2017). Note that despite Androutsopoulos and Zografos (2017) identify this as an inherent issue for the bi-objective time-dependent VRPs, it is even an issue in a single objective case. In fact, this is a largely ignored situation in any general routing for some time-dependent cost minimisation in a time-dependent network, and an important generalisation of Remark 2 is that the cheapest path in a time-dependent setting is not necessarily concatenated.

Let $\mathcal{D}_{ij}$ be the set of all minimum fuel consuming paths between required nodes i and j for all possible departure times $\tau \in T$ from node i. Then, building on some previous results for the time-dependent shortest path problems (Hamacher et al., 2006; Orda & Rom, 1990) the following theorem is proposed:

Theorem 1 *Suppose the set $\mathcal{E}_{ij}$ is the set of all paths p_{ij} with non-dominated vectors $[\cancel{t}^{\tau}(p_{ij}), \cancel{f}_{f}^{k\tau}(p_{ij})]$ for at least one $\tau \in T, k \in K,$ and $f \in [0, Q_k]$ (note that $\mathcal{D}_{ij} \subseteq \mathcal{E}_{ij}$), and let $\mathcal{E} = \{\mathcal{E}_{ij} | i, j \in N_R, i \neq j\}$; if departure time from the depot is unrestricted, then any optimal route-trajectory in terms of fuel consumption has its road-paths in $\mathcal{E}$.*

Proof. Without loss of generality, assume that $s_i = 0$ and $w_i = w_0$ for all $i \in N_1$. With this assumption, the departure time from a required node is upon the arrival time at the node from an upstream required node. Suppose that $r_1^{\tau_0} = \{(0, i, p_{0i,1}, \tau_0), (i, 0, p_{i0,1}, \tau_1)\}$ is an optimal route-trajectory in terms of fuel consumption. Since there is only one customer $i \in N_1$that is served over this route-trajectory, a truck $k \in K$, that is sufficiently large to carry q_i is used. According to the backward principle of optimality (see Definition 3.2 and Theorem 3.2 in Hamacher et al. - 2006) road-path $p_{i0,1}$ is optimal in terms of fuel consumption for departure time τ_1 from node i, and thus $p_{i0,1} \in \mathcal{E}$. Therefore, we must only prove that $p_{0i,1} \in \mathcal{E}$. To use a proof by contradiction, initially suppose that $p_{0i,1}$ is not in $\mathcal{D}_{0i}$. The assumption that $p_{0i,1}$ is not in $\mathcal{D}_{0i}$ implies that there is a fuel consumption minimising path $p_{0i,2} \in \mathcal{D}_{0i}$ that arrives at node i at time τ_1. Assume that to arrive at customer i at time τ_1, the truck must depart the origin of path $p_{0i,2}$ (i.e. the depot) at time τ_2 (remember that departure time from the depot is not restricted). This means $\mathscr{f}_{q_i}^{k\tau_2}(p_{0i,2}) < \mathscr{f}_{q_i}^{k\tau_0}(p_{0i,1})$, and there is a route-trajectory $r_2^{\tau_2} = \{(0, i, p_{0i,2}, \tau_2), (i, 0, p_{i0,1}, \tau_1)\}$, such that $\mathscr{f}^k(r_2^{\tau_2}) < \mathscr{f}^k(r_1^{\tau_0})$; contradicting the fact that $r_1^{\tau_0}$ is an optimal route trajectory. This proof is, however, incomplete under a certain condition; the departure time from the depot for path $p_{0i,2}$, i.e. τ_2 can be smaller or larger than τ_0, meaning that $\mathscr{t}^{\tau_0}(p_{0i,1})$ can be smaller or larger than $\mathscr{t}^{\tau_2}(p_{0i,2})$. Under the condition that $\tau_2 < \tau_0$ and $\tau_2 < e_0$, path $p_{0i,2}$ is infeasible; however, as in that case $p_{0i,1}$ has a non-dominated vector $[\mathscr{t}^{\tau_0}(p_{0i,1}), \mathscr{f}_{q_i}^{k\tau_0}(p_{0i,1})]$, it already exists in $\mathcal{E}$. □

Based on this theorem, the key to address the situation in Remark 2 is indeed departure time optimisation, and except for a special case, working on a graph based on $\mathcal{D}$ is usually sufficient for the minimisation of the fuel consumption by the routes. However, for completeness this theorem proposes to work on $\mathcal{E}$, since if the set $\mathcal{E}$ could be somehow constructed, the same minimum fuel consuming route-trajectories that can be found directly on G could be found on $\mathcal{E}$. In the sequel, we propose new results to construct this set.

In the rest of this section, whenever we refer to a road-path, it is meant to be a road-path between a given pair of required nodes $i, j \in N_R$, but for notational simplicity we drop origin/destination indices of paths and their attributes.

Moreover, instead of writing $\boldsymbol{t}^{\tau}(p_{\mathfrak{p}}), Z^{k\tau}(p_{\mathfrak{p}})$ and $\varGamma^{\tau}(p_{\mathfrak{p}})$ for the attributes of a path $p_{ij,\mathfrak{p}} \in \boldsymbol{p}_{ij}$, we simply write $\boldsymbol{t}_{\mathfrak{p}}^{\tau}, Z_{\mathfrak{p}}^{k\tau}$ and $\varGamma_{\mathfrak{p}}^{\tau}$, respectively.

Proposition 1 *(Ehmke et al., 2016) If path $\mathfrak{p}_1$ is a fuel consumption minimising path for both a fully loaded truck of type k, and an empty truck of the same type k for departure time instant τ, then this path is optimal in terms of fuel consumption for any other size of load on the truck of type k.*

This proposition modifies slightly the proposition stated in Ehmke et al. (2016), in the sense that they do not explicitly state that this is a condition that must be checked for all possible departure times. Moreover, to generalise it for a heterogeneous fleet, the type of the truck matters and is mentioned here. It is also worth noting that while they have proposed this proposition in the context of the average-speed CMEM, their proof is applicable for the case of the instantaneous CMEM, as well.

While Proposition 1 establishes an interesting result, which can be used to precompute expected time-dependent fuel consumption minimising paths between some customers at some time instants, our computational experiments on a real world urban road network demonstrate that there are cases when this condition is not satisfied for up to around 60% of the times (see section 2.7.1); nevertheless, Proposition 1 serves as a building block to a more important theorem that underpins the proposed PEP:

Theorem 2 *If for a given departure time τ, path $\mathfrak{p}_1$ is a fuel consumption minimising path for a fully loaded truck of type k, and path $\mathfrak{p}_2$ is a fuel consumption minimising path for an empty truck of the same type k such that $\mathfrak{p}_1 \not\equiv \mathfrak{p}_2$, then any other path $\mathfrak{p}_a$ is an eligible path iff it is a fuel consumption minimising path for truck type k carrying some load $f \in [\frac{Z_a^{k\tau}-Z_2^{k\tau}}{\varGamma_2^{\tau}-\varGamma_a^{\tau}} - \mu_k, \frac{Z_1^{k\tau}-Z_a^{k\tau}}{\varGamma_a^{\tau}-\varGamma_1^{\tau}} - \mu_k]$.*

Proof. We first lay out some useful valid inequalities derived from the assumptions: The optimality of p_1 for the fully loaded truck yields the following inequalities:

$$Z_1^{k\tau} + \Gamma_1^\tau(\mu_k + Q_k) \leq Z_2^{k\tau} + \Gamma_2^\tau(\mu_k + Q_k) \tag{2-5}$$

$$Z_1^{k\tau} + \Gamma_1^\tau(\mu_k + Q_k) \leq Z_a^{k\tau} + \Gamma_a^\tau(\mu_k + Q_k) \tag{2-6}$$

And the optimality of p_2 for the empty truck suggests the following:

$$Z_2^{k\tau} + \Gamma_2^\tau(\mu_k + 0) \leq Z_1^{k\tau} + \Gamma_1^\tau(\mu_k + 0) \tag{2-7}$$

$$Z_2^{k\tau} + \Gamma_2^\tau(\mu_k + 0) \leq Z_a^{k\tau} + \Gamma_a^\tau(\mu_k + 0) \tag{2-8}$$

Now, to prove the proposed "if and only if' statement a two-way proof must be given:

Part 1 (forward proof): Path p_a is a fuel consumption minimising path for truck type k carrying some load $f \in [\frac{Z_a^{k\tau} - Z_2^{k\tau}}{\Gamma_2^\tau - \Gamma_a^\tau} - \mu_k, \frac{Z_1^{k\tau} - Z_a^{k\tau}}{\Gamma_a^\tau - \Gamma_1^\tau} - \mu_k] \Rightarrow$ path p_a is an eligible path: if we only prove that $\frac{Z_a^{k\tau} - Z_2^{k\tau}}{\Gamma_2^\tau - \Gamma_a^\tau} - \mu_k \geq 0$, and $\frac{Z_1^{k\tau} - Z_a^{k\tau}}{\Gamma_a^\tau - \Gamma_1^\tau} - \mu_k \leq Q_k$, then we have proved $f \in [0, Q_k]$, which then makes the stated assumption per se sufficient for the eligibility of p_a (note that $\frac{Z_a^{k\tau} - Z_2^{k\tau}}{\Gamma_2^\tau - \Gamma_a^\tau} - \mu_k \leq \frac{Z_1^{k\tau} - Z_a^{k\tau}}{\Gamma_a^\tau - \Gamma_1^\tau} - \mu_k$ is already assumed). A proof by contradiction can be used where we assume either $\frac{Z_a^{k\tau} - Z_2^{k\tau}}{\Gamma_2^\tau - \Gamma_a^\tau} - \mu_k < 0$, or $\frac{Z_1^{k\tau} - Z_a^{k\tau}}{\Gamma_a^\tau - \Gamma_1^\tau} - \mu_k > Q_k$. If $\frac{Z_a^{k\tau} - Z_2^{k\tau}}{\Gamma_2^\tau - \Gamma_a^\tau} - \mu_k < 0$, since it is equivalent to $Z_a^{k\tau} + \Gamma_a^\tau \mu_k < Z_2^{k\tau} + \Gamma_2^\tau \mu_k$, we will have a contradiction with (2-8). At the same time, if $\frac{Z_1^{k\tau} - Z_a^{k\tau}}{\Gamma_a^\tau - \Gamma_1^\tau} - \mu_k > Q_k$, since it is equivalent to $Z_1^{k\tau} + \Gamma_1^\tau(\mu_k + Q_k) > Z_a^{k\tau} + \Gamma_a^\tau(\mu_k + Q_k)$, we will have a contradiction with (2-6). Therefore, neither $\frac{Z_a^{k\tau} - Z_2^{k\tau}}{\Gamma_2^\tau - \Gamma_a^\tau} - \mu_k < 0$, nor $\frac{Z_1^{k\tau} - Z_a^{k\tau}}{\Gamma_a^\tau - \Gamma_1^\tau} - \mu_k > Q_k$, and as $f \in [0, Q_k]$, path p_a is a fuel consumption minimising path for truck type k carrying some load $f \in [0, Q_k]$ and is hence an eligible path based on definition.

Part 2 (backward proof): path p_a is an eligible path $\Rightarrow$ Path p_a is a fuel consumption minimising path for truck type k carrying some load $f \in [\frac{Z_a^{k\tau} - Z_2^{k\tau}}{\Gamma_2^\tau - \Gamma_a^\tau} - \mu_k, \frac{Z_1^{k\tau} - Z_a^{k\tau}}{\Gamma_a^\tau - \Gamma_1^\tau} - \mu_k]$: The eligibility of p_a necessitates that both of the following inequalities hold for some $f \in [0, Q_k]$:

$$Z_a^{k\tau} + \Gamma_a^{\tau}(\mu_k + f) \leq Z_1^{k\tau} + \Gamma_1^{\tau}(\mu_k + f) \tag{2-9}$$

$$Z_a^{k\tau} + \Gamma_a^{\tau}(\mu_k + f) \leq Z_2^{k\tau} + \Gamma_2^{\tau}(\mu_k + f) \tag{2-10}$$

Since (2-9) is equivalent to $\frac{Z_1^{k\tau} - Z_a^{k\tau}}{\Gamma_a^{\tau} - \Gamma_1^{\tau}} - \mu_k \geq f$, and (2-10) is equivalent to $\frac{Z_a^{k\tau} - Z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_a^{\tau}} - \mu_k \leq f$, we have $\frac{Z_a^{k\tau} - Z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_a^{\tau}} - \mu_k \leq f \leq \frac{Z_1^{k\tau} - Z_a^{k\tau}}{\Gamma_a^{\tau} - \Gamma_1^{\tau}} - \mu_k$. □

Corollary 1 *If for a given departure time τ, path p_a is a fuel consumption minimising path for a truck of type k carrying load $f_a = \frac{Z_1^{k\tau} - Z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_1^{\tau}} - \mu_k$, then p_a is an eligible path.*

Proof. Based on Theorem 2 we must prove that $f_a \in [\frac{Z_a^{k\tau} - Z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_a^{\tau}} - \mu_k, \frac{Z_1^{k\tau} - Z_a^{k\tau}}{\Gamma_a^{\tau} - \Gamma_1^{\tau}} - \mu_k]$; that is, we must prove $\frac{Z_a^{k\tau} - Z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_a^{\tau}} \leq \frac{Z_1^{k\tau} - Z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_1^{\tau}} \leq \frac{Z_1^{k\tau} - Z_a^{k\tau}}{\Gamma_a^{\tau} - \Gamma_1^{\tau}}$. The optimality of p_a for the truck carrying load $f_a = \frac{Z_1^{k\tau} - Z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_1^{\tau}} - \mu_k$ yields:

$$Z_a^{k\tau} + \Gamma_a^{\tau}(\frac{Z_1^{k\tau} - Z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_1^{\tau}}) \leq Z_1^{k\tau} + \Gamma_1^{\tau}(\frac{Z_1^{k\tau} - Z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_1^{\tau}}) \tag{2-11}$$

$$Z_a^{k\tau} + \Gamma_a^{\tau}(\frac{Z_1^{k\tau} - Z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_1^{\tau}}) \leq Z_2^{k\tau} + \Gamma_2^{\tau}(\frac{Z_1^{k\tau} - Z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_1^{\tau}}) \tag{2-12}$$

where (2-11) is equivalent to $\frac{Z_1^{k\tau} - Z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_1^{\tau}} \leq \frac{Z_1^{k\tau} - Z_a^{k\tau}}{\Gamma_a^{\tau} - \Gamma_1^{\tau}}$, and (2-12) is equivalent to $\frac{Z_a^{k\tau} - Z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_a^{\tau}} \leq \frac{Z_1^{k\tau} - Z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_1^{\tau}}$. □

To understand better the proposed results and the proofs, we further provide some visual presentations in Figure 2-2. In Figure 2-2.a, the lines that correspond to the equations of the fuel consumption minimising paths for a full and an empty truck k at time instant τ, are given as p_1 and p_2, respectively. Figure 2-2.b illustrates an ineligible path that can never minimise the truck fuel consumption at any load level within the truck capacity. Figure 2-2.c, illustrates an eligible path satisfying all conditions set in the Theorem. This figure shows further the condition set in Corollary 1. Finally, Figure 2-2.d shows two different eligible paths satisfying all conditions. Obviously, no line of an eligible path could be sketched without its eligibility range being within the one defined in Theorem 2.

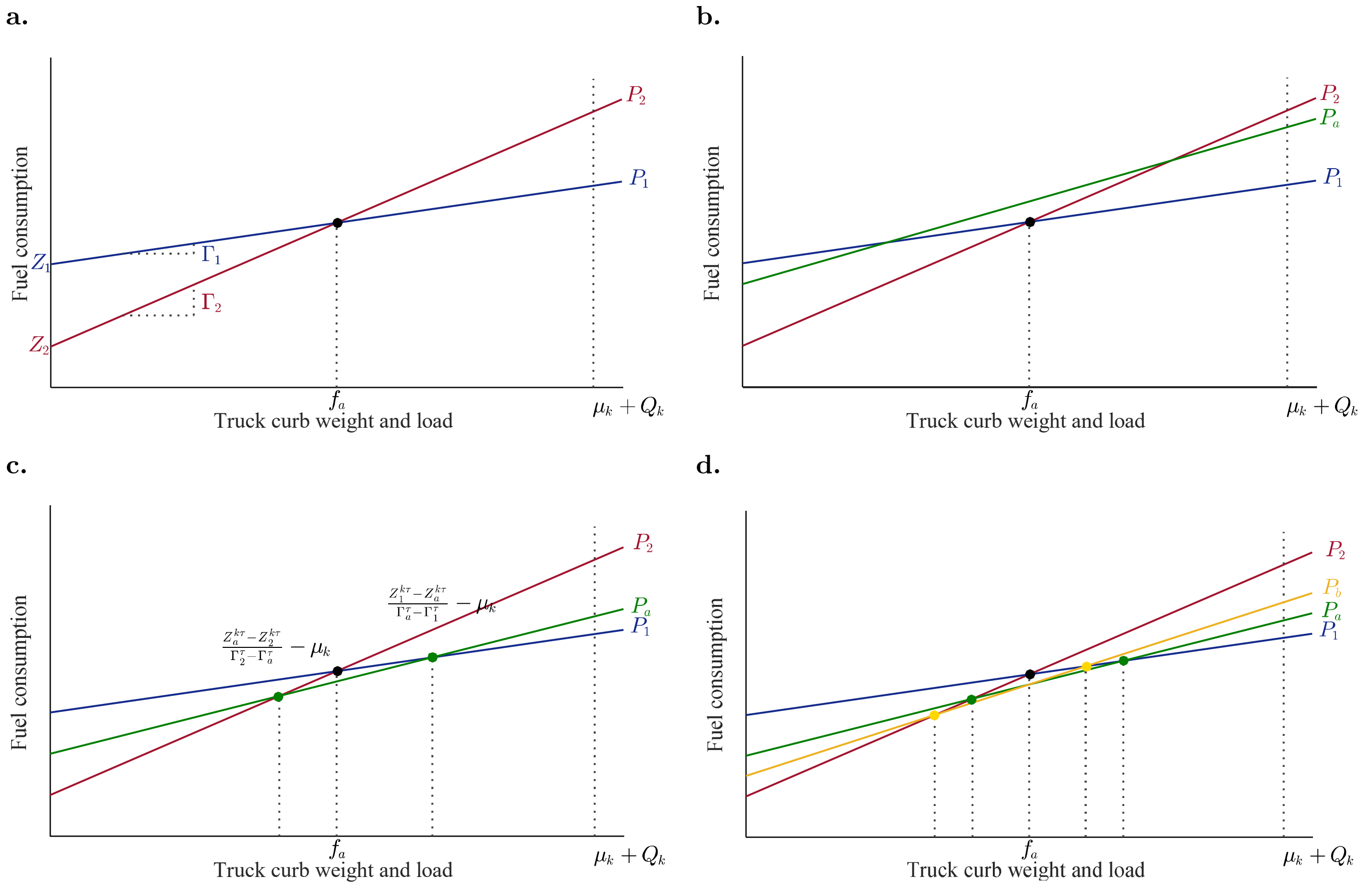

a.
Fuel consumption
P_2
P_1
Z_1
Γ_1
Γ_2
Z_2
f_a
Truck curb weight and load
$\mu_k + Q_k$
b.
Fuel consumption
P_2
P_a
P_1
f_a
Truck curb weight and load
$\mu_k + Q_k$
c.
Fuel consumption
P_2
$\frac{Z_1^{k\tau} - Z_a^{k\tau}}{\Gamma_a^\tau - \Gamma_1^\tau} - \mu_k$
P_a
P_1
$\frac{Z_a^{k\tau} - Z_2^{k\tau}}{\Gamma_2^\tau - \Gamma_a^\tau} - \mu_k$
f_a
Truck curb weight and load
$\mu_k + Q_k$
d.
Fuel consumption
P_2
P_b
P_a
P_1
f_a
Truck curb weight and load
$\mu_k + Q_k$

Figure 2-2 a. Paths p_1 and p_2 and their attributes, **b.** an ineligible path, **c.** an eligible path, and **d.** two different eligible paths

In order to use Theorem 2 in the development of the PEP, we need to generalise it to derive efficient progression and exit conditions for the algorithm:

Corollary 2 *Assume that for a given departure time τ, paths p_i and p_j are two eligible paths, such that $p_i \not\equiv p_j$. Then, if a distinct path p_a is the fuel consumption minimising path for load level $f_a = \frac{Z_i^{k\tau} - Z_j^{k\tau}}{\Gamma_j^{\tau} - \Gamma_i^{\tau}} - \mu_k$, it is an eligible path.*

Proof. A proof similar to the one used for corollary 1 can be employed. □

With these results, the PEP is given in Algorithm 2-1. In this algorithm, the core operation is assigned to the function $(p_{\mathtt{p}}, ATT_{\mathtt{p}}) := TDFCMP\ (i, j, \tau, k, f)$, that takes as input the origin and destination nodes (i, j), the departure time (τ), the truck type (k) and the load carried by the truck (f), and outputs the time-dependent fuel consumption minimising path $(p_{\mathtt{p}})$ and its attributes $(ATT_{\mathtt{p}})$, comprising $t_{\mathtt{p}}^{\tau}, Z_{\mathtt{p}}^{k\tau}$ and $\Gamma_{\mathtt{p}}^{\tau}$, under the given settings. In our implementation, *TDFCMP* is based on a modified extension of the time-dependent shortest-path algorithm of Ziliaskopoulos and Mahmassani(1993).

In the beginning of the algorithm (line 2) the set $\mathcal{R}_{ij}^{k\tau}$, $\mathcal{ATT}_{ij}^{k\tau}$, UP, and IL are initialised as empty sets to retain eligible paths (for departure time τ, vehicle type k), their attributes, untreated points, and intersecting lines, respectively. In lines 3 and 4 of the algorithm, the fuel consumption minimising paths for the full and the empty truck are respectively found, and then are compared with each other in line 5. If these two paths are the same, then only one of them is retained and the algorithm is terminated (Proposition 1). Otherwise, the algorithm computes and stores a new untreated point and an intersecting lines pair and goes to line 7. In lines 7 to 15, every time a new untreated point is pulled out from the front of the UP, and until UP is not empty the operations of these lines are repeated.

Assuming that $|\mathcal{p}|$ is the cardinality of $\mathcal{R}_{ij}^{k\tau}$, this algorithm must make a maximum of $2|\mathcal{p}| - 1$ calls to the *TDFCMP* function and hence is quite fast (note only that in the case of $|\wp| = 1$ two calls are required and not one). Another speeding up feature that is built in the proposed PEP algorithm is due to the use of the information from customers' time-windows and demands. Note that in line 1 of the algorithm instead of $\tau \in T$ the search space can be significantly reduced by using $\tau \in [e_i + s_i, min(l_i + s_i, l_j)]$, where $e_i + s_i$ is the earliest possible departure time, and $min(l_i + s_i, l_j)$ is the latest possible departure time from the origin node. Moreover, in lines 3 and 4, instead of using Q_k and 0 as input to *TDFCMP* to identify p_1 and p_2, respectively, we have used $Q_k - q_i$, and q_j. This is because even if upon departure from the depot the truck is fully loaded, its load over the path from i to j cannot exceed $Q_k - q_i$, and it is not going to be less than the demand of the destination customer, i.e. q_j. Note that we assume $q_0 = 0$, so this stays consistent when the origin node is the depot.

Algorithm 2-1 The PEP (phase I)

```
1  **Input** origin node $i \in N_R$, desination node $j \in N_R$, time instant $\tau \in [e_i + s_i, min(l_i + s_i, l_j)]$, vehicle type $k \in K$
2  Initialise $\mathcal{R}_{ij}^{k\tau} = \{\}$, $\mathcal{ATT}_{ij}^{k\tau} = \{\}$, $UP = \{\}$, and $IL = \{\}$
3  $(p_1, ATT_1) := TDFCMP(i, j, \tau, k, Q_k - q_i)$
4  $(p_2, ATT_2) := TDFCMP(i, j, \tau, k, q_j)$
5  **if** $p_1 \equiv p_2$ **then** $\mathcal{R}_{ij}^{k\tau} = \{p_1\}$ and $\mathcal{ATT}_{ij}^{k\tau} = \{ATT_1\}$ and go to line 15  **end if**
6  $\mathcal{R}_{ij}^{k\tau} = \{p_1, p_2\}$, $\mathcal{ATT}_{ij}^{k\tau} = \{ATT_1, ATT_2\}$, $f_1 = \frac{Z_1^{k\tau} - Z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_1^{\tau}} - \mu_k$, $UP = \{f_1\}$, and $IL = \{(p_1, p_2)\}$
7  **while** $UP$ is not empty **do**
8      Pull out the front element of $UP$ and denote it by $f_{active}$; also pull out the front pair in $IL$ and denote it by $(p_{\mathbb{i}}, p_{\mathbb{j}})$
9      $(p_a, ATT_a) := TDFCMP(i, j, \tau, k, f_{active})$
10     **if** $p_a \notin \mathcal{R}_{ij}^{k\tau}$ **then**,
11         Add $p_a$to $\mathcal{R}_{ij}^{k\tau}$,  $ATT_a$ to $\mathcal{ATT}_{ij}^{k\tau}$, and compute $f_{\mathbb{j}} = \frac{Z_a^{k\tau} - Z_{\mathbb{j}}^{k\tau}}{\Gamma_{\mathbb{j}}^{\tau} - \Gamma_a^{\tau}} - \mu_k$  and $f_{\mathbb{i}} = \frac{Z_{\mathbb{i}}^{k\tau} - Z_a^{k\tau}}{\Gamma_a^{\tau} - \Gamma_{\mathbb{i}}^{\tau}} - \mu_k$
12         Add $f_{\mathbb{j}}$ and $f_{\mathbb{i}}$ respectively to the end of $UP$; also add the pairs $(p_a, p_{\mathbb{j}})$ and $(p_{\mathbb{i}}, p_a)$ respectively to the end of $IL$
13     **end if**
14 **end while**
15 **return** $\mathcal{R}_{ij}^{k\tau}$, $\mathcal{ATT}_{ij}^{k\tau}$
```

In order to visualise the working of the proposed PEP algorithm, a step-by-step example is illustrated in Figure 2-3. In this figure, in the first step, Figure 2-3.a, p_1 is found (line 3 of the algorithm), and in the second step, Figure 2-3.b, p_2 is found (line 4 of the algorithm) and as it is not same as p_1, f_1 is calculated and added to the end of *UP* (line 5 of the algorithm). The pair (p_1, p_2) is also added to the end of *IL*. In the next step, Figure 2-3.c, f_1 is extracted from the beginning of *UP* (line 8 of the algorithm), and the fuel consumption minimising path for the truck at load f_1 is found (line 9 of the algorithm). Since a different path from p_1 and p_2, i.e. path p_3 is found, it is added to $\mathcal{R}_{ij}^{k\tau}$ and f_2 and f_3 are calculated and along with pairs (p_3, p_2) and (p_1, p_3) are added to the end of *UP* and *IL*, respectively (line 11 and 12 of the algorithm). In the next step, Figure 2-3.d, the active point is the first element in *UP*, i.e. f_2, and a new eligible path is found and the same operations as in the previous step are repeated. After this step, however, as no other new eligible path is found by examining all points in *UP* (Figure 2-3.e), the algorithm terminates and returns 4 distinct eligible paths, following a total of 7 calls to the *TDFCMP*. An interesting outcome of the algorithm is further shown in Figure 2-3.f, which implies it is possible to know exactly at what load ranges carried by the considered truck at the considered departure time, which path is optimal. In other words, the PEP can return also a piecewise linear function for fuel consumption based on payload.

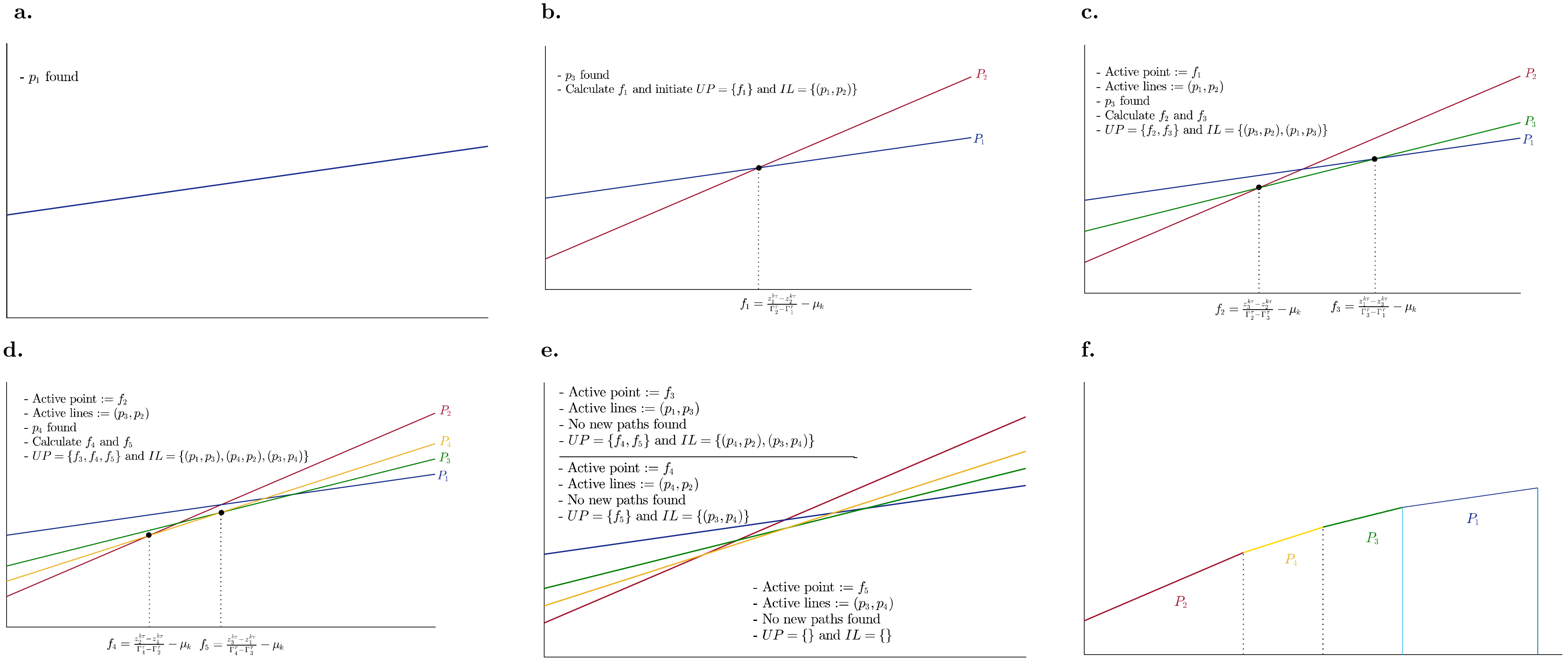

a.
- p_1 found
b.
- p_3 found
- Calculate f_1 and initiate $UP = \{f_1\}$ and $IL = \{(p_1, p_2)\}$
P_2
P_1
$f_1 = \frac{z_1^{k\tau} - z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_1^{\tau}} - \mu_k$
c.
- Active point := f_1
- Active lines := (p_1, p_2)
- p_3 found
- Calculate f_2 and f_3
- $UP = \{f_2, f_3\}$ and $IL = \{(p_3, p_2), (p_1, p_3)\}$
P_2
P_3
P_1
$f_2 = \frac{z_3^{k\tau} - z_2^{k\tau}}{\Gamma_2^{\tau} - \Gamma_3^{\tau}} - \mu_k$
$f_3 = \frac{z_1^{k\tau} - z_3^{k\tau}}{\Gamma_3^{\tau} - \Gamma_1^{\tau}} - \mu_k$
d.
- Active point := f_2
- Active lines := (p_3, p_2)
- p_4 found
- Calculate f_4 and f_5
- $UP = \{f_3, f_4, f_5\}$ and $IL = \{(p_1, p_3), (p_4, p_2), (p_3, p_4)\}$
P_2
P_4
P_3
P_1
$f_4 = \frac{z_2^{k\tau} - z_4^{k\tau}}{\Gamma_4^{\tau} - \Gamma_2^{\tau}} - \mu_k$
$f_5 = \frac{z_3^{k\tau} - z_4^{k\tau}}{\Gamma_4^{\tau} - \Gamma_3^{\tau}} - \mu_k$
e.
- Active point := f_3
- Active lines := (p_1, p_3)
- No new paths found
- $UP = \{f_4, f_5\}$ and $IL = \{(p_4, p_2), (p_3, p_4)\}$
- Active point := f_4
- Active lines := (p_4, p_2)
- No new paths found
- $UP = \{f_5\}$ and $IL = \{(p_3, p_4)\}$
- Active point := f_5
- Active lines := (p_3, p_4)
- No new paths found
- $UP = \{\}$ and $IL = \{\}$
f.
P_1
P_3
P_4
P_2

Figure 2-3 PEP steps

The constructed set $\mathcal{R}_{ij} = \{\mathcal{R}_{ij}^{k\tau}\colon \forall k \in K, \tau \in [e_i + s_i, \min(l_i + s_i, l_j)]\}$ after the application of the PEP-phase I, corresponds to the set $\mathcal{D}_{ij}$, and based on Theorem 1, it must be expanded to $\mathcal{E}_{ij}$. However, in most cases no further attempt is required for this expansion as the retained set is already equivalent to the set $\mathcal{E}_{ij}$. This is mainly because the fastest path already exists in $\mathcal{R}_{ij}^{\tau}$ ($\mathcal{R}_{ij}^{\tau} = \{\mathcal{R}_{ij}^{k\tau}\colon \forall k \in K\}$) in most cases (e.g. it is often observed that the fastest path is the fuel consumption minimising path for an empty light duty truck). In any case, for the sake of completeness any necessary further attempt must be identified and carried out in the second phase of the algorithm.

Assume that p_ℓ is the fastest path in the set $\mathcal{R}_{ij}^{\tau}$ after the application of the PEP phase-I for all vehicle types at time instant $\tau \in T$, and $p_{\mathbb{t}}$ is the globally fastest path at this time instant. Let the ordered set $\mathcal{M}_{ij}^{\tau} = \{p_{\mathbb{t}}, p_{\mathbb{t}+1}, \ldots, p_{\mathbb{t}+k}\}$ be the set of the k fastest paths at time instant τ, such that $\mathscr{t}^{\tau}(p_{\mathbb{t}+k}) < \mathscr{t}^{\tau}(p_\ell)$. Then, none of the paths in the set $\mathcal{M}_{ij}^{\tau}$ are dominated by the paths in $\mathcal{R}_{ij}^{\tau}$ because of their first element in the vector $[\mathscr{t}^{\tau}(p_{ij}), \mathscr{f}_{f}^{k\tau}(p_{ij})]$ (refer back to Theorem 1). However, lower ranked paths in the set $\mathcal{M}_{ij}^{\tau}$ might be dominated by higher ranked paths in this set in terms of fuel consumption; hence, this set could be further refined using the following strong dominance rule:

Proposition 2 *At time instant τ, a path $p_{\mathbb{i}}$ in $\mathcal{M}_{ij}^{\tau}$ is not dominated by the higher ranked path $p_{\mathbb{i}-1}$ in $\mathcal{M}_{ij}^{\tau}$ iff $\exists k \in K\colon \mathscr{f}_{0}^{k\tau}(p_{\mathbb{i}}) < \mathscr{f}_{0}^{k\tau}(p_{\mathbb{i}-1})$ and/or $\exists k \in K\colon \mathscr{f}_{Q_k}^{k\tau}(p_{\mathbb{i}}) < \mathscr{f}_{Q_k}^{k\tau}(p_{\mathbb{i}-1})$.*

Proof. The forward statement is obvious and requires no proof; that is, if there is at least one truck $k \in K$ that prefers path $p_{\mathbb{i}}$ over path $p_{\mathbb{i}-1}$when it is empty and/or when it is fully loaded, then $p_{\mathbb{i}}$ is not dominated by $p_{\mathbb{i}-1}$. Yet, we need to prove the backward statement; i.e.: path $p_{\mathbb{i}}$ is not dominated by path $p_{\mathbb{i}-1} \Rightarrow \exists k \in K\colon \mathscr{f}_{0}^{k\tau}(p_{\mathbb{i}}) < \mathscr{f}_{0}^{k\tau}(p_{\mathbb{i}-1})$ and/or $\exists k \in K\colon \mathscr{f}_{Q_k}^{k\tau}(p_{\mathbb{i}}) < \mathscr{f}_{Q_k}^{k\tau}(p_{\mathbb{i}-1})$: Recall that the domination rule established in Theorem 1 is based on the vector $[\boldsymbol{t}^{\tau}(p_{\ell}), \boldsymbol{f}_{f}^{k\tau}(p_{\ell})]$, for at least one $\tau \in T, k \in K$, and $f \in [0, Q_k]$. At time instant τ, from the definition of $\mathcal{M}_{ij}^{\tau}$ we know that $\boldsymbol{t}(p_{\ell-1}^{\tau}) < \boldsymbol{t}(p_{\ell}^{\tau})$; therefore, for path p_{ℓ} to be not dominated by $p_{\ell-1}$, we must have $\boldsymbol{f}_{f}^{k\tau}(p_{\ell}) < \boldsymbol{f}_{f}^{k\tau}(p_{\ell-1})$ for at least one $k \in K$, *and* $f \in [0, Q_k]$, and this is equivalent to (2-13) below:

$$Z_{\ell}^{k\tau} + \mathit{\Gamma}_{\ell}^{\tau}(\mu_k + f) < Z_{\ell-1}^{k\tau} + \mathit{\Gamma}_{\ell-1}^{\tau}(\mu_k + f) \tag{2-13}$$

Now, to use a proof by contradiction we assume that path p_{ℓ} is not preferred over path $p_{\ell-1}$neither when truck k is empty, nor when it is fully loaded; that is:

$$Z_{\ell-1}^{k\tau} + \mathit{\Gamma}_{\ell-1}^{\tau}\mu_k < Z_{\ell}^{k\tau} + \mathit{\Gamma}_{\ell}^{\tau}\mu_k \tag{2-14}$$

$$Z_{\ell-1}^{k\tau} + \mathit{\Gamma}_{\ell-1}^{\tau}(\mu_k + Q_k) < Z_{\ell}^{k\tau} + \mathit{\Gamma}_{\ell}^{\tau}(\mu_k + Q_k) \tag{2-15}$$

The combination of (2-13) and (2-14) yields that $\mathit{\Gamma}_{\ell}^{\tau} < \mathit{\Gamma}_{\ell-1}^{\tau}$, while the combination of (2-13) and (2-15) yields $\mathit{\Gamma}_{\ell}^{\tau} > \mathit{\Gamma}_{\ell-1}^{\tau}$ which is a contradiction. □

Hence, the second phase of the PEP is presented in Algorithm 2-2. Note that in line 3 of the algorithm, a k-fastest path algorithm, that takes $\boldsymbol{t}^{\tau}(p_{\ell})$ as input, must be used. This algorithm begins by finding the time-dependent fastest path, and loops for the k fastest path where $k = \infty$ or any large number, and breaks out of the loop once the last path found has a travel time greater than or equal to $\boldsymbol{t}^{\tau}(p_{\ell})$.

Algorithm 2-2 The PEP (phase II)

```
1   Input origin node $i \in N_R$, desination node $j \in N_R$, time instant $\tau \in [e_i + s_i, \min(l_i + s_i, l_j)]$,
    $p_\ell$, $\boldsymbol{t}^\tau(p_\ell)$ , $\mathcal{R}^\tau_{ij}$
2   $\mathcal{E}^\tau_{ij} \leftarrow \mathcal{R}^\tau_{ij}$
3   Construct the set $\mathcal{M}^\tau_{ij} = \{p_{\mathbb{t}}, p_{\mathbb{t}+1}, \ldots, p_{\mathbb{t}+k}\}$, such that $\boldsymbol{t}^\tau(p_{\mathbb{t}+k}) < \boldsymbol{t}^\tau(p_\ell)$
4   if $\mathcal{M}^\tau_{ij} = \{\}$ then go to line 15  end if
5   Pull ou the front element of $\mathcal{M}^\tau_{ij}$, denote it by $p_{i-1}$ and add it to the end of $\mathcal{E}^\tau_{ij}$
6   while $\mathcal{M}^\tau_{ij}$ is not empty do
7       Pull out the front element of $\mathcal{M}^\tau_{ij}$ and call it $p_i$
8       for $k =$ 1 to $\mathbb{k}$ do
9           if $\cancel{f}^{k\tau}_0(p_i) < \cancel{f}^{k\tau}_0(p_{i-1})$ or $\cancel{f}^{k\tau}_{Q_k}(p_i) < \cancel{f}^{k\tau}_{Q_k}(p_{i-1})$ then
10              $p_{i-1} \leftarrow p_i$ and add $p_i$ to the end of $\mathcal{E}^\tau_{ij}$
11              break
12          end if
13      end for
14  end while
15  return $\mathcal{E}^\tau_{ij}$
```

The output of the PEP is a set of retained road-paths between the required nodes with a complete archive of their distance, time-dependent travel times, UTM and RTM attributes in easy-to-access look-up tables, which greatly facilitate the application of any subsequent solution algorithm. However, we still need to generalise the results for the multi-objective case of the SPRP.

Indeed, thanks to the second phase expansion based on paths' travel times, the proposed PEP can be already generalised to the multi-objective case of the SPRP. Let $\hat{G} = (\hat{N}, \hat{A})$ be the resulting multi-graph after the application of the PEP, where $\hat{N} = \{N_0 \cup N_1\}$, and $\hat{A}$ is the set of retained directed road-paths between the nodes, i.e. $\hat{A} = \{(i, j, p) | i, j \in \hat{N}, p \in \mathcal{E}\}$. Then, we propose the following theorem:

Theorem 3 *Let $\mathcal{PF}$ be the POS of any instance of the SPRP solved on the reduced graph $\hat{G}$, and $\mathcal{PF}^*$ be the POS of the very SPRP instance solved directly on G. Then, $\mathcal{PF} \equiv \mathcal{PF}^*$.*

Proof. We must prove that no Pareto optimal path is discarded from $\hat{G}$ by applying the PEP. Consider a proof by contradiction and suppose that at least for one given departure time instant τ, and a vehicle of type k carrying a load of size f, road-path $p_a \in G$ between required nodes $i, j \in N_R$, which is discarded from $\hat{G}$ by applying the PEP (i.e. $p_a \notin \hat{G}$), is a Pareto optimal path, and its corresponding non-dominated objective value in the criterion space is $\mathscr{u}^a = (\mathscr{u}_1^a, \mathscr{u}_2^a, \mathscr{u}_3^a)$. The fact that p_a is excluded from $\hat{G}$ implies that $p_a \notin \mathcal{E}_{ij}^{\tau}$ and hence the vector $[\mathscr{t}^{\tau}(p_a), \mathscr{f}_f^{k\tau}(p_a)]$ is a dominated vector. On the other hand, since $\mathscr{u}_2^a = \mathscr{f}_f^{k\tau}(p_a)$ and $\mathscr{u}_3^a = \mathscr{t}^{\tau}(p_a)$, the only way for $\mathscr{u}^a$ to be a non-dominated vector is to be non-dominated based on $\mathscr{u}_1^a$. However, the vehicle hiring cost objective is path-independent. □

2.4.1 The PEP-based MILP for the SPRP

A MILP formulation of the SPRP based on the PEP, which is equivalent to a multi-objective, time and load dependent, fleet size and mix PRP, with time windows, flexible departure times, and multi-trips is proposed.

Prior to introducing the decision variables and the model, however, to reduce computational complexity, we need to describe an alternative discretisation of the planning horizon T, independently for each road-path $(i,j,p) \in \hat{A}$. As we discussed earlier the time-dependent travel time of a road-link (and hence a road-path) is assumed integer. With this assumption it is probable to have the same travel time at several consecutive departure times. For example, it is possible that if the truck departs the origin of the road-path at any of the consecutive minutes $\{\tau, \tau+1, \tau+2, .., \tau+\ell\} \in T$, the travel times would be equal; i.e. $\boldsymbol{t}_{ijp}^{\tau} = \boldsymbol{t}_{ijp}^{\tau+1} = \boldsymbol{t}_{ijp}^{\tau+2} = \cdots = \boldsymbol{t}_{ijp}^{\tau+\ell}$. Correspondingly, the UTM and the RTM attributes would be the same. Therefore, the whole set of these 'time instants' might be bundled together as a 'time period' $\mathbb{t}$, to which a unique $\boldsymbol{t}_{ijp}^{\mathbb{t}}$, $Z_{ijp}^{k\mathbb{t}}$ and $\mathit{\Gamma}_{ijp}^{\mathbb{t}}$, can be attributed. In other words, departing at any time instant τ during time period $\mathbb{t}$, will yield $\boldsymbol{t}_{ijp}^{\mathbb{t}}$, $Z_{ijp}^{k\mathbb{t}}$ and $\mathit{\Gamma}_{ijp}^{\mathbb{t}}$.

With this explanation, the planning horizon T could be discretised independently for each road-path $(i,j,p) \in \hat{A}$ (using a customised notation $\mathcal{T}_{ijp}$), into a number of time periods $\boldsymbol{h}_{ijp}$, proportionate to the changes in the travel time of the path during the planning horizon. Therefore, the discretisation of $\mathcal{T}_{ijp}$ would yield (we drop path indices from $\boldsymbol{h}$ and $\mathcal{T}$ for notational simplicity), $\mathcal{T} = \{[\boldsymbol{a}_{ijp}^{1}, \boldsymbol{b}_{ijp}^{1}], [\boldsymbol{a}_{ijp}^{2}, \boldsymbol{b}_{ijp}^{2}], \ldots, [\boldsymbol{a}_{ijp}^{\boldsymbol{h}}, \boldsymbol{b}_{ijp}^{\boldsymbol{h}}]\}$, where $\boldsymbol{a}_{ijp}^{1} = e_0, \boldsymbol{b}_{ijp}^{\boldsymbol{h}} = l_0$, and $\boldsymbol{a}_{ijp}^{\mathbb{t}}$ and $\boldsymbol{b}_{ijp}^{\mathbb{t}}$ denote the lower boundary and the upper boundary of time period $\mathbb{t} \in \mathcal{T}$, respectively. Confining this discretisation further by using the information from time-windows, it is possible to impose that $\boldsymbol{a}_{ijp}^{1} = e_i + s_i$ and $\boldsymbol{b}_{ijp}^{\boldsymbol{h}} = min(l_i + s_i, l_j)$, $\forall (i,j,p) \in \hat{A} \mid i \neq 0$. The following decision variables are then introduced and used by the formulation: (i) the binary variable $x_{ijp}^{k\mathbb{t}}$, which is equal to 1 iff vehicle $k \in K$ departs node $i \in \hat{N}$ during time period $\mathbb{t} \in \mathcal{T}$ to go to node $j \in \hat{N}$, through road-path $(i, j, p) \in \hat{A}$, (ii) the continuous variable $f_{ijp}^{k\mathbb{t}} \in [0, Q_k]$ which represents the size of load carried by vehicle $k \in K$ over the road-path $(i, j, p) \in \hat{A}$ during time period $\mathbb{t}$, and (iii) the integer variable $y_{ijp}^{k\mathbb{t}}$, which indicates the exact departure time from the origin of path $(i, j, p) \in \hat{A}$ given that it is departed by vehicle $k \in K$ during time period $\mathbb{t}$. Note that $f_{ijp}^{k\mathbb{t}}$ must be time-indexed as it is multiplied by the RTM component in the instantaneous CMEM which is time-dependent.

In order to allow vehicles to make multiple trips, assuming that each vehicle is allowed to make a maximum of $\mathscr{r}$ trips during the planning horizon, $\mathscr{r}$ copies of the set K is added to its end. With this modification, the length of the set K will be $\mathscr{r}\vartheta$ and vehicles $\{k + \vartheta, k + 2\vartheta, \ldots, k + \mathscr{r}\vartheta\}$ all are the dummy copies of vehicle $k \in K$, but with no assignment cost. It is worth mentioning that the definition of a fixed set of vehicles follows two main reasons and has no contradiction with defining the problem as a fleet size and mix problem; first, the use of the fleet set in the formulation of the problem adds to its generality as it could be simply used for the case of a heterogeneous or homogeneous fixed size fleet as well, and second, it helps multi-trip scheduling. It is clear that if a large enough number of each vehicle type is included in the fleet the problem is a fleet size and mix problem.

The mathematical formulation of the proposed problem is given by (2-16)-(2-28).

$$z_1 := \sum_{(0,i,p)\in\hat{A}} \sum_{k\in K} \sum_{\mathbb{t}\in\mathcal{T}} c_k x_{0ip}^{k\mathbb{t}} \tag{2-16}$$

$$\begin{aligned} z_2 := & \sum_{(i,j,p)\in\hat{A}} \sum_{k\in K} \sum_{\mathbb{t}\in\mathcal{T}} Z_{ijp}^{k\mathbb{t}} x_{ijp}^{k\mathbb{t}} \\ & + \sum_{(i,j,p)\in\hat{A}} \sum_{\mathbb{t}\in\mathcal{T}} \Gamma_{ijp}^{\mathbb{t}} \big(\sum_{k\in K} x_{ijp}^{k\mathbb{t}} \mu_k + \sum_{k\in K} f_{ijp}^{k\mathbb{t}}\big) \end{aligned} \tag{2-17}$$

$$z_3 := \sum_{(i,0,p)\in\hat{A}} \sum_{k\in K} \sum_{\mathbb{t}\in\mathcal{T}} (y_{i0p}^{k\mathbb{t}} + \boldsymbol{t}_{i0p}^{\mathbb{t}} x_{i0p}^{k\mathbb{t}}) - \sum_{(0,i,p)\in\hat{A}} \sum_{k\in K} \sum_{\mathbb{t}\in\mathcal{T}} y_{0ip}^{k\mathbb{t}} \tag{2-18}$$

$$Min\ (z_1, z_2, z_3) \tag{2-19}$$

Subject to:

$$\sum_{j\in\hat{N}} \sum_{p\in\mathcal{E}_{ij}} \sum_{k\in K} \sum_{\mathbb{t}\in\mathcal{T}} x_{ijp}^{k\mathbb{t}} = 1, \qquad \forall i \in N_1 \tag{2-20}$$

$$\sum_{j\in\hat{N}} \sum_{p\in\mathcal{E}_{ij}} \sum_{\mathbb{t}\in\mathcal{T}} (x_{ijp}^{k\mathbb{t}} - x_{jip}^{k\mathbb{t}}) = 0, \qquad \forall k \in K, i \in \hat{N} \tag{2-21}$$

$$\sum_{j\in\hat{N}} \sum_{p\in\mathcal{E}_{ij}} \sum_{k\in K} \sum_{\mathbb{t}\in\mathcal{T}} (f_{jip}^{k\mathbb{t}} - f_{ijp}^{k\mathbb{t}}) = q_i, \qquad \forall i \in N_1 \tag{2-22}$$

$$q_j x_{ijp}^{k\mathbb{t}} \le f_{ijp}^{k\mathbb{t}} \le (Q_k - q_i) x_{ijp}^{k\mathbb{t}}, \qquad \forall (i,j,p) \in \hat{A}, k \in K, \mathbb{t} \in \mathcal{T} \tag{2-23}$$

$$\boldsymbol{a}_{ijp}^{\mathbb{t}} x_{ijp}^{k\mathbb{t}} \le y_{ijp}^{k\mathbb{t}} \le \boldsymbol{b}_{ijp}^{\mathbb{t}} x_{ijp}^{k\mathbb{t}}, \qquad \forall (i,j,p) \in \hat{A}, k \in K, \mathbb{t} \in \mathcal{T} \tag{2-24}$$

$$\begin{aligned} &\sum_{p\in\mathcal{E}_{ij}} \sum_{\mathbb{t}\in\mathcal{T}} (y_{ijp}^{k\mathbb{t}} + \boldsymbol{t}_{ijp}^{\mathbb{t}} x_{ijp}^{k\mathbb{t}}) - \sum_{\jmath\in\hat{N}} \sum_{p\in\mathcal{E}_{j\jmath}} \sum_{\mathbb{t}\in\mathcal{T}} (y_{j\jmath p}^{k\mathbb{t}} - s_j x_{j\jmath p}^{k\mathbb{t}}) \\ &\qquad \le 1 - \sum_{p\in\mathcal{E}_{ij}} \sum_{\mathbb{t}\in\mathcal{T}} x_{ijp}^{k\mathbb{t}}, \qquad \forall i \in \hat{N}, j \in N_1, k \in K \end{aligned} \tag{2-25}$$

$$\begin{aligned} &\sum_{j\in\hat{N}} \sum_{p\in\mathcal{E}_{ij}} \sum_{k\in K} \sum_{\mathbb{t}\in\mathcal{T}} (y_{ijp}^{k\mathbb{t}} - s_i x_{ijp}^{k\mathbb{t}}) \\ &\qquad = max(\sum_{j\in\hat{N}} \sum_{p\in\mathcal{E}_{ij}} \sum_{k\in K} \sum_{\mathbb{t}\in\mathcal{T}} (y_{jip}^{k\mathbb{t}} + \boldsymbol{t}_{jip}^{\mathbb{t}} x_{jip}^{k\mathbb{t}}), e_i), \quad \forall i \\ &\qquad \in N_1 \end{aligned} \tag{2-26}$$

$$\sum_{j\in\hat{N}} \sum_{p\in\mathcal{E}_{ij}} \sum_{\mathbb{t}\in\mathcal{T}} x_{0jp}^{k\mathbb{t}} - x_{0jp}^{(k+\vartheta)\mathbb{t}} \ge 0, \qquad \forall k \in \{1, ..(\boldsymbol{r} - 1)\vartheta\} \tag{2-27}$$

$$\begin{aligned} &\sum_{j\in\hat{N}} \sum_{p\in\mathcal{E}_{ij}} \sum_{\mathbb{t}\in\mathcal{T}} y_{0jp}^{(k+\vartheta)\mathbb{t}} - (y_{j0p}^{k\mathbb{t}} + (\boldsymbol{t}_{j0p}^{\mathbb{t}} + s_0) x_{j0p}^{k\mathbb{t}}) \\ &\qquad \ge M(\sum_{j\in\hat{N}} \sum_{p\in\mathcal{E}_{ij}} \sum_{\mathbb{t}\in\mathcal{T}} x_{0jp}^{(k+\vartheta)\mathbb{t}} - 1), \forall k \in \{1, ..(\boldsymbol{r} - 1)\vartheta\} \end{aligned} \tag{2-28}$$

Expressions (2-16) to (2-18) are the objective functions, constraints (2-20) and (2-21) are routing constraints, constraints (2-22) and (2-23) are capacity and load flow constraints, constraints (2-24) to (2-26) are scheduling constraints, and constraints (2-27) and (2-28) are multi-trip constraints. The first objective function (2-16) represents the total hiring cost of the trucks assigned to the routes. Note again that $c_k = 0, \forall\{k \in K | k > \vartheta\}$, so that trucks operating extra rounds of trips are not penalised more than once. The second objective function (2-17) estimates the total time, load and truck type dependent fuel consumption of the routes; and the third objective (2-18) represents the total duration of each truck route.

Constraints (2-20) indicate that each customer must be visited exactly once for delivery. Constraints (2-21) guarantee that the same vehicle that enters each customer node also exits the node. Constraints (2-22) model the flow on each road-path. Constraints (2-23) are used to restrict the total load a vehicle carries by its capacity. Constraints (2-24) determine the time-period during which the origin of a path must be departed. Note that with these constraints $y_{ijp}^{k\mathbb{t}}$ variables will be a non-negative integer just for one time period, and zero for all other periods. Constraints (2-25) tune the time-dependent travel time of each road-path based on the departure time from its origin. Constraints (2-26) indicate that a customer is departed upon the completion of the service. It is worth noting that through the variable domain definition, we have already implicitly imposed that service takes place within the customers' time windows. Constraints (2-27) and (2-28) are multi-trip constraints and together ensure that vehicles could operate another round of trip only if they are back from their first trip and are reloaded for a new one. In (2-28), M is a sufficiently large number, and without loss of generality could be set equal to $l_0 + s_0$.

2.4.2 Alternative approximate extensions of the PEP

Exploiting the tight conditions set by Theorem 2, and given the fact that the PEP phase-I is usually sufficient for the identification of all Pareto optimal paths, efficient alternative approximate path elimination algorithms could be developed based on pre-mature termination of the PEP. Here we propose three alternative approximate path elimination algorithms which are as simple as a TDSP algorithm to implement (but much better performing) and faster than the PEP. We call these alternative algorithms *Alt1*, *Alt2* and *Alt3*:

- *Alt1.* This algorithm only finds the TDSP paths, along with the time-dependent fuel consumption minimising paths for all truck types in two modes of empty and full, and stops.
- *Alt2.* An unprovable but intuitive heterogeneous fleet extension of Proposition 1 is to think that if path p_1 is a fuel consumption minimising path for both a fully loaded heavy-duty truck, and an empty light-duty truck at departure time τ, then this path might be optimal in terms of fuel consumption for any other vehicle type with any size of load. Based on this, an alternative algorithm can be the one that finds only the TDSP paths, along with the time-dependent fuel consumption minimising paths for a fully loaded heavy-duty truck, and an empty light-duty truck, and stops.
- *Alt3.* This algorithm is a less conservative extension of *Alt2* where we hypothesize that if path p_1 is both the fastest path and the fuel consumption minimising path for a fully loaded heavy-duty truck at departure time τ, then this path might be optimal in terms of fuel consumption for any other vehicle type with any size of load. Hence, this algorithm only finds the TDSP paths and the time-dependent fuel consumption minimising paths for a fully loaded heavy-duty truck.

The performance of these algorithms against the exact algorithm is reported in section 2.7 of the chapter.

2.5 Generating the full set of the ND points to the SPRP

It is well-known that *Multi-Objective Combinatorial Optimisation (MOCO)*, dealing with *Multi-Objective Integer and Mixed Integer Linear Programming (MOILP/MOMILP)* problems, is much more difficult than the *Multi-Objective*

Linear Programming (MOLP), since the feasible set is no longer convex, and unsupported ND points may exist. Hence, even if a complete parameterization on the weight of each objective is attempted, unlike in the MOLP, the ND solution set of the problem cannot be fully determined (Alves & Clı´maco, 2007). Some of these methodological difficulties can be easier overcome in the bi-objective case than in the multi-objective one (Alves & Clı´maco, 2007), and this is mainly the reason why most of the solution methodologies in the literature focus on a bi-objective case.

Noticeable developments have taken place recently in the area of *Mathematical Programming Techniques (MPTs)* for the exact solution of tri- and multi-objective integer programming problems, and efficient algorithms have been proposed that can find the full set of the ND solutions, saving greatly in the total number of IPs required to be solved (Sylva & Crema, 2004; Özlen & Azizoglu, 2009; Lokman & Köksalan, 2013; Özlen, Burton, & MacRae, 2014; Boland, Charkhgard, & Savelsbergh, 2016; Boland, Charkhgard, & Savelsbergh, 2017). These algorithms, however, cannot be directly applied on the MOMILP problems as infinite number of ND solutions can lie (e.g. on a line segment) in the continuous parts of the solution space to an MOMILP, and the current state-of-the-art in tackling MOMILPs only allows the consideration of two objectives within a branch-and-bound scheme. However, given the characteristics of the SPRP, the following useful remark allows us to apply directly any successful MPT developed for MOILPs on our problem:

***Remark** 3 If in an MOMILP continuous variables only appear in at most one of the objective functions, then the given MOMILP has a discrete ND frontier and there is no continuous part in the ND frontier. For such MOMILP, since the ND frontier is discrete, the methods developed for pure MOIPs can be used to find all ND points.*

This remark is based on the fact that the projection of the image of a feasible solution to a multi-objective problem in the criterion space with continuous variables in only one of the objectives, is a point, and no line segment can exist along the Pareto frontier (assume a bi-objective problem for visualisation). Therefore, based on this remark, since in the case of the SPRP continuous variables only appear in the objective function related to the fuel consumption minimisation, efficient MPTs for MOILPs are applicable.

In order to justify our choice of the best technique for the purpose of our paper, in Table 2-3 a review of the most efficient criterion space search algorithms for solving MOILPs is given, along with the bounds on the number of IPs required to be solved by them. Observe that in this table $\mathcal{Y}_{\mathcal{N}}$ is the set of non-dominated points. For a concise description of the methods in this table we refer to Boland et al. (2017).

Table 2-3

Existing efficient criterion space search MPTs for finding the full set of all ND solutions of the MOILPs

Method	Bounds
Sylva & Crema's method (SCM): (Sylva & Crema, 2004)	$\vert\mathcal{Y}_{\mathcal{N}}\vert + 1$
The *enhanced recursive method (ERM)*: (Özlen et al., 2014)	$\mathcal{O}(\vert\mathcal{Y}_{\mathcal{N}}\vert^2)$
The *full p-split method (FPS)*: (Dächert & Klamroth, 2015; Dhaenens et al., 2010)	$3\vert\mathcal{Y}_{\mathcal{N}}\vert + 1$
The *full $(p-1)$-split method (FP-1S)*: (Kirlik & Sayın, 2014; Lokman & Köksalan, 2013)	$\mathcal{O}(\vert\mathcal{Y}_{\mathcal{N}}\vert^2)$
The L-shape search method (LSM): (Boland et al., 2016)	$\mathcal{O}(\vert\mathcal{Y}_{\mathcal{N}}\vert^2)$
The *quadrant shrinking method (QSM)*: (Boland et al., 2017a)	$3\vert\mathcal{Y}_{\mathcal{N}}\vert + 1$

In selecting the most appropriate MPT for solving the SPRP, a trade-off between solving a small number, but increasingly difficult single-objective IPs, and solving a larger number, but manageable size single-objective IPs must be made (Boland et al., 2016). While the best-known bound is due to the SCM, the IPs to be solved in the SCM soon become intractable in hard optimisation problems. Hence, the next best existing algorithm in this regard is the QSM, which outperforms the FPS in terms of the number of infeasible IPs solved. Therefore, we find the QSM as the most appealing choice due to its ease of implementation and competitive bound on the number of IPs to be solved for the generation of the full set of the ND vectors.

For the sake of brevity, a full exposition of the QSM method is avoided here and the reader is referred to Boland et al. (2017a) for an introduction. In a nutshell, the QSM works in a projected 2D criterion space, defined by the first two objectives $z_1(x)$ and $z_2(x)$. The approach uses quadrants in the projected space, which are defined using an upper bound $u(u_1, u_2)$; i.e. $\mathcal{Q}(u) = \{\mathscr{y} \in \mathbb{R}^2 : \mathscr{y} \leq u\}$, where $\mathscr{y}$ is the projection of a point in the 2D space. The core operation of the QSM is searching for an as-yet-unknown ND point (if one exists) by solving two IPs, through a two-stage scalarisation technique. First, an intermediate point $x^i \in \mathcal{X}$ with minimal third objective value over points $x \in \mathcal{X}$ with $\bar{x} \leq u$ ($\bar{x}$ is the projection of x in the 2D space) is found via: $x^i \in \arg min\{z_3(x) : x \in \mathcal{X} \text{ } and \text{ } z_k(x) \leq u_k, k \in \{1{,}2\}\}$. If this IP is feasible, it is followed by a second IP that converts the weakly efficient solution x^i into an efficient solution x^*: $x^* \in \arg min\{\sum_{k=1}^{3} z_k(x) \colon x \in \mathcal{X} \text{ } and \text{ } z_k(x) \leq z_k(x^i), k \in \{1{,}2{,}3\}\}$. This search is denoted by *2D-NDP-Search(u)*, and if the first IP is infeasible, *2D-NDP-Search(u)* returns *Null* and x^i does not exist. Otherwise, if x^i exists, the second IP must be feasible and *2D-NDP-Search(u)* returns x^*. Ultimately, this search returns a ND point z^* with z_3^* minimal over those $z \in \mathcal{Y}_{\mathcal{N}}$ with $\bar{z} \leq u$. Following the identification of z^*, any other ND point with the property that its projection is in $\mathcal{Q}(u)$ can now be found by searching for

ND points with the property that their projection is in either quadrant $\mathcal{Q}(u^1)$ or $\mathcal{Q}(u^2)$, with $u^1 = (u_1, z_2 - \epsilon_2)$ and $u^2 = (z_1 - \epsilon_1, u_2)$, where ϵ_1 and ϵ_2 are problem-dependent small positive constants. The search is carried out by finding ND points on the top and the right boundary of the quadrant, and when it is established that ND points can no longer be found using *2D-NDP-Search(u)*, the quadrant is shrunk, and the process repeats until it is shown that the quadrant does not contain any ND points. [refer to Boland et al. (2017a) for related proofs and the algorithm].

In order to solve the SPRP for the identification of the full set of ND solutions, we use the PEP-based MILP for the SPRP as the core optimisation problem inside QSM. The two (M)IPs that must be solved using the *2D-NDP-Search(u)* for returning a ND vector in each iteration of the QSM, correspond to (i) an IP which is a single-objective problem in the third objective of the SPRP, i.e. the total travel time of the tours, and (ii) a MIP which is an aggregation of all the three objectives of the SPRP. This hybridisation turns out to be very efficient, and as will be reported in the computational experiments section of the paper, we are able to find the POS of all instances considered over a reasonable computational time despite the difficulty of the problem.

2.6 Construction of realistic spatiotemporal driving cycles

As discussed earlier, with the current technological advancements in *Intelligent Transportation Systems (ITSs)* and GPSs it is possible to collect data on fine-grained speeds variations over any given road-link in the road network at different times of a given day using probe vehicles. However, these data are usually unavailable at the planning stage. On the other hand, it has been shown that lack of such data, especially the instantaneous A/D data, can cause inaccuracy in estimating fuel consumption and hence might lead to unreliable and misleading routing decisions (Turkensteen, 2017).

The only relevant study in the area of EMVRPs trying to take this situation into account is a recent study by Kancharla and Ramadurai (2018) who propose to incorporate driving cycles into the estimation of fuel consumption in a time-independent routing context. The authors collect 450 km (144 h of driving) GPS data in the city of Chennai, India, and then for each arc in their test graphs randomly combine the collected micro-trips through an iterative process until the distance of the intended arc is covered. Despite the effort that is put in collecting these data, a major shortcoming of their proposed approach lies in the fact that the spatial and temporal characteristics of the road-links in the graph are completely ignored. The location of a road-link in the roadway graph and the time of the day the given road-link is traversed have a fundamental impact on the shape of the speed-time profile and the frequency of vehicles A/Ds.

The spatiotemporal characteristics of a road-link, however, are very well reflected in the macroscopic time-dependent speed data, which are widely available for decision making and could be efficiently used for constructing reliable synthetic driving cycles. Developing microscopic traffic data from macroscopic traffic data based on reconstructed synthetic vehicle trajectories is not something new and is a well-known stream of research in transportation engineering (Silvas et al., 2016; Wang et al., 2011; Zegeye et al., 2013). However, in this paper we adopt a completely different operational research approach and propose a simple but reliable method for the generation of synthetic spatiotemporal driving cycles with using only the road-link distance and the time-dependent average speed as input, and with no parameter tuning. The proposed approach is validated against an extensive library of real-world driving cycles and the results are presented in the 'computational results' section of the paper.

Our proposed approach builds on a model for the generation of worst-case driving cycles, which is then simply weakened with slight parameter relaxation to lead to realistic cycles. Let d_{ij} be the distance of a given road-link $(i, j) \in A$, and

$\boldsymbol{t}_{ij}^{\mathbb{t}}$ be the time-dependent travel time of the given road-link during time period $\mathbb{t} \in \mathcal{T}$, deduced from the macroscopic traffic speed data using expression (2-3) (note that it is not necessary to generate the cycles for every time instant of the planning horizon; instead, in line with the arguments in section 2.4.1, we can generate cycles in the 'time period' level). Also, let $\mathcal{A}_{max}$ and $\mathcal{D}_{max}$ respectively denote the maximum possible acceleration and deceleration rates for a truck. Finally, suppose the maximum possible speed in the network is v_U. Then, the worst-case second-by-second A/D rates (a_t) for the given road link during time period $\mathbb{t}$ could be constructed by determining speed levels of every second $(\boldsymbol{v}_t)$ (where $a_t = \boldsymbol{v}_t - \boldsymbol{v}_{t-1}$) using the following nonlinear programming model:

$$Max \sum_{t=\boldsymbol{a}_{ij}^{\mathbb{t}}+1}^{\boldsymbol{a}_{ij}^{\mathbb{t}}+\boldsymbol{t}_{ij}^{\mathbb{t}}} |\boldsymbol{v}_t - \boldsymbol{v}_{t-1}| \tag{2-29}$$

Subject to:

$$\sum_{t=\boldsymbol{a}_{ij}^{\mathbb{t}}}^{\boldsymbol{a}_{ij}^{\mathbb{t}}+\boldsymbol{t}_{ij}^{\mathbb{t}}} \boldsymbol{v}_t = d_{ij}, \tag{2-30}$$

$$-\mathcal{D}_{max} \leq \boldsymbol{v}_t - \boldsymbol{v}_{t-1} \leq \mathcal{A}_{max}, \qquad \forall t \in \{\boldsymbol{a}_{ij}^{\mathbb{t}}+1, \boldsymbol{a}_{ij}^{\mathbb{t}}+2, \dots, \boldsymbol{a}_{ij}^{\mathbb{t}}+\boldsymbol{t}_{ij}^{\mathbb{t}}\} \tag{2-31}$$

$$\boldsymbol{v}_{\boldsymbol{a}_{ij}^{\mathbb{t}}} = \boldsymbol{v}_{\boldsymbol{a}_{ij}^{\mathbb{t}}+\boldsymbol{t}_{ij}^{\mathbb{t}}} = 0, \tag{2-32}$$

$$0 \leq \boldsymbol{v}_t \leq v_U, \qquad \forall t \in \{\boldsymbol{a}_{ij}^{\mathbb{t}}, \boldsymbol{a}_{ij}^{\mathbb{t}}+1, \dots, \boldsymbol{a}_{ij}^{\mathbb{t}}+\boldsymbol{t}_{ij}^{\mathbb{t}}\} \tag{2-33}$$

The nonlinear objective function (2-29) maximises the positive difference between speed levels of every two consecutive seconds of the cycle, and hence the A/D rates. Constraint (2-30) tunes the instantaneous speeds in a way that the cycle is completed within the estimated time-dependent travel time of the given road link. Constraints (2-31) ensure that the A/D rates do not violate the maximum possible A/D rate of the truck. Constraint (2-32) indicate that the truck accelerates from idle (departing node i) and comes to a full stop at the end of the cycle (arriving at node j). Note that this is based on the assumption that in an urban road network trucks are usually forced to reduce their speed significantly or come to a full stop at network junctions (e.g. at a cross road traffic light or a turning point). Finally, constraints (2-33) determine the range of speed values.

In order to linearize (2-29)-(2-33), we define two new non-negative continuous decision variables $ACC_t, DEC_t, \forall t \in \{\mathscr{a}_{ij}^{\mathbb{t}}+1, \mathscr{a}_{ij}^{\mathbb{t}}+2, \ldots, \mathscr{a}_{ij}^{\mathbb{t}}+\mathscr{t}_{ij}^{\mathbb{t}}\}$ which indicate the acceleration rates and the deceleration rates during second $t-1$ until t, respectively; and a new binary decision variable $\varkappa_t \in \{0{,}1\}, \forall t \in \{\mathscr{a}_{ij}^{\mathbb{t}}+1, \mathscr{a}_{ij}^{\mathbb{t}}+2, \ldots, \mathscr{a}_{ij}^{\mathbb{t}}+\mathscr{t}_{ij}^{\mathbb{t}}\}$, which is 1 iff vehicle accelerates during second $t-1$ until t, and 0 otherwise. Then, the following MILP which is called the Driving Cycle ($DC_{ij}^{\mathbb{t}}$) model hereafter can be developed:

$$DC_{ij}^{\mathbb{t}}\text{: } Max \sum_{t=\mathscr{a}_{ij}^{\mathbb{t}}+1}^{\mathscr{a}_{ij}^{\mathbb{t}}+\mathscr{t}_{ij}^{\mathbb{t}}} (ACC_t + DEC_t) \tag{2-34}$$

Subject to:

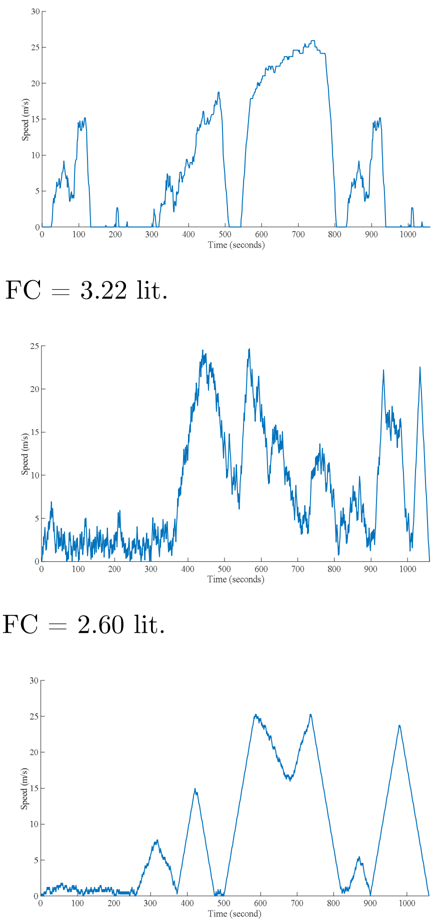
$$\mathscr{v}_t - \mathscr{v}_{t-1} = ACC_t - DEC_t, \qquad \forall t \in \{\mathscr{a}_{ij}^{\mathbb{t}}+1, \mathscr{a}_{ij}^{\mathbb{t}}+2, \ldots, \mathscr{a}_{ij}^{\mathbb{t}}+\mathscr{t}_{ij}^{\mathbb{t}}\} \tag{2-35}$$

$$0 \le ACC_t \le \varkappa_t \mathcal{A}_{max}, \qquad \forall t \in \{\mathscr{a}_{ij}^{\mathbb{t}}+1, \mathscr{a}_{ij}^{\mathbb{t}}+2, \ldots, \mathscr{a}_{ij}^{\mathbb{t}}+\mathscr{t}_{ij}^{\mathbb{t}}\} \tag{2-36}$$

$$0 \le DEC_t \le (1-\varkappa_t)\mathcal{D}_{max}, \qquad \forall t \in \{\mathscr{a}_{ij}^{\mathbb{t}}+1, \mathscr{a}_{ij}^{\mathbb{t}}+2, \ldots, \mathscr{a}_{ij}^{\mathbb{t}}+\mathscr{t}_{ij}^{\mathbb{t}}\} \tag{2-37}$$

and (2-30), (2-32) and (2-33).

While one can use this model in the development of the robust extension of the SPRP, as we will show later in the computational results section of the paper, very realistic cycles could be generated from the same model only by simply using empirical 'mean' acceleration and deceleration rates ($\mathcal{A}_{avg}$ and $\mathcal{D}_{avg}$) in the model instead of $\mathcal{A}_{max}$ and $\mathcal{D}_{max}$. In this paper, for these parameters we use the reported results by Bokare and Maurya (2017) from their study on the A/D behaviour of various vehicle types including trucks. Based on their results while $\mathcal{A}_{max} \cong 1$ m/s^2, and $\mathcal{D}_{max} \cong 0.88$ m/s^2, the mean acceleration rate of a truck is around 0.3 *m/s*^{2} and the mean deceleration rate is around 0.5 *m/s*^{2}. In Figure 2-4.a, as a real-world driving cycle, we are illustrating the EPA heavy duty urban driving schedule (available at: https://www.epa.gov/sites/production/files/2015-10/huddscol.txt) which covers a distance of around *8935 m* in *1060 s*, with an estimated fuel consumption of 2.55 *litres* for an empty light-duty truck. Using $DC_{ij}^{\mathfrak{t}}$ the worst-case driving cycle is generated and illustrated in Figure 2-4.b with a fuel consumption of 3.22 *litres*, and the cycle based on the mean A/Ds with a very close amount of fuel consumption to the original cycle, i.e. 2.60 *litres*, is shown in Figure 2-4.c.



(a) FC = 2.55 lit.

(b) FC = 3.22 lit.

(c) FC = 2.60 lit.

Figure 2-4 Example synthetic driving cycles constructed from the proposed MILP model

Using the proposed model, if historical microscopic data are unavailable at the planning stage, it is possible to generate reliable driving cycles without collecting field data, and estimate fuel consumption more accurately than using the average-speed CMEM.

To generate the required cycles for all network links at all time periods, the proposed model must be used iteratively. An intuitive iterative algorithm for this purpose is presented in Algorithm 2-3. To speed up this procedure, two things are done; firstly, we set the relative MIP gap tolerance of the CPLEX mixed integer programming setting to 0.01, and the global time limit to 3 seconds. That is, if an optimal solution or a solution with 1% MIP gap is observed in less than 3 seconds, it is accepted; otherwise, the solution that is returned by CPLEX after 3 seconds is accepted and used in the calculation of the road-link attributes (we never encountered a solution with over 5% optimality gap with this setting). While CPLEX usually requires a few seconds to close the MIP gap and return the optimal solution, in almost all our observations, a solution with 1% MIP gap is returned within fractions of a second. This solution is either the same as the optimal solution or very marginally different from that. Secondly, we store the obtained information iteratively in a hash table to use in later iterations. Indeed, the $DC_{ij}^{\mathbb{t}}$ MILP relies mainly on two pieces of information; i.e. d_{ij} and $\boldsymbol{t}_{ij}^{\mathbb{t}}$. If we use $\lceil d_{ij} \rceil$ instead of d_{ij}, then the combination $\lceil d_{ij} \rceil$ and $\boldsymbol{t}_{ij}^{\mathbb{t}}$ is repeatedly observed for many road-links at different time periods. Note that the effect of this rounding up of distances on the ultimate values for the UTM and RTM attributes of the road-link is very negligible. Hence, in each iteration of the algorithm once the cycle is returned by the $DC_{ij}^{\mathbb{t}}$ MILP and the UTM and RTM attributes of the road link are computed, they are stored in the hash table (*HT*) with their key $(\lceil d_{ij} \rceil, \boldsymbol{t}_{ij}^{\mathbb{t}})$. In following iterations before calling CPLEX to solve the $DC_{ij}^{\mathbb{t}}$ MILP, the key is checked with the hash table to see if UTM and RTM values could be directly obtained from the table. Observe that at the end of the algorithm execution, it only needs to return the hash table, from which all UTM and RTM attributes of all road-links at all time periods could be extracted.

Algorithm 2-3 Networkwide generation of driving cycles

```
**Input** $G$, $d_{ij}\ \forall i,j \in N$, $\boldsymbol{t}_{ij}^{\mathbb{t}}\ \forall i,j \in N, \mathbb{t} \in \mathcal{T}_{ij}$
$HT = \{\}$
**for** $i = 0$ **to** $n + \boldsymbol{m}$ **do**
    **for** $j = 0$ **to** $n + \boldsymbol{m}$ **do**
    **if** $i \neq j$ **then**
        **for** $\mathbb{t} = 1$ **to** $\boldsymbol{h}_{ij}$ **do**
            **if** $HT$ does not contain the key $(\lceil d_{ij} \rceil, \boldsymbol{t}_{ij}^{\mathbb{t}})$ **then**
                Solve $DC_{ij}^{\mathbb{t}}$ MILP
                Compute $Z_{ij}^{k\mathbb{t}}\ \forall k \in K$ and $\Gamma_{ij}^{\mathbb{t}}$ and add them to the $HT$ along with the
                key $(\lceil d_{ij} \rceil, \boldsymbol{t}_{ij}^{\mathbb{t}})$
            **end if**
        **end for**
    **end for**
**end for**
**return** $HT$
```

In the next section, we report the computational cost of generating the network-wide driving cycles in the case of each instance using this algorithm.

As a wrapper of all modules and processes introduced in the paper, in the flow chart in Figure 2-5, we present the flow of the operations that are carried out on the original roadway graph until the full set of the ND points to an instance of the SPRP are identified. In this flow chart, we are indicating the section of the chapter that is relevant to each of the four main modules introduced.

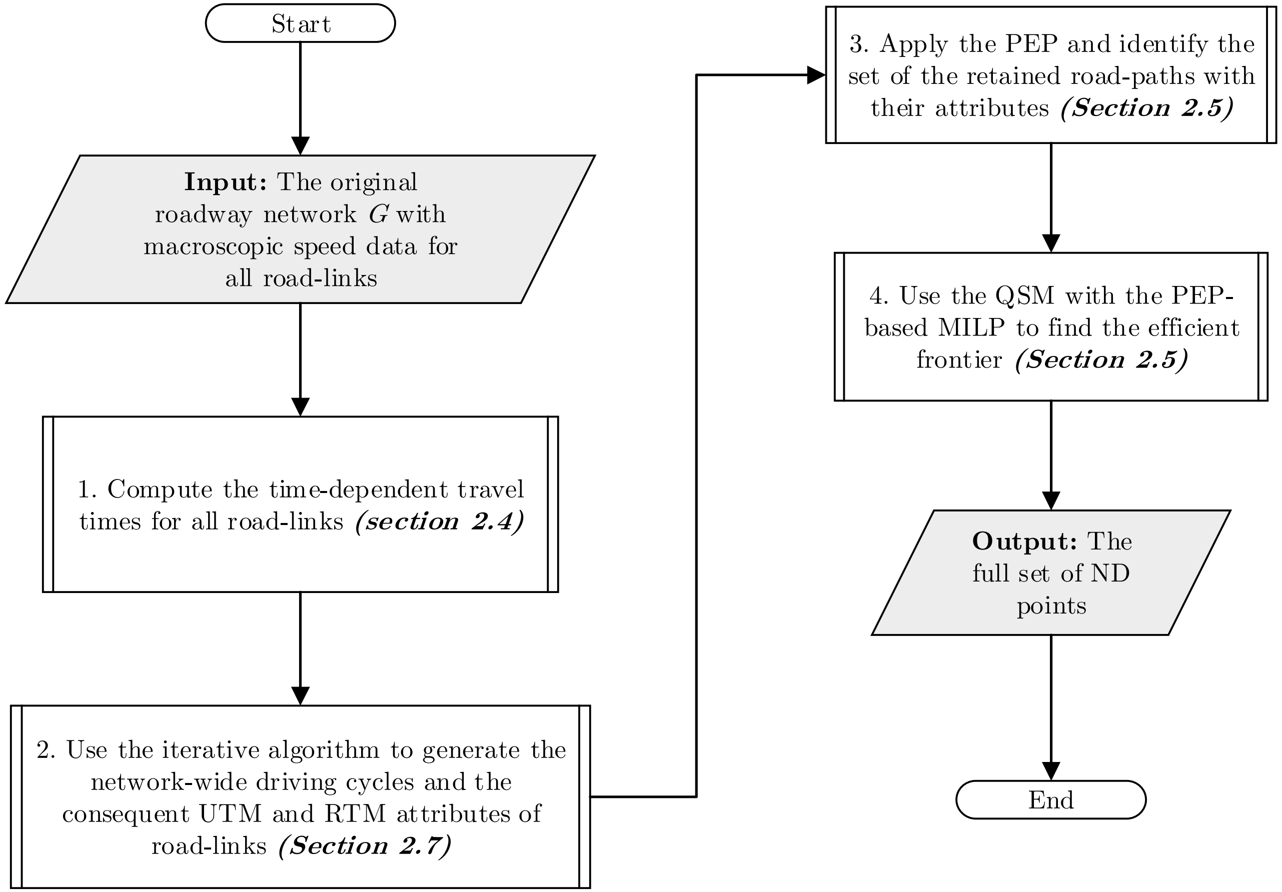

Start
Input: The original roadway network G with macroscopic speed data for all road-links
1. Compute the time-dependent travel times for all road-links ***(section 2.4)***
2. Use the iterative algorithm to generate the network-wide driving cycles and the consequent UTM and RTM attributes of road-links ***(Section 2.7)***
3. Apply the PEP and identify the set of the retained road-paths with their attributes ***(Section 2.5)***
4. Use the QSM with the PEP-based MILP to find the efficient frontier ***(Section 2.5)***
Output: The full set of ND points
End

Figure 2-5 The flow of modules introduced in the paper until finding the full set of the ND points

2.7 Computational results

A set of time-dependent roadway graphs with 100 nodes are created as the test bed of the proposed PEP-based MILP. For these instances, the raw graphs are generated using the procedure proposed by Letchford et al. (2013) that leads to graphs which resemble real-life road networks. To generate time-dependent travel times for the arcs, a planning horizon of 480 minutes is assumed, and traffic condition is supposed to follow a non-congested/congested/non-congested/congested pattern. The two congested periods represent the morning and evening rush hours with speed values generated randomly within the range 15-40 *km/h*. Non-congested speeds, on the other hand, are determined randomly within the range 40-70 *km/h*. Within this pattern, random speed observations are generated independently for each arc in 15-minute increments (i.e. 32 speed observations per road-link) such that speed levels change from one period to the next smoothly. Following this, the time-dependent travel times of arcs are computed per minute using the proposed closed form formula (2-3) and rounded up to the nearest integer. For all arcs, driving cycles are generated using $DC_{ij}^{\mathbb{t}}$ MILP for each time period, and the resulting UTM and RTM attributes of the arcs are computed and stored.

Across the generated road networks, we have randomly selected 10 customers to be served by a central depot. For multi-objective experiments, to be able to generate the full set of the ND points on the efficient frontier within a reasonable time, we have considered 5 customers. It is worth mentioning that while the dimension of the proposed instances compares well with other existing papers with an exact approach given the extra complications from multi-trips and flexible paths, it is not an intention of this paper to solve large size test instances, and it has been the task of our follow up paper on the development of multi-objective optimisation heuristics for the SPRP. Instead, here we are more interested in observing the performance of the PEP, the contributions of the multi-trips, the trade-offs on the Pareto front of the SPRP and presenting a benchmark for examining the performance of future heuristics for the problem.

In all test instances, service times and the reloading time at the depot for vehicles executing an extra round of trip are assumed 20 minutes. Feasible time-windows and demands are induced for the customers using a procedure based on a nearest neighbour algorithm where a heavy-duty truck is dispatched to visit the nearest customer in each iteration of the algorithm, until capacity or time constraints are violated. Customers' demands are drawn randomly from the discrete uniform distribution on the interval [1000kg, 15000kg], and relatively wide time-windows covering up to 40% of the planning horizon are generated around the arrival time of the dispatched trucks. For all instances, the fleet is supposed to be composed of two light-duty, two medium-duty, and two heavy-duty trucks, and all trucks are allowed to execute a maximum of two rounds of trips during the planning horizon. All the test instances and reported solutions in this section are available at: https://dx.doi.org/10.17635/lancaster/researchdata/266.

All the experiments were performed on a computer with Intel Core™ i5 3.20 GHz processor with 8 GB RAM. The branch-and-bound solver of CPLEX™ 12.6.3 was used as the exact solver, and except for the PEP and the travel time calculation algorithms that were coded in MATLAB™, all other algorithms were modelled as OPL scripts in pre-processing, post-processing and model flow control on top of the OPL models of the core MILPs. No global time limit was used to allow the solver to generate the full set of the ND points for benchmarking purposes.

2.7.1 Performance of the PEP

To demonstrate the efficiency of the proposed PEP and the MILP based on it in addressing the SPRP, all test instances with 10 customers are solved to optimality for the fuel consumption minimisation objective using the PEP-based MILP, and solutions are compared with the solutions from a TDSP-based MILP, where a multi-graph based on the full set of the TDSPs between required nodes is used. The results of this comparison are presented in Table 2-4. In this table, the total number of arcs in the original networks, and in the reduced networks after applying the PEP and the TDSP are shown, along with the litres of fuel consumed (FC) by the optimal solutions returned by each of the formulations, and the runtime (in minutes). One column is also devoted to reporting the percentage optimality gap of the TDSP-based formulation from the optimal solution returned by the PEP-based formulation. Table 2-4 shows that for all the instances considered, the solution based on the TDSP is suboptimal. The table also indicates that TDSPs on average can only represent less than 44% of the eligible road-paths in the graph. Observe that the runtime columns of the table suggest that despite the difficulty of the problem, the proposed MILP formulation can find the optimal solutions in a reasonable runtime with an average of less than 10 minutes.

Table 2-4

Optimal fuel consumption yielded by the PEP-based and the TDSP-based formulations

Instance	Network Arcs	PEP-based MILP			TDSP-based MILP			
#		Arcs	FC	Runtime (min)	Arcs	FC	Runtime (min)	Gap
1	344	149	54.24	11.03	109	55.56	29.72	2.43%
2	370	270	79.32	1.26	82	86.43	0.23	8.96%
3	358	195	55.36	18.03	127	57.27	4.68	3.46%
4	348	187	77.06	1.94	96	84.64	4.06	9.83%
5	362	259	65.85	5.26	112	69.97	3.28	6.26%
6	366	182	68.30	1.02	92	74.21	0.26	8.65%
7	354	247	59.09	32.20	152	63.38	80.98	7.26%
8	358	218	71.56	0.76	108	79.02	0.92	10.42%
9	374	190	82.15	3.44	131	84.80	3.45	3.22%
10	350	208	82.95	0.82	117	90.14	0.79	8.67%
11	358	172	56.49	4.65	119	58.99	10.72	4.42%
12	360	157	81.25	0.90	90	87.60	0.60	7.82%
13	362	180	67.39	1.39	96	68.86	0.50	2.18%
14	358	214	79.84	1.62	121	85.51	1.35	7.11%
15	360	247	49.12	15.86	145	51.15	12.62	4.13%
16	348	149	60.28	0.28	99	65.75	0.86	9.08%
17	412	210	65.48	3.76	102	69.25	1.47	5.76%
18	348	158	78.31	1.71	89	86.21	1.36	10.09%
19	426	238	51.58	5.21	102	58.01	6.77	12.48%
20	352	167	71.20	0.23	85	77.27	0.22	8.53%
21	346	248	77.16	3.96	99	82.73	0.54	7.22%
22	370	211	76.04	0.59	106	81.58	0.71	7.27%
23	350	200	77.47	1.93	103	85.20	0.49	9.96%
24	368	221	77.80	11.52	103	83.87	1.43	7.80%
25	362	182	78.69	2.03	83	87.70	1.89	11.45%
Avg.	362.56	202.36	69.76	5.26	106.72	75.00	3.33	7.52%

For a more in-depth analysis of the performance of the PEP and the three other approximate algorithms, we are also using a publicly available real-world urban road network with time-dependent speed observations. This is based on the Chicago's arterial (non-freeway) streets (https://data.cityofchicago.org) (Figure 2-6), which has 1485 nodes and 1257 arcs, and congestion estimates are produced every fifteen minutes in real-time. However, the database has some incompleteness issues (i.e. not all nodes are accessible from one another) due to the omission of roads and streets in the lower hierarchy of roads, such as collectors, and local roads. Also, real-time speed observations only become available for a subset of all segments, and speed data are missing for some arcs at some time instants. Therefore, the graph and the traffic updates provided by Dokka and Goerigk (2017) after treating this database for the mentioned issues are utilised. Dokka and Goerigk (2017) record traffic updates in a 15-minute interval over a time horizon of 24 hours spanning Monday March 27th, 2017 morning to Tuesday March 28th, 2017 morning, leading to a total of 98 data observations, and their final graph contains 538 nodes and 1308 arcs.

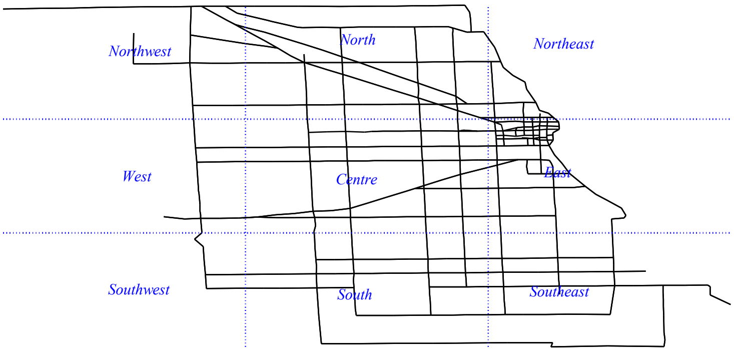


Figure 2-6 Chicago's arterial streets

Four snapshots of Chicago's arterial streets at different time instants of a typical day are displayed in Figure 2-7. In this figure, green arcs have an average speed above 30 km/h, yellow arcs represent an average speed between 15 and 30 km/h, and red arcs show an average speed below 15 km/h.

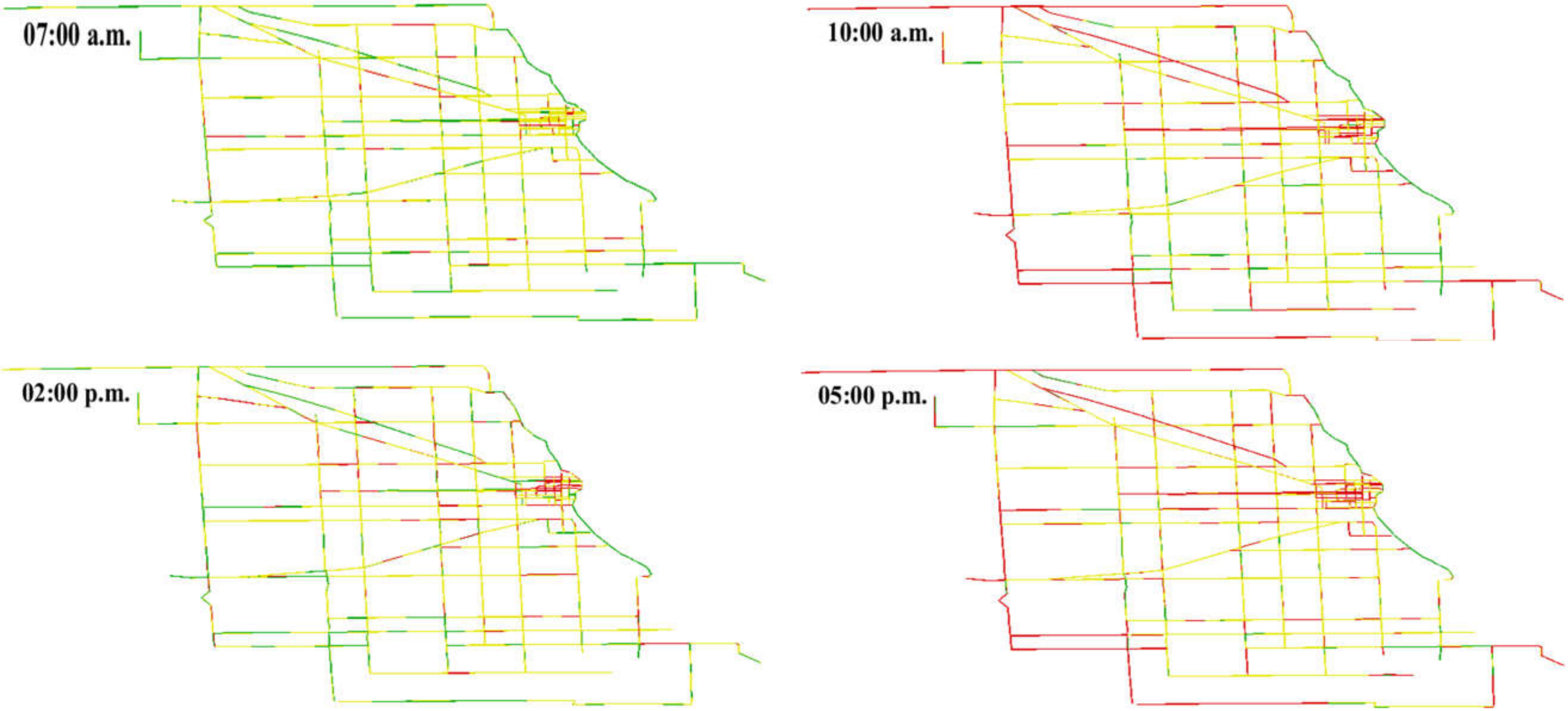


Figure 2-7 Time dependent congestion in Chicago's arterial streets at four different time instants in a typical day

A set of 11 source-sink pairs were selected on this road network, and a planning horizon of 12 hours from 07:00 to 19:00 was considered for experiments. These pairs are identified based on the location of the source and the sink, which are placed somewhere in the *Centre (C)*, *North (N)*, *East (E)*, *West (W)*, *south(S)*, *Northeast (NE)*, *Northwest (NW)*, *Southeast (SE)*, or *Southwest (SW)* of the graph (Figure 2-6). Therefore, we denote by the pair *(N, S)*, for instance, a pair of source-sink, where the source is in the North of the graph and the sink is in the South. To validate the performance of the PEP, an exhaustive approach for the exact identification of the full set of the eligible paths was used. This approach is based on the discretisation of the load range [0,26000] in 20 kg increments (1301 increments), and computation of the emissions minimising paths for all the resulting increments and feasible vehicle types, at all 720 possible departure times (every minute in the planning horizon). With this approach (called the Exact approach hereafter) we can identify the full set of all eligible paths and consider this set as the benchmark for the evaluation of the alternative approaches. In Table 2-5 the performance of each of the PEP and TDSP algorithms against the exact set of the eligible paths is reported. In this table the column with the heading “No. of Paths” indicates the total number of paths between the given source-sink pair, identified by each of the three approaches. The next three columns report the average number of arcs in all the paths identified, the average distance of these paths, and the average travel times of them based on each of the PEP and the TDSP algorithms. Finally, the last column reports the runtime of each algorithm for returning the given set of paths. For the PEP and TDSP the reported runtime is based on the average of 10 runs.

Table 2-5

The performance of the PEP and the TDSP algorithms against the Exact set of the eligible road-paths

Pair	Regions	No. of Paths			Avg. No. of arcs		Avg. distance		Avg. travel time		Runtime (seconds)		
		Exact	PEP	TDSP	PEP	TDSP	PEP	TDSP	PEP	TDSP	Exact	PEP	TDSP
(499,481)	(*C*, *C*)	5	5	1(20%)	13.00	13.00	4.27	4.28	17.28	15.00	1400.97	4.21	0.87
(7,314)	(*E*, *E*)	2	2	2(100%)	11.00	11.00	7.72	7.72	13.59	13.59	1110.91	3.96	0.90
(106,325)	(*N*, *N*)	9	9	4(45%)	15.56	16.25	20.59	21.57	44.85	46.17	955.36	4.88	0.58
(426,117)	(*S*, *S*)	6	6	2(34%)	13.67	13.00	12.93	12.90	28.36	27.11	984.09	3.21	0.46
(3,72)	(*W*, *W*)	1	1	1(100%)	8.00	8.00	3.77	3.77	13.13	13.13	1330.73	3.71	0.80
(20,175)	(*C*, *E*)	16	16	6(40%)	23.50	20.00	20.01	21.34	47.50	41.08	1505.43	3.60	0.89
(19,325)	(*NW*, *N*)	10	10	4(40%)	16.10	16.25	21.23	21.98	45.97	45.94	963.59	2.49	0.90
(49,111)	(*S*, *SE*)	9	9	5(56%)	27.56	26.00	38.08	38.06	68.93	67.12	2303.67	3.65	1.17
(82,55)	(*N*, *S*)	18	18	7(39%)	39.17	39.86	23.45	23.59	62.99	61.54	1407.33	3.46	1.10
(3,15)	(*W*, *E*)	28	28	14(50%)	29.32	29.07	36.34	36.34	73.88	73.45	1805.42	4.13	0.77
(47,430)	(*SW*, *NE*)	46	46	24(52%)	39.48	39.83	29.59	29.61	71.27	70.31	1760.21	4.34	0.64

Table 2-6

The performance of alternative approximate path elimination algorithms

Pair	No. of Paths			Runtime (seconds)		
	Alt_1	Alt_2	Alt_3	Alt_1	Alt_2	Alt_3
(499,481)	5(100%)	5(100%)	5(100%)	4.20	2.13	1.26
(7,314)	2(100%)	2(100%)	2(100%)	3.96	2.79	2.16
(106,325)	9(100%)	9(100%)	7(77.78%)	4.76	2.04	1.49
(426,117)	6(100%)	6(100%)	6(100%)	3.17	2.11	1.35
(3,72)	1(100%)	1(100%)	1(100%)	3.71	2.74	2.13
(20,175)	15(93.75%)	15(93.75%)	15(93.75%)	3.50	1.77	2.34
(19,325)	10(100%)	8(80%)	7(70%)	2.42	1.90	1.40
(49,111)	9(100%)	9(100%)	9(100%)	3.60	3.05	1.60
(82,55)	17(94.44%)	15(88.24%)	11(64.71%)	3.34	2.18	1.73
(3,15)	28(100%)	26(92.86%)	23(82.14%)	3.98	2.20	1.84
(47,430)	46(100%)	44(95.65%)	40(86.96%)	4.01	2.40	1.55

According to Table 2-5, while the PEP is able to exactly identify all of the eligible paths, the TDSP can identify only less than 50% of the eligible paths in most of the cases. Note that this also implies that adding the shortest distance path to the set of the TDSPs (if it is not already there) as proposed by Huang et al. (2017) cannot help much. This table also suggests that the number of the paths to retain is a pretty much relative value depending on several factors, and it is not possible to issue any prescription on a safe number of paths to retain as in the k-shortest path network reduction technique of Androutsopoulos and Zografos (2017). While their results may suggest that keeping 5 paths can reduce the number of excluded paths, based on Table 2-5, this can be a too big number (e.g. for source-sink (3, 27)), or a too small one (e.g. for source-sink (47, 430)).

A rather intuitive implication of the results in Table 2-5, however, is that as the distance and the travel times between a given pair of source-sink increase, so does the number of arcs on the paths connecting them and the total number of eligible paths. For representation, some of the eligible paths between some of the pairs in Table 2-5 are illustrated in Figure 2-8.

The percentage of time that the fuel consumption minimising path for a full truck of a given type, i.e. p_1, has been not the same as the fuel consumption minimisation path for the same truck type with no load, i.e. p_2, (see Proposition 1) in our experiments is also illustrated in the stacked bar chart in Figure 2-9 for all the 11 source-sink pairs in case of each vehicle type. This is the ratio of time instants out of the total 720 minutes in the time-horizon when the difference is observed. It is clear that in the case of the light-duty truck, due to its very small capacity, much less difference is observed compared with the medium and the heavy-duty trucks. Furthermore, again as the source and the sink become more distant on the graph, this ratio tends to increase. Therefore, while the approach proposed by Ehmke et al. (2016) might be helpful in case of the light-duty trucks on small graphs, the need to compute the shortest path on-the-fly for time instants when a difference between p_1 and p_2 is observed, make it prohibitive on larger graphs and larger vehicle types.

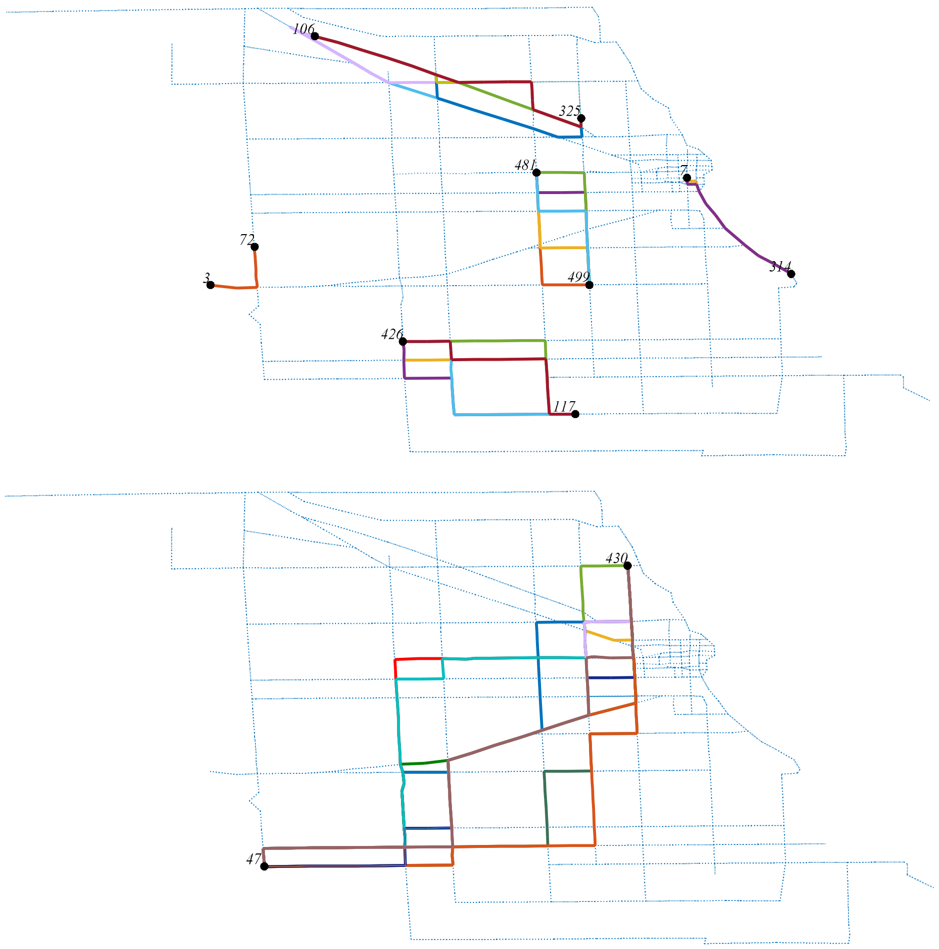


Figure 2-8 Illustration of some of the eligible paths between some of the considered source-sink pairs on Chicago road graph

The performance of the three alternative approximate path elimination algorithms discussed in section 2.4.2, i.e. Alt_1, Alt_2 *and* Alt_3 is reported in Table 2-6. The table shows that all these algorithms that are based on the premature termination of the PEP, and rely on the tight bounds set by p_1 and p_2 according to Theorem 2 have a good performance, with Alt_1 being the best alternative and

Alt₂ a bit better performing than *Alt₃*. These algorithms are as simple as the TDSP to implement while they promise a much better performance.

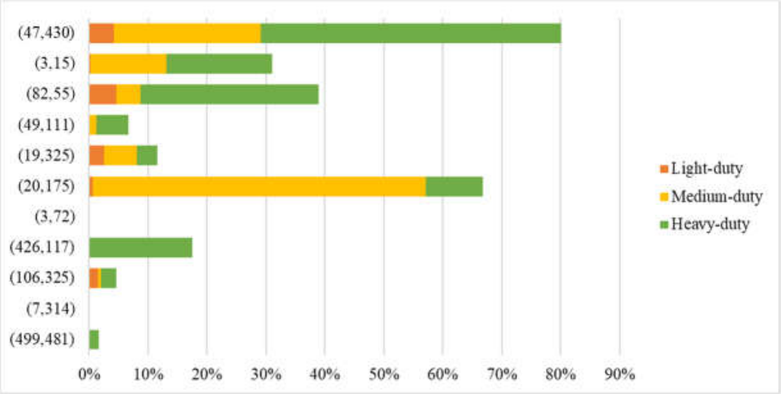


Figure 2-9 The percentage of time that $p_1 \neq p_2$ for each truck type

2.7.2 Fleet size and mix, and the effect of using multiple-trips

The fleet size and mix of the optimal solutions to instances with 10 customers in case of the fuel consumption minimisation is analysed and presented in Table 2-7. In this table, the column 'C' indicates the cost of hiring the trucks in the solution, and the column 'R' denotes the number of routes in the solution. Number of each vehicle type light-duty (L), medium-duty (M) and heavy-duty (H) employed by the solution is presented under the heading 'NU'. Average percentage of capacity of each truck type used when departing the depot is shown by 'CU', and the number of each truck type employed for multiple trips is given under the heading 'MT'. Since only medium-duty trucks are selected by all solutions for multi-trips, the average percentage of their capacity used for the second round of delivery is shown in column 'CUM'. Finally, the last two columns of the table denote the percentage savings in fuel consumption and cost due to the use of multi-trips. To measure the savings, we have prohibited multi-trips from all instances and then re-optimised the problems for fuel consumption minimisation and calculated the deviation. Note that in case of instance #9 the problem is infeasible if multi-trips are not allowed.

Table 2-7

Fleet size and mix, average capacity use and the use of the multiple-trips in the optimal fuel consuming solutions

Inst.	C	R	NU			CU			MT			CUM	MT savings	
			L	M	H	L	M	H	L	M	H	M	Fuel	cost
1	162	5	1	2	0	79%	91%	-	0	2	0	68%	2%	84%
2	214	4	0	2	1	-	74%	94%	0	1	0	82%	2%	16%
3	120	4	0	2	0	-	74%	-	0	2	0	77%	9%	78%
4	308	4	0	2	2	-	60%	82%	0	0	0	-	0%	0%
5	204	5	2	2	0	98%	63%	-	0	1	0	97%	4%	46%
6	214	4	0	2	1	-	76%	89%	0	1	0	64%	3%	44%
7	120	4	0	2	0	-	77%	-	0	2	0	79%	2%	148%
8	214	4	0	2	1	-	53%	75%	0	1	0	80%	0%	39%
9	392	8	2	2	2	92%	82%	73%	0	2	0	77%	-	-
10	214	5	0	2	1	-	54%	86%	0	2	0	85%	3%	64%
11	204	5	2	2	0	73%	81%	-	0	1	0	94%	3%	0%
12	214	5	0	2	1	-	68%	94%	0	2	0	75%	2%	44%
13	298	6	2	2	1	68%	85%	87%	0	1	0	67%	2%	32%
14	298	6	2	2	1	84%	51%	51%	0	1	0	80%	0%	0%
15	162	4	1	2	0	73%	67%	-	0	1	0	80%	10%	58%
16	214	5	0	2	1	-	83%	58%	0	2	0	86%	5%	44%
17	214	4	0	2	1	-	95%	97%	0	1	0	74%	4%	44%
18	214	5	0	2	1	-	83%	69%	0	2	0	75%	2%	44%
19	204	5	2	2	0	79%	59%	-	0	1	0	78%	0%	46%
20	214	4	0	2	1	-	72%	68%	0	1	0	40%	2%	0%
21	308	5	0	2	2	-	85%	70%	0	1	0	56%	4%	0%
22	214	4	0	2	1	-	89%	84%	0	1	0	73%	4%	16%
23	214	4	0	2	1	-	85%	94%	0	1	0	92%	5%	64%
24	256	6	1	2	1	72%	79%	30%	0	2	0	70%	4%	37%
25	214	5	0	2	1	-	66%	84%	0	2	0	77%	3%	44%
Avg.	224.16	4.80	0.60	2.00	0.84	80%	74%	77%	0.00	1.36	0.00	76%	3%	31%

It is clear from the table that the medium-duty truck is the most preferred resource in the fleet, as it establishes an adequate balance between the energy-efficient but capacity-inefficient light-duty truck, and the energy-inefficient but capacity-efficient heavy-duty truck. Moreover, capacity usage in all cases for all vehicle types is very well distributed, for which some contribution could be attributed to multiple-trips possibility. It is also interesting to see that when a vehicle is assigned to an extra round of trip it is not under-utilised in terms of its capacity.

2.7.3 The Pareto front of the SPRP

All the test instances with 5 customers were solved to multi-objective optimality for the generation of the full set of the ND points on the true Pareto fronts using the approach discussed in section 5 of the paper, i.e. embedding the PEP-based MILP within the framework of the QSM (QSM+PEP-based MILP). Again, an alternative approach based on the integration of the QSM with the TDSP-based MILP (QSM+TDSP-based MILP) is considered to further see the effect of suboptimal network reduction techniques on the generation of the true Pareto fronts. The QSM+TDSP-based MILP uses the TDSP-based MILP described earlier as the core optimisation problem. That is, a multi-graph based on the full set of the TDSPs between required nodes is considered, and the MILP described in (2-16)-(2-28) is solved on this multi-graph rather than on the original PEP-based multi-graph. Like the case of the QSM+PEP-based MILP, in each iteration of the QSM+TDSP-based MILP, for returning a ND vector, the two (M)IPs discussed in section 5 must be solved. For all test instances we had to calibrate only ϵ_2 as the first objective is integer, and hence considered $\epsilon_1 = 1$, and $\epsilon_2 = 0.01$ (ϵ_1 and ϵ_2 are QSM parameters). Also, as earlier mentioned no time limit was applied on the solver to allow the full set of the ND points to be found.

In Table 2-8, the total number of ND points and the objective-wise values of the extreme points on the Pareto fronts found by each of the two approaches are shown. In this table, the columns *Obj*$_1$, *Obj*$_2$, and *Obj*$_3$ represent the global minimum value found for the first objective (vehicle cost in £), the second objective (fuel consumption in *lit.*) and the third objective (travel time in *mins.*), respectively. The important column here is the column with the heading '$\text{I} \prec \text{II}$' which indicates the number of ND points found by the *QSM+TDSP-based MILP* which are strictly dominated by the ND points on the *QSM+PEP-based SPRP* Pareto front.

It is observed that in several of the cases the number of the ND points found by the TDSP-based formulation is much less than what is found by the PEP-based one, and more importantly, many of these are strictly dominated. For example, there are 50 ND points on the true Pareto front of instance #3 found by the PEP-based formulation, and there are only 12 ND points on the Pareto front of the TDSP-based model; however, 10 of these are strictly dominated by the solutions on the true front, meaning that the TDSP-based approach has been in effect able to find only 2 ND points out of the 50 solutions. On average, the number of the ND points found by the TDSP-based formulation is 70% of what is found by the PEP-based formulation, but around 75% of these are strictly dominated by the true front. Comparison of the extreme points, on the other hand, reveals that as expected the solutions with minimal vehicle cost and travel time could be found by the TDSP-based formulation; however, complying with the results in Table 2-4, in all cases a suboptimal solution for fuel consumption minimisation is yielded by the TDSP-based formulation..

Table 2-8

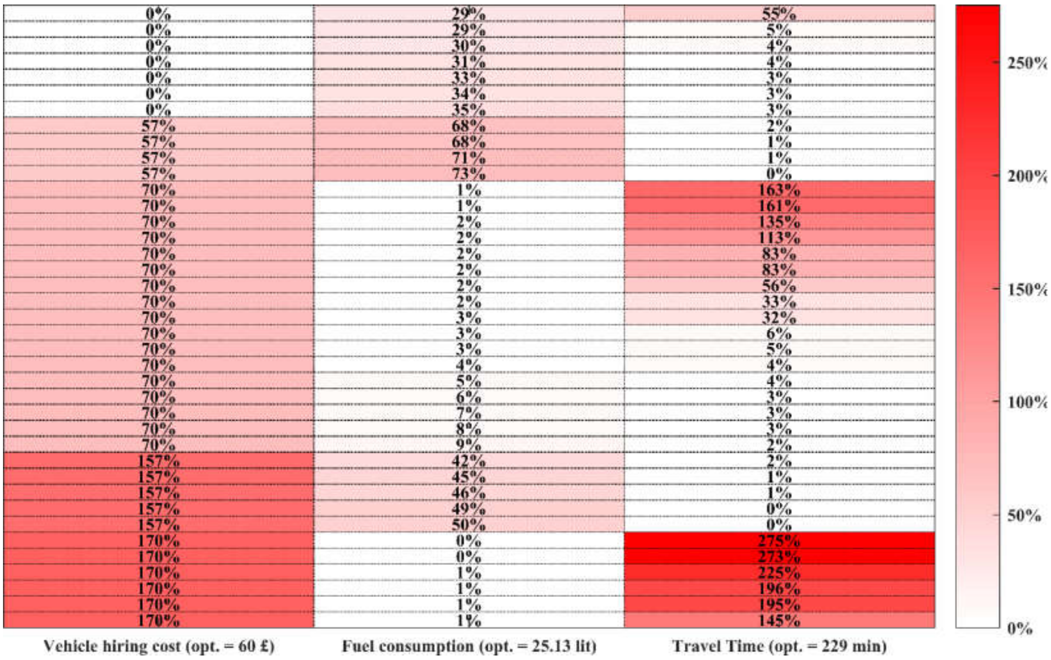
Extreme points on the Pareto fronts generated by each approach

Instance #	I: *QSM+PEP-based MILP*					II: *QSM+TDSP-based MILP*					
	# of ND points	Extreme points			Runtime (min)	# of ND points	Extreme points			I $\prec$ II	Runtime (min)
		Obj_1	Obj_2	Obj_3			Obj_1	Obj_2	Obj_3		
1	40	94	43.66	228	127.24	28	94	49.54	228	22	21.78
2	43	60	28.16	245	808.28	37	60	29.11	245	33	46.63
3	50	94	38.86	222	150.20	12	94	41.41	222	10	10.13
4	39	60	25.13	229	73.45	33	60	25.52	229	24	43.90
5	31	94	36.96	267	166.40	12	94	38.46	267	10	12.73
6	65	94	36.91	227	1095.91	41	94	38.35	227	34	175.08
7	51	94	41.91	256	662.77	46	94	43.57	256	31	171.47
8	79	94	41.52	270	222.03	72	94	42.11	270	70	123.31
9	77	136	50.08	250	218.51	57	136	50.74	250	35	103.90
10	101	154	67.75	323	616.62	78	154	68.46	323	43	299.35
11	39	94	44.31	210	51.84	25	94	46.30	210	12	8.12
12	41	94	25.97	192	657.65	28	94	26.89	192	19	116.95
13	27	154	56.75	239	82.50	22	154	58.67	239	13	108.78
14	82	136	41.90	234	133.68	55	136	43.31	234	43	67.65
15	30	94	31.67	207	76.88	23	94	32.29	207	16	63.87
16	100	94	41.04	247	982.92	88	94	41.18	247	82	775.78
17	34	94	24.52	180	159.99	39	94	26.62	180	35	63.89
18	53	60	33.98	231	177.97	13	60	34.81	231	6	8.22
19	61	94	38.21	212	48.66	33	94	39.14	212	21	7.75
20	41	94	25.97	192	566.37	28	94	26.89	192	19	111.30
21	27	154	56.75	239	83.78	22	154	58.67	239	13	106.04
22	82	136	41.90	234	108.65	55	136	43.31	234	43	70.34
23	34	94	24.52	180	133.43	39	94	26.62	180	35	59.76
24	53	60	33.98	231	129.54	13	60	34.81	231	6	7.87
25	61	94	38.21	212	41.26	33	94	39.14	212	21	7.29
Average	53.64	100.8	38.82	230.28	303.06	37.28	100.8	40.24	230.28	27.84	103.68

It is worth mentioning that despite the size of the POSs (i.e. around 54 ND points on average) the full sets of ND points are generated within a very reasonable runtime of around 304 minutes on average, suggesting the successful integration of the QSM with the proposed MILP formulation.

Unlike bi-objective optimisation, visualisation of the Pareto front in the case of the multi-objective optimisation with more than two objectives is not easy. However, to provide the decision maker with a useful visual presentation of the trade-offs among the ND points on the Pareto front, and aid her/him to select a solution that provides a suitable compromise among the objective values, we propose the use of enhanced heat maps similar to the one shown in Figure 2-10. In this figure, the Pareto front of instance #4 is selected and the percentage deviations from the absolute minimum in case of each objective function for all the 39 solutions on the considered front is shown. While it can be observed that there exists a significant trade-off among the three objectives of the SPRP and the minimisation of one objective can significantly deteriorate the value of the other two, with the help of the colour gradient, this figure makes it possible to visually locate the more balanced solutions.

It is also clear from Figure 2-10 that in the case of the considered instance the range of deviations in the travel time objective is much larger than the other two objectives, and it is particularly maximised when the solution tends to minimise fuel consumption. To investigate this further, we analyse that when one of the objectives is minimised, how much sacrifice is made in case of the other two objectives. As an average over all the instances with 5 customers, Figure 2-11 illustrates this trade-off. Similar to Figure 2-10, Figure 2-11 shows clearly that less sacrifice must be made in the other two objectives, when one minimises travel time, whereas the minimisation of fuel consumption can lead to a significant increase in the other two objectives.



Vehicle hiring cost (opt. = 60 £)	Fuel consumption (opt. = 25.13 lit)	Travel Time (opt. = 229 min)
0%	29%	55%
0%	29%	5%
0%	30%	4%
0%	31%	4%
0%	33%	3%
0%	34%	3%
0%	35%	3%
57%	68%	2%
57%	68%	1%
57%	71%	1%
57%	73%	0%
70%	1%	163%
70%	1%	161%
70%	2%	135%
70%	2%	113%
70%	2%	83%
70%	2%	83%
70%	2%	56%
70%	2%	33%
70%	3%	32%
70%	3%	6%
70%	3%	5%
70%	4%	4%
70%	5%	4%
70%	6%	3%
70%	7%	3%
70%	8%	3%
70%	9%	2%
157%	42%	2%
157%	45%	1%
157%	46%	1%
157%	49%	0%
157%	50%	0%
170%	0%	275%
170%	0%	273%
170%	1%	225%
170%	1%	196%
170%	1%	195%
170%	1%	145%

Figure 2-10 Heat map illustrating the ND points to a given SPRP instance

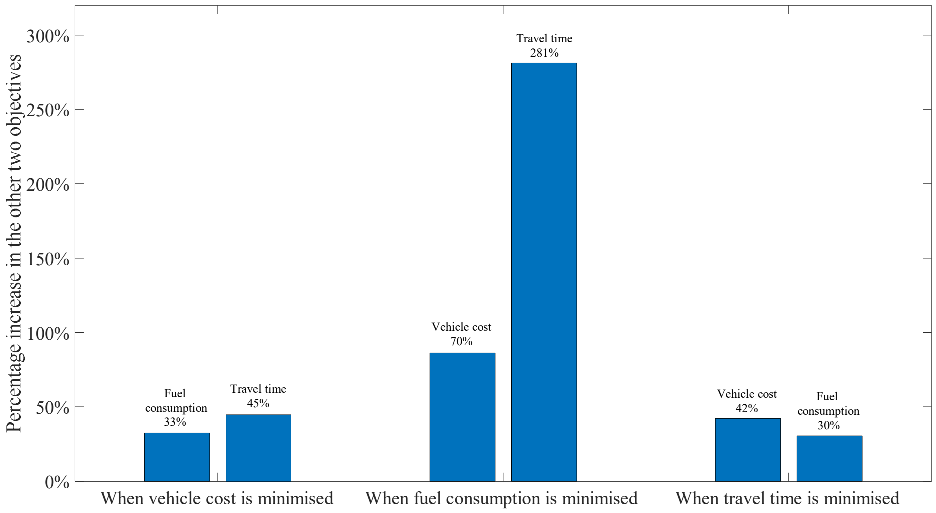
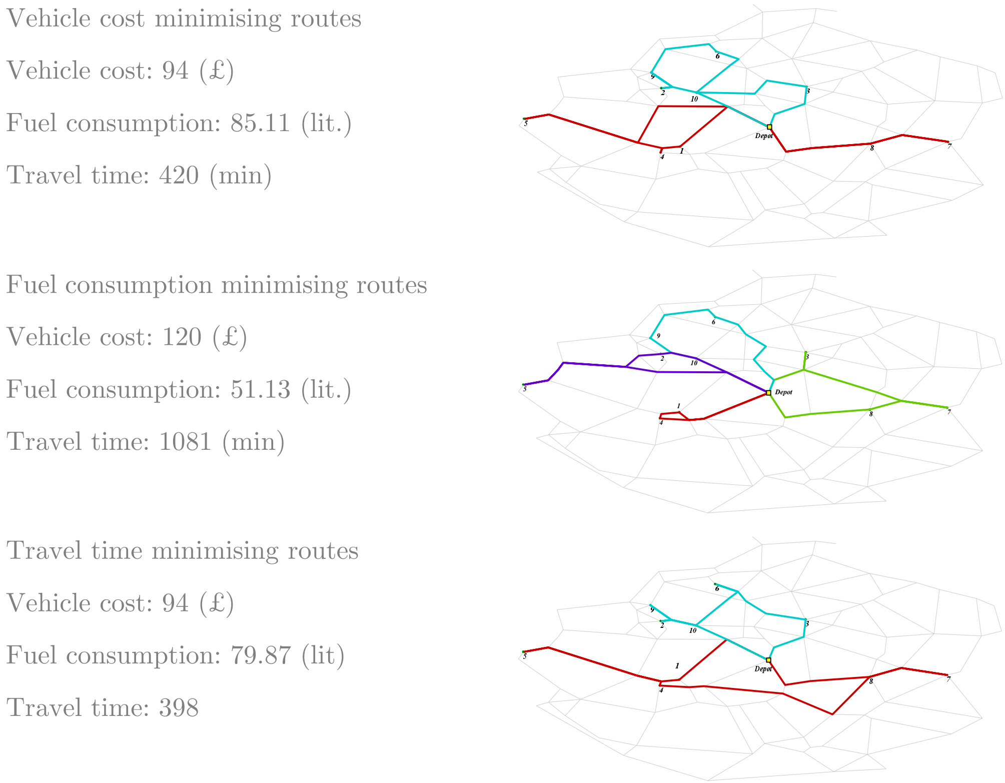


Figure 2-11 Average trade-off among the objective functions in case of the 5-customer instances

Finally, in Figure 2-12 for one instance we illustrate the routing patterns of the solution with minimum vehicle cost, the solution with minimum fuel consumption, and the solution with minimum travel time.



Vehicle cost minimising routes

Vehicle cost: 94 (£)

Fuel consumption: 85.11 (lit.)

Travel time: 420 (min)

Fuel consumption minimising routes

Vehicle cost: 120 (£)

Fuel consumption: 51.13 (lit.)

Travel time: 1081 (min)

Travel time minimising routes

Vehicle cost: 94 (£)

Fuel consumption: 79.87 (lit)

Travel time: 398

Figure 2-12 Routing patterns in case of the minimal objective value for each of the objectives

In this figure, in case of the minimum fuel consumption routes, two medium duty trucks are used for four routes (each making an extra round of trip), whereas in the other two cases all customers are served by one heavy–duty truck that also makes an extra round of delivery.

2.7.4 Reliability of the constructed driving cycles

In order to demonstrate the reliability of the proposed model for the generation of synthetic driving cycles, an extensive library of real-world driving cycles, consisting of over 19,000 different on-road driving cycles collected by Kancharla and Ramadurai(2018), are used as the benchmark set. For each cycle, the travel time and the distance of the cycle is fed into the proposed model, and the model is used to generate a synthetic cycle. Following the cycle generation, an empty light-duty truck (which is similar to the probe truck used in data collection in their study) is assumed to traverse both the real and the synthetic cycles, and the incurred fuel consumption based on the instantaneous CMEM in each case is computed. The percentage deviation of the incurred fuel consumption over the synthetic cycle from the fuel consumed over the real driving cycle (i.e. $[(Fuel_{synthetic} - Fuel_{real})/Fuel_{real}] \times 100$) is then calculated. A descriptive statistics summary of the percentage deviations in all the 19,362 cases is presented in Table 2-9, and a histogram of these deviations is illustrated in Figure 2-13.

Table 2-9
Descriptive statistics summary table of percentage deviation of the generated cycles from on road-cycles

Count	19362	Sample Variance	0.000127964
Mean	0.018511164	Kurtosis	1.31344784
Standard Error	8.12959E-05	Skewness	1.025999919
Median	0.016021384	Range	0.0983492
Mode	0.054775047	Minimum	-0.020289727
Standard Deviation	0.011312111	Maximum	0.078059472

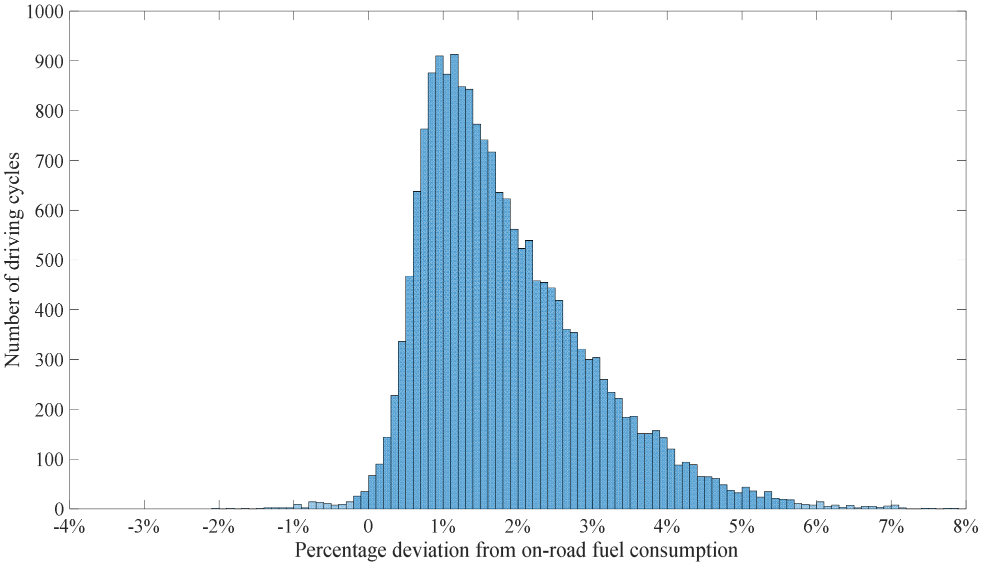


Figure 2-13 Histogram of percentage deviations of fuel consumption on synthetic cycles from on-road data

Based on Table 2-9, the proposed MILP can generate driving cycles, which have fuel consumption characteristics very close to real-life driving cycles, and on average lead to a fuel consumption estimation inaccuracy of less than 2%. Based on the histogram in Figure 2-13, for over 75% of the cycles considered, deviation lies between 0 and 2.5%.

Finally, it is useful to see the value of gaining this information and using truck A/D rates in fuel consumption estimation when routing on congested road networks. To this aim, we have solved all the instances with 10 customers to fuel consumption optimality when A/D rates are ignored and assumed zero all over the network, throughout the planning horizon. The performance of these solutions is then evaluated under the 'real' speed-time profiles; i.e. when the truck in practice accelerates and decelerates based on the generated driving cycles. With this, we are able to see to some extent the sub-optimality of these solutions and the estimation inaccuracy they contain. The results of these experiments are presented in Table 2-10.

Table 2-10

The effect of estimating fuel consumption inaccurately due to ignoring truck A/D rates

Instance #	Arcs	Percentage optimality gap	Estimation inaccuracy	DCs generation runtime (seconds)
1	129	11.67%	75.51%	294.90
2	157	5.73%	76.89%	512.66
3	160	2.80%	61.08%	320.29
4	169	1.02%	68.03%	484.26
5	185	2.06%	56.64%	351.54
6	151	0.37%	67.26%	483.57
7	191	2.13%	57.33%	317.61
8	168	3.68%	65.44%	532.96
9	120	9.59%	81.36%	1006.06
10	172	2.97%	66.49%	469.92
11	156	4.41%	68.61%	325.69
12	130	4.23%	71.33%	514.18
13	138	2.49%	52.77%	527.12
14	190	4.74%	73.33%	459.93
15	188	0.71%	57.80%	319.48
16	142	3.04%	58.43%	463.71
17	155	0.84%	58.82%	617.72
18	147	2.17%	68.74%	504.84
19	188	3.53%	45.77%	673.53
20	172	1.13%	58.52%	496.20
21	188	2.90%	75.05%	480.11
22	174	7.54%	76.86%	522.29
23	150	2.96%	65.74%	450.84
24	184	2.20%	55.03%	492.37
25	145	8.57%	79.10%	531.54

It must be noted that ignoring truck A/D rates results in a completely different reduced network when the PEP is applied. As it is presented in Table 2-10, when A/D rates are ignored the PEP can only represent on average around 80% of all eligible road-paths that are identified in the case of the instantaneous CMEM with A/D data. As the table suggests, ignoring A/D rates leads to suboptimal solutions in all cases.

In Table 2-10, we have also reported the total computational time required for the generation of all driving cycles for all road-links in the network at all time periods in case of each instance, under the column 'DCs generation runtime'. Observe that in comparison with field data collection, the required computational cost reported in the table can be considered insignificant.

2.8 Discussion and concluding remarks

In the paper presented in this chapter, we introduced a realistic urban freight distribution model that can address traditional business and environmental objectives simultaneously while integrating several factors affecting fuel consumption, on the original roadway network. The proposed model is a variant of the well-known PRP, called the SPRP, and is a multi-objective, time and load dependent, fleet size and mix, emissions minimising vehicle routing and scheduling problem, with time windows, flexible departure times, and multi-trips on congested urban road networks. The paper focused mainly on a key complication arising from emissions minimisation in a time and load dependent setting, corresponding to the identification of the full set of the eligible road-paths between consecutive truck visits. It was shown that the state-of-the-art pre-processing approaches are unable to extract all such paths from the underlying roadway graph and thus lead to sub-optimal solutions with an optimality gap of as high as 12% in terms of fuel consumption. It was also observed that compared with the proposed approach in the paper, the other approach based on the TDSPs has a very limited ability to identify true ND points on the Pareto front in a multi-objective case, where it is only able to identify less than 18% of the true ND points on average. Further experiments on a real road network based on Chicago's arterial streets indicated that the set of eligible road-paths between a given origin/destination pair can be so large, and while TDSPs constitute a very limited subset of these paths, the PEP can identify them all.

All models in the paper are based on the instantaneous CMEM formula and can incorporate second-by-second speed variations and thus A/D rates for a more accurate estimation of fuel consumption. However, acknowledging the fact that such well-grained speed data are rarely available to the decision maker at the planning stage, the paper proposed a simple optimisation model for the construction of reliable spatiotemporal driving cycles that with very few model inputs and no parameter tuning can yield synthetic driving cycles that very well approximate the expected real-life fuel consumption of the truck. Experiments that are carried out on over 19000 different on-road driving cycles confirm an average over-estimation of less than 2% for the proposed model. This model hence can replace the costly and time-consuming data acquisition phase for attaining reliable figures on the expected fuel consumption in routing applications.

The proposed model in the paper also shed light on an interesting opportunity to further cut down on GHG emissions and costs by using more energy-and-cost-efficient resources in the fleet multiple times during the planning horizon through multi-trip optimisation; especially in urban areas where trips are rather short, and trucks could be simply reloaded and dispatched for an extra round of trip. Multi-trip optimisation in a time-dependent setting, however, has never been studied before, and would constitute an interesting line of research due to new and previously unvisited challenges that it brings about when tackling real life size test instances.

The SPRP is a realistic and hence a very difficult problem to solve, and the main limitation of the current work lies in its inability in addressing large practical size SPRP instances. We develop tailored multi-objective optimisation heuristics for the SPRP in the next chapter of this thesis to cope with this situation. While there are multiple research opportunities relevant to the study of EMVRPs directly on the roadway networks, we identify the incorporation of the effect of non-recurrent congestion in the routing decisions through the development of real-time or stochastic variants of the SPRP as a significant line for future research. Moreover, as a recommendation for further research, the development of realistic problem instances that can reflect the real daily congestion patterns, and allow the analyses of different what-if scenarios for the departure time, depot location, customers' demands and time-windows negotiation, planning horizon alterations, and fleet size and mix decisions, can help gaining many practical insights for a logistics system operating in an urban area.

3. MULTI-OBJECTIVE OPTIMISATION HEURISTICS FOR THE STEINER POLLUTION ROUTING PROBLEM

3.1 Introduction

Urban Freight Distribution (UFD) is more polluting than long distance freight transport, generates between 20% and 60% of the local transport-based pollution, and represents about one fourth of CO_2 emissions coming from transport activities in European cities (Dablanc, 2009). Planning traffic-aware and fuel-efficient distribution routes that can realistically include the real operating conditions of urban roadway networks and address traditional business and environmental criteria simultaneously is inevitable to gain tangible reductions in the amount of *Greenhouse Gas (GHG)* emissions while fulfilling business objectives.

As a variant of the *Pollution-Routing Problem (PRP)*, the *Steiner PRP (SPRP)* has been recently introduced (Raeesi & Zografos, 2019) to deal with finding cost and emissions efficient distribution routes directly on the congested urban roadway networks. This variant is a tri-objective, time and load dependent, fleet size and mix PRP, with time windows, flexible departure times, and multiple-trips on road networks that aims at minimising the three objective functions pertaining to: (i) vehicle hiring cost, (ii) total amount of fuel consumed, and (iii) total makespan (duration) of the routes. The main distinctive features of the SPRP comprise: (i) studying the problem directly on the original roadway network in a time and load dependent setting, (ii) integrating all previously identified factors affecting fuel consumption, (iii) consideration of the expected time-dependent truck instantaneous *Acceleration/Deceleration (A/D)* rates over each road-link in the network for a more accurate estimation of fuel consumption, and (iv) bringing in an important decision regarding multiple uses of cost and energy-efficient resources in the fleet through multi-trip optimisation. The latter feature of the SPRP is particularly relevant in the context of UFD where delivery routes are rather short, and trucks could be usually utilised for one or more rounds of trip (Olivera & Viera, 2007).

Solving the SPRP to optimality is intractable even in case of small sized instances, and as it encompasses several hard variants of the *Vehicle Routing Problem (VRP)* and contains several distinctive and previously unvisited characteristics, available heuristic solution algorithms are unable to approach realistic instances of the problem efficiently within a reasonable computational cost. A major complication in the context of the SPRP arises from the emerging problem of intermediate road-path(s) identification on the original roadway graph. In a time-and-load-dependent setting, the consideration of a priori determined single road-path for travelling between consecutive truck visits is not possible, and determining a set of eligible paths requires a knowledge of the departure time from the origin node, the truck type utilised, and the load carried by the truck over the road-path; however, none of these factors are known prior to realising the full route plan and schedule. In the face of this complication, existing algorithms either employ a built-in path identification procedure to find such road-paths on-the-fly (Ehmke et al., 2016; Qian & Eglese, 2016), which can be computationally expensive, or rely on a limited subset of all such paths that are pre-computed in advance (Androutsopoulos & Zografos, 2017; Y. X. Huang et al., 2017). In addition to this primary complication, the features that are included in the SPRP call for solving several NP-hard multi-objective lower level optimisation problems including the route-path and departure time optimisation, fleet size and mix optimisation and multi-trip optimisation, which are intertwined optimisation decisions that must be made simultaneous with the consideration of the three conflicting objectives of the SPRP in order to be able to evaluate the efficiency of a single solution in the context of the problem. Hence, designing algorithms that can carry out all these tasks while tracing the non-inferior surface efficiently within a non-prohibitive computational time to solve real-life sized instances of the SPRP is a significant challenge that requires new algorithmic developments.

This paper proposes three new *Multi-Objective Optimisation Heuristics (MOOHs)* used within a multi-phase hybridised exact and heuristic solution framework to approximate the true *Pareto Optimal Set (POS)* of practical real-life sized instances of the hard to solve SPRP efficiently. The first algorithm of the paper is based on the hybridisation of an efficient *Mathematical Programming Technique (MPT)* with a two-stage *Local-Search (LS)* based heuristic, the second one is a hybrid *Multi-Objective Evolutionary Algorithm (MOEA)* with generational target attainment, while the third one is a simple order-first-split-second based MOEA. All these higher-level solution algorithms benefit extensively but differently from a new concept of ubiquitous external *Non-Dominated (ND)* solutions archive, new lower-level dedicated procedures for tackling the emerging optimisation problems that arise during SPRP solution evaluation, and a new neighbourhood exploration strategy. The proposed algorithms are tested on a library of SPRP benchmark test instances that are based on a network of Chicago's arterial streets, and another set of large time-dependent graphs resembling real world urban roadway networks. While the problem features in the SPRP are considerably more complicated and comprehensive than the problems considered in the existing related studies (Androutsopoulos & Zografos, 2017; Ehmke et al., 2016; Y. X. Huang et al., 2017; Qian & Eglese, 2016), the problem sizes accounted for in this paper go well beyond the ones solved by those studies at a more reasonable computational cost.

In the remainder of the chapter, first in section 3.2 we present a brief review of the previous literature. In section 3.3, we describe formally the SPRP and its modelling features including the travel time and fuel consumption estimation model. Section 3.4 elaborates on the proposed solution algorithms. Section 3.5 is devoted to the computational study of the algorithms; and finally, section 3.6 concludes the chapter.

3.2 Previous related work

The literature on the VRP and its variants is huge, and the SPRP relates well with several of those variants. For a general review on the state-of-the-art in the VRP and its variants the reader might be referred to the book of Toth and Vigo (2014) and the recent paper of Braekers et al.(2016). More specific reviews on green VRP (Bektaş et al., 2019a; Demir, Bektaş, et al., 2014b; Lin et al., 2014), *Time-Dependent VRP (TDVRP)* (Gendreau et al., 2015), *Multi-Objective VRP (MO-VRP)* (Jozefowiez et al., 2008), and heterogeneous VRP (Koç et al., 2016) are also available.

The main stream of research in the area of VRPs with explicit consideration of environmental performance most pertinent to the current study is related to emissions minimising vehicle routing models that consider alternative road-paths between the consecutive stops when routing on a congested urban roadway network. This line of research has very recently emerged in the literature (Androutsopoulos & Zografos, 2017; Ehmke et al., 2016; Y. X. Huang et al., 2017; Qian & Eglese, 2016; Raeesi & Zografos, 2018; Raeesi & Zografos, 2019), and is particularly very difficult to solve due to the problem of intermediate road-paths identification on the original roadway graph.

In Table 3-1, a summary of the key research papers in this area and the current study is provided. The SPRP attributes considered by the problems studied in these papers are indicated using tick marks, and their approach to deal with the problem of intermediate road-paths identification, their solution algorithms, test beds used, and problem sizes considered are also presented. In these studies, the intermediate road paths are either identified recursively as the algorithm constructs a solution (Qian & Eglese, 2016), or a limited subset of the eligible paths is precomputed in a pre-processing stage and used (Androutsopoulos & Zografos, 2017; Y. X. Huang et al., 2017). A third approach is also used by Ehmke et al. (2016) who identify a set of load-invariant paths a priori and find the remaining paths on-the-fly as the need arises. We, on the other hand, use the *Path Elimination Procedure (PEP)* proposed by Raeesi and Zografos (2019) that is applied on the original roadway graph and guarantees no ad-hoc Pareto optimal path is eliminated from the set of the retained road-paths.

Qian and Eglese (2016) propose a *Column Generation (CG)* based *Tabu Search (TS)* algorithm to solve the problem on the road network of London with instances containing 25 customers. Due to the need to call the interwoven path identification heuristic iteratively, the runtime of their original algorithm increases to over 20 hours on a high performance computer cluster; however, a less accurate distance-based approach they propose requires around 5 minutes for an acceptable approximation of the original algorithm. Ehmke et al. (2016) propose a TS algorithm to solve problems with 10 and 30 customers on a road network based on the metropolitan area of Stuttgart. The runtime of their algorithm varies as they change the experiments setting, but for example, solving a TSP in the context of their problem takes over 1800 seconds. Androutsopoulos and Zografos (2017) study the problem in a bi-objective context and propose an *Ant Colony System (ACS)* algorithm to solve generated test instances with a maximum of 50 customers, while at a computational time of over 3 hours due to the difficulty of the lower-level bi-objective time-dependent shortest path problem which must be solved at each iteration.

Table 3-1
Characteristics of the key pertinent studies

	Study characteristics	Studies				
		Qian and Eglese (2016)	Ehmke et al. (2016)	Androutsopoulos and Zografos (2017)	Huang et al. (2017)	Current study
SPRP attributes covered	Time-dependency	✓	✓	✓	✓	✓
	Load-dependency	✗	✓	✓	✓	✓
	Time-windows	✓	✗	✓	✓	✓
	Fleet mix	✗	✗	✗	✗	✓
	Fleet size	✗	✗	✗	✗	✓
	Alternative paths	✓	✓	✓	✓	✓
	Departure time	✗	✗	✓	✓	✓
	Multi-tour	✗	✗	✗	✗	✓
	Multi-objective	✗	✗	✓	✗	✓
	Fuel estimation accuracy	✗	✗	✗	✗	✓
Solution features	Paths identification approach	On-the-fly	Precomputed and on-the-fly	Precomputed subset	Precomputed subset	Exact complete set (PEP)
	Solution algorithm	CG-based TS	TS	ACS	CPLEX	Three MOOHs
	Test bed	London road network	Metropolitan area of Stuttgart	Synthetic instances	Urban area of Beijing	Chicago arterial streets and synthetic instances
	Problem size	25	30	50	20	100

We are aware of only two studies that consider *Emissions Minimising Vehicle Routing Problems (EMVRPs)* in a bi-objective context where fuel consumption and driving time are identified as the two conflicting objectives. Besides the paper by Androutsopoulos and Zografos (2017) discussed above, Demir et al. (2014a) use an *Adaptive Large Neighbourhood Search (ALNS)* algorithm as the search engine in four different a posteriori methods to solve the bi-objective PRP. Note that studies like that of Ehmke et al. (2018), are not categorised as multi-objective EMVRPs as the various objectives that are considered in these papers are either treated separately, or linearly aggregated to a single objective function. The wider literature on the multi-objective VRPs, on the other hand, has mostly focused on multi-objective *VRP with Time Windows (VRPTWs)* (Banos et al., 2013; Garcia-Najera & Bullinaria, 2011; Ghoseiri & Ghannadpour, 2010; Ombuki et al., 2006; Qi et al., 2015) and evolutionary algorithms constitute the most popular algorithms. It is worth mentioning that all the stated papers consider bi-objective problems and none of them considers more than two objective functions, and the current study is the first one in this respect.

Despite its important implications with regard to multiple uses of energy-efficient resources multiple times during the planning horizon, *Multiple-Trip (MT)* decision-making has not been incorporated into emissions minimising vehicle routing algorithms so far. Existing algorithms for MT-VRPs, however, could be broadly categorised into models with and without time-windows. While in models without time-windows (Olivera & Viera, 2007; Petch & Salhi, 2003; Salhi & Petch, 2007; Taillard et al., 1996) routes could be easier assigned to the vehicles by solving simpler lower-level optimisation problems such as the *Bin-Packing Problem (BPP)*, MT-VRPTWs (Brandao & Mercer, 1998, 1997; Cattaruzza et al., 2016; Cattaruzza, Absi, Feillet, & Vigo, 2014; Macedo et al., 2011) are more challenging to deal with as the sequence in which routes are assigned to trucks affects the feasibility of the solutions.

Overall, this brief review suggests that algorithms to solve VRPs directly on the original roadway networks with the consideration of alternative road-paths between consecutive truck visits are still very rare (Androutsopoulos & Zografos, 2017; Ehmke et al., 2016; Garaix et al., 2010; Qian & Eglese, 2016), and due to the difficult nature of the problem, these algorithms are unable to approach real-life sized instances within a reasonable computational budget. Therefore, it is a significant open research direction to develop new efficient approaches that can address such problems. Furthermore, the literature is significantly lagging behind in developing solution algorithms that can take into account the realistic conditions of UFD while unifying several factors affecting fuel consumption in a multi-objective context. This paper is trying to address these gaps.

3.3 Formal description of the SPRP and its modelling features

The SPRP is defined on a directed graph $G = (N, A)$, representing a real roadway network, with N being the set of network nodes and A the set of directed road-links. The set $N = \{N_0 \cup N_1 \cup N_2\}$ is comprised of the depot $N_0 = \{0\}$, customer nodes $N_1 = \{1{,}2, \dots, n\}$, and network junctions $N_2 = \{n + 1, \dots, n + m\}$. There is an unlimited fleet of heterogeneous vehicles set K to hire from, which is assumed to be composed of $\mathcal{k}$ different types of trucks. To each truck $k \in K$ a curb weight μ_k (*kg)*, a maximum payload Q_k (*kg)*, and a daily hiring fixed cost c_k (*£)*, among other vehicle-specific factors such as engine friction factor, engine speed, engine displacement, coefficient of aerodynamic drag, and frontal surface area is attributed. Each customer $i \in N_1$ is associated with a certain demand $q_i \leq \max\limits_{k \in K} Q_k$ to be delivered within its pre-determined hard time window denoted by $w_i = [e_i, l_i]$, with service time s_i. The depot working hours, which is considered as the planning horizon, is denoted by $T = w_0 = [e_0, l_0]$, and while it is assumed that trucks are initially loaded, reloading them for operating a new tour takes s_0 time at the depot. To each road-link $(i,j) \in A$, a distance d_{ij}, and an integer time-dependent travel time $\mathscr{t}_{ij}^{\tau}$, depending on the departure time from the origin node i, i.e. $\tau \in [e_0, l_0]$, is attributed. To estimate the $\mathscr{t}_{ij}^{\tau}$s, in this paper we use the closed-form formula proposed in Raeesi and Zografos (2019). The aim of the SPRP is to determine an optimal composition of vehicles in the fleet to operate routes that start and finish at the depot and serve every customer exactly once within their pre-defined time-windows, without violating vehicle capacities and working day limits, such that the three objectives pertaining to: (i) vehicle hiring cost, (ii) total amount of fuel consumed, and (iii) total makespan (duration) of the tours, are minimised.

In the SPRP the spatiotemporal instantaneous driving cycles DC_{ij}^{τ}, denoting the expected second-by-second speed variations for each road-link $(i,j) \in A$ and for all time instants $\tau \in T$, are assumed either available or generated synthetically using the proposed model in Raeesi and Zografos (2019). Given such cycles, the instantaneous time, load and truck-type dependent fuel consumption (in litres) over the given road link for departure time $\tau \in T$ could be computed using the instantaneous CMEM formulae of Barth et al. (2004), which could be summarised as follows:

$$\mathcal{F}_{ij}^{k\tau} = Z_{ij}^{k\tau} + (\mu_k + f_{ij})\mathit{\Gamma}_{ij}^{\tau}, \qquad \forall (i,j) \in A, \tau \in T \tag{3-1}$$

Expression (3-1) divides CMEM into a time-dependent term *Unrelated to Truck Mass* (called the UTM attribute hereafter), and a time-dependent term linearly *Related to the Truck Mass* (called the RTM attribute hereafter), and both of these could be precomputed and stored for all road-links (or road-paths) at all possible departure times based on the available DC_{ij}^{τ}s (for a full exposition refer to Raeesi and Zografos, 2019).

In this study, for experimental purposes, similar to the work of Koç et al. (2014) on the fleet size and mix PRP, we consider the fleet to be composed of light, medium and heavy duty trucks and use the same values they use for the common and vehicular specific parameters, which they obtain for the three main vehicle types of MAN Trucks (see Tables 1 and 2 in Koç et al. - 2014).

3.4 Solution algorithms

The main source of complication that inhibits a straightforward application of the existing VRPTW heuristics on the SPRP is due to the difficulty of solution evaluation in the context of the SPRP. In the case of the conventional VRPTWs, once the sequence of customer visits for a given truck (a route) is known, it is straightforward to calculate the cost of the corresponding route; however, the same route can have an exponential number of SPRP evaluations depending on the objective function that is considered, the road-path that is taken between every pair of visits, the time that the depot is departed, and the type of truck that is used and if it is an extra round of trip for the selected truck or not; and these are all intertwined optimisation decisions that must be made simultaneously in order to be able to evaluate a single solution to the SPRP.

We propose that the key to handle the mentioned complications is in: (i) decomposing the interrelated optimisation problems that must be solved, while preserving the necessary interactions among them, (ii) reducing these optimisation problems to single-objective problems whenever possible while applying a higher-level control on the selected cost function for ensuring that ad-hoc ND solutions are not significantly eliminated, and (iii) designing a built-in implicit search scheme that operates independently from the main search direction of the algorithms and stores all encountered ND solutions.

Therefore, to solve the SPRP a multi-phase solution framework that is shown in Figure 3-1 is used. In the first phase of the proposed algorithm, the exact Path Elimination Procedure of Raeesi and Zografos (2019) is applied on the underlying roadway network in order to discard all proven to be redundant paths and retain only eligible paths that are potential to contribute to the generation of ND solutions. The output from the PEP is a multi-graph containing a set of pre-computed road-paths between the required nodes with a complete archive of their distance, time-dependent travel times, UTM and RTM attributes in easy-to-access look-up tables, which greatly facilitates the application of any subsequent solution algorithms. Following the application of the PEP, the problem is solved on the resulting multi-graph using the MOOHs proposed in this paper for approximating the true POS of the SPRP.

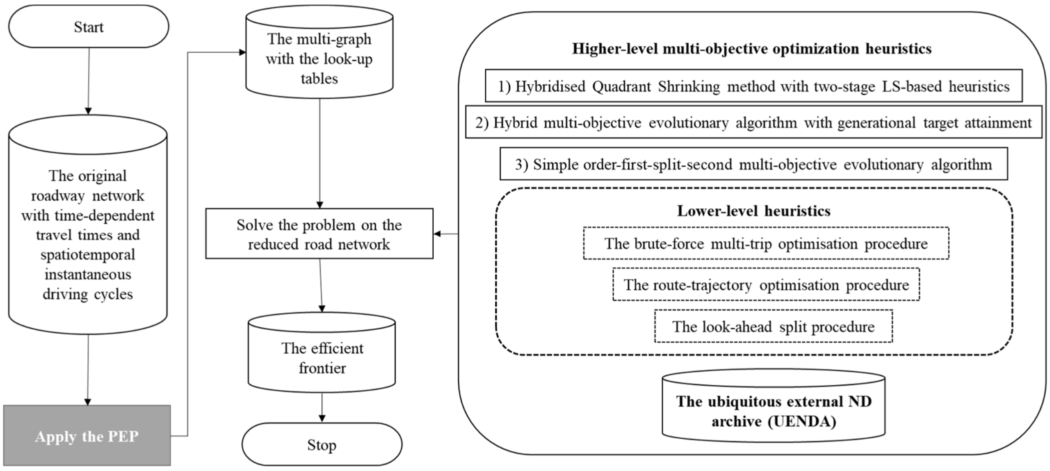

Start
The original roadway network with time-dependent travel times and spatiotemporal instantaneous driving cycles
Apply the PEP
The multi-graph with the look-up tables
Solve the problem on the reduced road network
The efficient frontier
Stop
Higher-level multi-objective optimization heuristics
1) Hybridised Quadrant Shrinking method with two-stage LS-based heuristics
2) Hybrid multi-objective evolutionary algorithm with generational target attainment
3) Simple order-first-split-second multi-objective evolutionary algorithm
Lower-level heuristics
The brute-force multi-trip optimisation procedure
The route-trajectory optimisation procedure
The look-ahead split procedure
The ubiquitous external ND archive (UENDA)

Figure 3-1 The proposed multi-phase solution framework for the SPRP

As indicated in Figure 3-1, we are proposing three different higher-level MOOHs that use extensively (but differently) several lower-level heuristics to carry out LS tasks, and deal with the lower-level optimisation problems that arise in the context of the SPRP. The success of the proposed algorithms greatly relies on a *Ubiquitous External ND Archive (UENDA)* scheme that is actively present in all levels of the proposed algorithms in order to store all encountered ND solutions implicit to the main search direction of the algorithm.

In the rest of this section, we begin by introducing the proposed lower-level heuristics, and then we discuss how they are employed by the higher-level MOOHs.

3.4.1 Lower-level heuristics

The new heuristics that are proposed for the lower-level decomposed problems of route-trajectory optimisation, fleet size and mix optimisation, and multi-trip optimisation are discussed in this section. For brevity, we refer the reader to Raeesi and Zografos (2019) for detailed definitions relevant to scheduled road-path, route-path, and route-trajectory that are required in the rest of the paper.

3.4.1.1 The Threshold-Based Route-Trajectory Optimisation (TB-RTO) procedure

The multi-objective route-path optimisation problem, which is called the *Fixed Sequence Arc Selection Problem (FSASP)* by Garaix et al. (2010), corresponds to the *Multidimensional Multiple Choice Knapsack Problem (MMKP)*, which is an NP-hard generalization of the knapsack problem (Garaix et al., 2010). To deal with FSASP, Garaix et al. (2010) and Androutsopoulos and Zografos (2010) use pseudo-polynomial solution algorithms based on dynamic programming. The evaluation of every newly generated candidate solution in the course of the LS using these algorithms, however, is very computationally expensive and prohibitive.

The optimisation of the departure time from the depot, on the other hand, has implications both in terms of the travel time and the fuel consumption of the truck operating the tour. Departure time optimisation, however, cannot be carried out independently from route-path optimisation, because any given departure time for a given sequence of road-paths corresponds to objective values that might differ from a different sequence of road-paths for the same departure time, and still both of the two solutions might be non-dominated. As a result, these two decisions should be made together by solving a joint optimisation problem, called the *Route-Path and Departure Time Optimisation (RPDTO)* problem, or by definition, the *Route-Trajectory Optimisation (RTO)* problem.

At least two main difficulties are associated with incorporating any RTO procedure into the search process. The first one certainly arises from the difficulty of solving the corresponding NP-hard multi-objective optimisation problem, and the second one is deciding on the frequency at which this possibly costly procedure should be called during the search. In this paper, to tackle the first difficulty, we are proposing to deal with a much easier to solve single-objective RTO problem, that by incorporating the *UENDA* scheme and applying a higher-level control on the determined single-objective function, tries to approximate the true set of ND solutions associated with the original multi-objective problem. Solving the RTO problem as a single objective problem can be done very efficiently, since as Raeesi and Zografos (2019) prove, in a single objective case if departure time is not restricted, an optimal route-trajectory could be found that uses the "cheapest" road-path between every two consecutive visits. In order to address the second difficulty stated above, we establish a subtle balance between the overall computational time of the algorithm and the solution quality by calling the RTO procedure only in promising areas of the solution space. This is done by examining each new candidate solution against a pre-set threshold value to see if it is worth searching for its optimal departure time.

A description of the proposed *TB-RTO* procedure is given in Algorithm 3-1. Very generally, suppose that x is a vector of the decision variables of the problem at hand, and $f(x)$ is the objective function to be minimised by the *TB-RTO* procedure for a given solution $\mathcal{S} = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_{\Bbbk}\}$ representing a set of $\Bbbk$ truck trips $\mathcal{R}_{\mathscr{s}} \in \mathcal{S}$, where $\mathcal{R}_{\mathscr{s}} = \{0,1,2,\dots,n,0\}$ is a trip that starts from the depot $\{0\}$ serves customers in $\mathcal{C}_{\mathscr{s}} = \{1,2,\dots,n\}, \mathcal{C}_{\mathscr{s}} \subseteq N_1$, and returns to the depot. This objective function is decided by the higher-level heuristics and will be discussed duly in appropriate sections of the paper.

Algorithm 3-1 `TB-RTO`

```
input $f(x)$, $\mathcal{S}$, UENDS, $\delta$, $S$, $f^*$
$f(\mathcal{S}^{\tau^*}) = 0$
for $\mathscr{s} = 1$ to $\mathbb{K}$ do
    $\tau = e_0$
    $(f(\mathcal{R}_{\mathscr{s}}^{\tau})$, UENDA$) :=$ Route-path-opt ($f(x)$, $\mathcal{R}_{\mathscr{s}}$, load($\mathcal{R}_{\mathscr{s}}$), $\tau$, UENDA)
    $f(\mathcal{S}^{\tau^*}) = f(\mathcal{S}^{\tau^*}) + f(\mathcal{R}_{\mathscr{s}}^{\tau})$
    $\tau^* = \tau, f(\mathcal{R}_{\mathscr{s}}^{\tau^*}) = f(\mathcal{R}_{\mathscr{s}}^{\tau})$
end for
if $f^* \geq \delta f(\mathcal{S}^{\tau^*})$ then
    for $\mathscr{s} = 1$ to $\mathbb{K}$ do
        while $\tau \neq l_0$ do
            $(f(\mathcal{R}_{\mathscr{s}}^{\tau})$, UENDA$) :=$ Route-path-opt ($f(x)$, $\mathcal{R}_{\mathscr{s}}$, load($\mathcal{R}_{\mathscr{s}}$), $\tau$, UENDA)
            if $f(\mathcal{R}_{\mathscr{s}}^{\tau}) \leq f(\mathcal{R}_{\mathscr{s}}^{\tau^*})$ then $\tau^* = \tau, f(\mathcal{R}_{\mathscr{s}}^{\tau^*}) = f(\mathcal{R}_{\mathscr{s}}^{\tau})$ end if
            If departing at time $\tau$ is infeasible then break end if
            $\tau = \tau + S$
        end while
    end for
end if
return $\tau^*, f(\mathcal{S}^{\tau^*})$, UENDA
```

The key role in the `TB-RTO` is played by a route-path optimisation function (`Route-path-opt`) that selects a sequence of road-paths in the current vehicle route, i.e. $\mathcal{R}_{\mathscr{s}}$, and minimises $f(\mathcal{R}_{\mathscr{s}}^{\tau})$ for a given time instant τ. This is a simple function that between every pair of visits i and $i+1$ only needs to select the path $p_{ij} \in \mathcal{p}_{ij}$ ($\mathcal{p}_{ij}$ is the set of all retained PEP path between nodes i and j) that minimises $f(x)$. Therefore, the cost of this trip for departure time τ, i.e. $f(\mathcal{R}_{\mathscr{s}}^{\tau})$ based on the objective function $f(x)$, is returned by `Route-path-opt` using the following expression:

$$f(\mathcal{R}_{\mathscr{s}}^{\tau}) = \min_{p_{01} \in \mathcal{p}_{01}} f(p_{01}^{\tau}) + \min_{p_{12} \in \mathcal{p}_{12}} f(p_{12}^{\tau_1}) + \cdots + \min_{p_{n0} \in \mathcal{p}_{n0}} f(p_{n0}^{\tau_2}) \qquad (3\text{-}2)$$

where $\tau_1 = max\{e_1 + s_1, \tau + \min_{p_{01} \in \mathcal{p}_{01}} \mathcal{t}(p_{01}^{\tau}) + s_1\}$, and so on.

It is worth mentioning that the calculation of (3-2) is done in $\mathcal{O}(n)$ thanks to the PEP which has already made access to all paths attributes possible in look-up tables. Also, since infeasible path attributes for certain departure times (e.g. violating the upper boundary of time-windows) are already set to ∞ in the PEP look-up tables, feasibility check for RTO can be done very quickly. Note that the

`Route-path-opt` also takes UENDA in its input and checks if it must be updated.

To optimise route-trajectory of a given truck trip $\mathcal{R}_{\mathcal{s}} \in \mathcal{S}$ now, i.e. to find $f(\mathcal{R}_{\mathcal{s}}^{\tau^*})$, the following task must be carried out by `TB-RTO`:

$$f(\mathcal{R}_{\mathcal{s}}^{\tau^*}) = \min_{\tau \in T} f(\mathcal{R}_{\mathcal{s}}^{\tau}) \tag{3-3}$$

To carry out this task efficiently, the `TB-RTO` initially starts only by getting the `Route-path-opt` function to find $f(\mathcal{R}_{\mathcal{s}}^{e_0})$ for all $\mathcal{R}_{\mathcal{s}} \in \mathcal{S}$, and thus $f(\mathcal{S}^{\tau^*}) = \sum_{\mathcal{s} \in \mathcal{S}} f(\mathcal{R}_{\mathcal{s}}^{e_0})$ (lines 2 to 8). In its input, the `TB-RTO` always has some information regarding the f^*, i.e. the value of the best solution with regard to $f(x)$ found so far by the higher-level algorithm. Hence, in line 9 of the algorithm, the threshold check can be carried out, by checking the value of $f(\mathcal{S}^{\tau^*})$ against f^*, such that lines 10 to 17 are run iff $f^* \geq \delta f(\mathcal{S}^{\tau^*})$; where $\delta_{min} \leq \delta \leq 1$ is a user defined parameter and, without loss of generality, $\delta_{min} = \max_{p_{ij} \in \hat{A}}(\min_{(\tau \in T)} \boldsymbol{t}(\boldsymbol{p}_{ij}^{\tau}) / \max_{(\tau \in T)} \boldsymbol{t}(\boldsymbol{p}_{ij}^{\tau}))$. The closer the selected δ is to the upper bound of its defined interval, the less likely it is to call the complete RTO for new candidate solutions, and therefore, the faster will the algorithm run, but also the lower might be the ultimate quality of the solutions found.

If the threshold in line 9 of the algorithm is met, the stepwise search for optimising the route-trajectory of each trip must be carried out throughout the planning horizon; however, as the time-dependent travel times satisfy the non-passing property, well-known as the *First-In-First-Out (FIFO)* property, that ensures a later start time cannot lead to an earlier arrival time, there is no need to sweep the entire time horizon and the algorithm can break out of the loop prematurely (line 14):

Remark 1 *Due to the FIFO property, if it is infeasible to depart the origin node of a trip $\mathcal{R}_{\mathcal{s}}$ at some time $\tau \in T$, then it is also infeasible to depart the origin at any later time $\tau^1 \in T, \tau^1 \geq \tau$.*

In addition to this exit condition, to speed up the search in the planning horizon a larger time step (S) could be used (line 15). Therefore, the complete `TB-RTO` algorithm is an $\mathcal{O}(nT/S)$ algorithm (if the threshold is met). With this algorithm, while we can save greatly on the number of times that it is required to be run during the solution process, it can be ensured that potential search directions are not missed.

The algorithm for determining the fleet size and mix of a given solution is described next.

3.4.1.2 Fleet size and mix optimisation: the look-ahead split procedure

Order-first split-second methods have recently led to successful evolutionary based algorithms for various VRPs (Prins, 2009), and in particular, for the *Fleet Size and Mix VRP (FSMVRP)* and the *Heterogeneous Fixed Fleet VRP (HFFVRP)* (Koç et al., 2015; Liu et al., 2009; Prins, 2009). In these methods, a solution is represented as a permutation of customers without trip delimiters, and can therefore be viewed as a giant TSP tour for a vehicle with infinite capacity, which is then optimally partitioned into a set of feasible vehicle trips by applying a tour splitting procedure. Using this strategy, the algorithm searches the set of TSP tours, which is much smaller than the set of FSM and HFFVRP tours (Prins, 2009). For a complete review of the state-of-the-art order-first split-second methods, the interested reader is referred to Prins et al. (Prins et al., 2014).

In a multi-objective case, the split procedure is NP-hard and very computationally expensive. Therefore, using a similar approach to the case of the RTO procedure we rather deal with a simple single-objective split procedure whose objective function $f(x)$ is adaptively decided by the higher-level heuristic algorithm. This procedure is also accompanied by the *UENDA* to archive all ND solutions encountered during the search.

It is worth mentioning that MT optimisation has previously been integrated with the split procedure (Cattaruzza et al., 2016; Cattaruzza, Absi, Feillet, & Vidal, 2014). However, as we will discuss in the next sub-section, it is more efficient to keep these two operations separated, and instead, for getting the best results, a look-ahead extension of the split (`LA-split`) is proposed in this paper that tries to leave room for subsequent MT decisions basically by 'rewarding' the shorter arcs in the auxiliary acyclic graph (refer to Prins et al. (2014) for the description of the auxiliary graph in the context of the split procedure). Each arc in the auxiliary graph represents a vehicle tour, and clearly the shorter is a vehicle tour, the more it will be likely for the vehicle to operate an extra round of trip during the planning horizon. In the look-ahead split we reduce the original cost of the shorter trips that reinforce the chance of MT for the vehicles based on a "*Rewarding Policy (RP)*", so that these arcs get a higher chance to appear in the optimal split of the giant tour. For these arcs to get rewarded based on the RP, however, they are checked against some "*Rewarding Conditions (RCs)*" which are generally a duration check against some maximum allowable durations. The ultimate split is sensitive to the selected RP, and therefore it should be determined with caution, such that it only reflects the value of MT to the solution with regard to the current objective of interest, and not more than that. For example, assume the single objective function of the split determined by the higher-level heuristic is to minimise an aggregated function of all the three objective functions to the SPRP. Also suppose that a maximum of three rounds of trips are allowed for each truck during the planning horizon. Then, if the duration of an arc is less than 1/3 of the planning horizon (RC), a good RP can be to allocate a cost equal to "the cost of the vehicle/3 + fuel cost + travel time" to the corresponding arc, as opposed to the original cost of the arc which is "the cost of the vehicle + fuel cost + travel time". In our implementation, the RP is adaptively selected for the `LA-split` based on the objective function determined for the split by the higher-level heuristic.

Algorithm 2 `LA-split`

```
input $f(x)$, GT, RP, RC(s), UENDA
$V_0 = 0$;
for $i = 1$ to $n$ do $V_i = +\infty$ end for
for $k = 1$ to $n$ do
    $j = 1, load = 0$
    while $j \leq n$ and $load \leq Q_{\mathcal{H}}$ do
        $load = load + q_{GT_j}$
        if $i = j$ then
            $PR = \{0, GT_i, 0\}$
            $(f(PR^{e_0})$, UENDA) ≔ Route-path-opt ($f(x)$, PR, load(PR), $e_0$, UENDA)
            if RC(s) are met then apply RP on $\mathbb{Z}_{PR}$ end if
        Else
            $PR_{|PR|} = \emptyset$
            $PR = \{PR, GT_j, 0\}$
            $(f(PR^{e_0})$, UENDA) ≔ Route-path-opt ($f(x)$, PR, load(PR), $e_0$, UENDA)
            if RC(s) are met then apply RP on $f(PR^{e_0})$ end if
        end if
        If $load \leq Q_{\mathcal{H}}$ and $V_{i-1} + \mathbb{Z}_{PR} \leq V_j$ then $V_j = V_{i-1} + \mathbb{Z}_{PR}, P_j = i - 1$ end if
        $j = j + 1$
    end while
end for
return $V, P$, UENDA
```

A description of the proposed look-ahead split procedure is shown in Algorithm 3-2. An objective function $f(x)$ is supposed to be minimised for a given giant tour (GT) by the `LA-split` procedure. Two labels V and P are computed in this algorithm for each node $j \in \{1, . . , n\}$, in order to record the cost of the shortest path from node $\mathbf{0}$ to node j in the auxiliary graph, and to point to the predecessor of j on this path, respectively. In each loop of the algorithm a partial route (PR) which represents one arc in the auxiliary graph is examined using the `Route-path-opt` function, and if its cost for the earliest departure time e_0, i.e. $f(PR^{e_0})$, meets the determined RC(s), then it is rewarded based on the RP (line 16). Note that, in this algorithm $Q_{\mathcal{H}}$ refers to the payload of a heavy duty truck. Also, note that the beginning of the planning horizon is chosen because based on the FIFO principle stated in Remark 1, if the arc is not feasible at the earliest possible departure time, i.e. e_0, then it is also infeasible for all later departure times. The yielded solution by `LA-split` can undergo departure time optimisation later using `TB-RTO` if the threshold in `TB-RTO` is satisfied.

It is discussed next how the possibility of the MTs could be checked for a given SPRP solution.

3.4.1.3 The Brute-Force Multi-Trip Optimisation (BF-MTO) procedure

Multi-Trip Optimisation (MTO) can contribute significant savings to the total vehicle cost incurred by hiring trucks to operate the trips, and the total amount of fuel consumed at routes by using environmentally-efficient resources more than once. As stated earlier, to solve the MTO problem heuristically, an approach unified with the split procedure (similar to the one in Cattaruzza et al., 2016) can be employed. However, besides its computational cost, such integration, especially in a time-dependent setting with a heterogeneous fleet, does not allow a complete examination of all MT possibilities. The main reason is that with this approach it is only possible to position a trip after the trips that are already assigned to the vehicle, as arcs are assigned to a vehicle in the order they appear in the path. As a result, later trips (or arcs in the auxiliary graph) are never given a chance to be the first trips allocated to the vehicles. In the time-dependent setting, the travel time of each trip depends on the time the depot is departed and therefore, while an allocation of two trips A and B to a vehicle might be infeasible in the order A-B, the alternative allocation of B-A is not essentially infeasible. Furthermore, in the case of a heterogeneous fleet, in addition to the aforementioned shortcoming, many more MT possibilities are simply ignored due to heterogeneity of vehicles carrying out the trips. For example, an optimal split for an FSM problem that results in three trips that are carried out by a light, medium, and heavy-duty truck, respectively, may imply the impossibility of MTs, since the payload of a light-duty truck is not large enough to operate the trip allocated to a medium-duty truck, and so on. However, it is clear that (as one of the several MT possibilities) a heavy-duty truck can take care of all the three trips if temporal constraints do not get violated.

Therefore, to solve the MTO problem, we propose a new procedure that takes all forward and backward combinations, and all possible vehicle variations into account, and outputs the optimal multi-trip scenario with clear schedules. This procedure is an all-possibility-check procedure and is called the *Brute-Force MTO* (`BF-MTO`) procedure. Since `BF-MTO` is an exhaustive search algorithm, it can be time consuming to examine all the possible combinations of multi-trips, many of which might turn out to be infeasible. However, to speed-up the `BF-MTO` procedure, several quick feasibility checks, mainly based on the temporal characteristics of the trips, such as the time-windows information and the time-dependent travel time of the trips could be utilised.

As stated above, in a time-dependent setting if for a given departure time instant τ of trip B, it could not be allocated to the same vehicle operating trip A, it might still be possible to do so by departing the depot at another departure time.

However, checking the entire planning horizon for finding such possibly existing departure time is very time consuming and unreasonable. Instead, a helpful property of the FIFO principle could be exploited to check the feasibility of such allocation by checking only one time instant, i.e. the beginning of the planning horizon:

Remark 2 *Based on the FIFO property, if a truck that departs the depot in the beginning of the planning horizon to operate trip A cannot accommodate trip B after trip A, then trip B can never be operated after trip A by a single vehicle, at no other time during the planning horizon.*

The clear reason for this is that a later departure time from the origin cannot lead to an earlier arrival time at the destination. Based on Remark 2, if *A*-*B* is infeasible, then all other combinations of trips containing the same sequence of trips *A* and *B*, such as *A*-*B*-*C* and *A*-*C*-*B* are also infeasible and there is no need to check them. However, note that still the opposite order *B*-*A* can be feasible and must be checked.

If information about the optimal travel times of the trips is available (which is the case in our solution algorithms as MTO is always carried out after RTO), another very helpful feasibility check that can quickly eliminate the need for checking lots of multi-trip possibilities could be used:

Remark 3 *Suppose* $\boldsymbol{t}^*(A)$ *and* $\boldsymbol{t}^*(B)$ *denote the optimal time-dependent travel times of two different trips A and B, respectively. If* $\boldsymbol{t}^*(A)+\boldsymbol{t}^*(B)+s_0 > T$, *then allocation of any combination of these two trips, and any other combination of these trips and other trips, to a single truck of any type is infeasible.*

Information from time windows (especially in case of tight time windows) could be also used for quick feasibility checks, such as comparing the arrival time of trip A with the smallest upper boundary of time windows of all customers on trip B.

Similar to the cases of `TB-RTO` and `LA-split`, the `BF-MTO` procedure is accompanied by the *UENDA*, and we assume that a single objective function $f(x)$ (decided by the higher-level heuristic) must be minimised. Given that a maximum of r trips are allowed for each truck during the planning horizon, the implementation of the `BF-MTO` requires r nested loops, where the uppermost outer loop checks the first round of trip for each vehicle $k \in \{1, .., \mathbb{K}\}$, the first inner loop the second round of trip and so on. This implementation is $\mathcal{O}(n\mathbb{K}^r)$, however, due to the discussed feasibility checks, in practice most of the loops are never executed and the algorithm is sufficiently fast. In our experiments, we have been routinely able to use an $\mathcal{O}(n\mathbb{T}\mathbb{K}^r)$ implementation to also optimise the departure times for the trips in each MT.

In what follows, we describe the higher-level MOOHs and the way the described heuristics are used by them.

3.4.2 Higher-level solution algorithms

Three different higher-level MOOHs for the SPRP are proposed in this paper and are compared with one another. The first method, which is the main MOOH of the paper, hybridises an efficient *Mathematical Programming Technique (MPT)* with a two-stage LS-based heuristic for the first time to solve a multi-objective optimisation problem with more than two objectives. The main benefit of this hybridisation is that all emerging lower-level multi-objective optimisation problems needed to be solved in the context of the SPRP could be reduced to single objective problems, which can be handled much more efficiently. Moreover, using a MPT as the pedestal of search for ND solutions has the added advantage of more systematic non-inferior surface tracing.

The second and the third methods are based on the well-established concept of the *Multi-Objective Evolutionary Algorithms (MOEAs)*. The second MOOH is a *Hybrid MOEA (HMOEA)* with a generational target attainment scheme that particularly aims at preventing the algorithm from wandering around the huge feasible solution region of the SPRP, and guiding the search better towards and along the Pareto front. The third MOOH is a MOEA which has a much simpler structure than the second MOOH and is more agile.

The rest of this section introduces these higher-level solution algorithms, the metaheuristics and the new neighbourhood exploration strategy employed by them.

3.4.2.1 MOOH I: Hybridised Quadrant Shrinking Method with a Simulated Annealing and a Memetic Algorithm (HQSM-SA-MA)

Despite several successful applications in solving hard optimisation problems (Becerra & Coello, 2006; Demir, Bektaş, et al., 2014a; Ranjithan et al., 2001; Srigiriraju, 2000), the integration of MPTs that guarantee the identification of the full set of the ND solutions, and LS-based heuristics is a promising line of research that has drawn insufficient attention in the literature. The main reason why this sort of hybridization is usually refrained is its supposedly high computational cost (Ranjithan et al., 2001; Srigiriraju, 2000). However, as Becerra and Coello (2006) argue, if the single-objective optimiser is well-designed and implemented, this hybridisation is able to generate the true Pareto front of very difficult multi-objective optimisation problems at a reasonable computational cost. The authors demonstrate this by solving a number of hard test problems that are considered to be very difficult to be solved by current MOEAs, and show that in most cases, even when performing a very high number of fitness function evaluations, the employed MOEA is unable to reach the true Pareto front, while the MPT-based algorithm they propose is able to converge to the true Pareto front (or very close to it) of all the problems (Coello et al., 2007).

To the best of our knowledge, most of the existing algorithms of this sort of hybridisation mainly rely on the MPTs based on the ϵ-constraint approach in a bi-objective context, and no such algorithm has so far been reported for solving a problem with more than two objectives. Noticeable developments have been made recently in the area of MPTs for tri- and multi-objective optimisation of *Integer Programming (IP)* problems, and efficient algorithms have been proposed that can find the full set of the ND solutions. Reviewing the bounds on the total number of IPs required to be solved by the state-of-the-art MPTs, and the difficulty of these IPs at each iteration, Raeesi and Zografos (2019) conclude that the Quadrant Shrinking Method (QSM) by Boland et al. (2017b) offers the most promising performance among the existing approaches.

The reader is referred to Boland et al. (2017b) for an introduction to the QSM, but very concisely, similar to any other method for generating the ND frontier of a *Multi-Objective IP (MOIP)*, the core operation of the QSM is searching for an as-yet-unknown ND point. An ND point $z^n = z(x^n)$ with the property that its projection $\overline{z^n}$ satisfies $\overline{z^n} \leq u$ for a given point u in the projected space, if one exists, is found by solving two IPs, through a two-stage scalarisation technique. First, an intermediate point $x^i \in \mathcal{X}$ with minimal third objective value over points $x \in \mathcal{X}$ with $\bar{x} \leq u$ is found via: $x^i \in \arg min\{z_3(x){:}\, x \in \mathcal{X} \ and \ z_k(x) \leq u_k, k \in \{1{,}2\}\}$. If this IP is feasible, it is followed by a second IP that converts the weakly efficient solution x^i into an efficient solution x^n: $x^n \in \arg min\{\sum_{k=1}^{3} z_k(x){:}\ x \in \mathcal{X} \ and \ z_k(x) \leq z_k(x^i), k \in \{1{,}2{,}3\}\}$. This search is denoted by `2D-NDP-Search(u)`, and if the first IP is infeasible, `2D-NDP-Search(u)` returns *Null* and x^i does not exist. Otherwise, if x^i exists, the second IP must be feasible and `2D-NDP-Search(u)`returns x^n. Ultimately, this search returns an ND point z^n with z_3^n minimal over those $z \in \mathcal{Y}_{\mathcal{N}}$ (the set of ND points) with $\bar{z} \leq u$ [refer to Boland et al. (2017b) for the full algorithm and related proofs].

In the case of the SPRP two IPs must be solved in each iteration of the QSM, where the first IP is a single-objective problem in the third objective of the SPRP, i.e. the total travel time of the tours, and the second IP is a composite function of all the three objectives of the SPRP. This task is intractable to be assigned to an exact solver for even very small sized instances of the SPRP, and thus, the idea is to assign it to LS-based heuristic optimisers instead. However, this is a critical hybridisation and several considerations must be made to be able to replace the exact solver with a heuristic one successfully.

The QSM (or any other MPT for the identification of the set of the ND points) is mainly based on the premise that the solution found at the end of each iteration is an ND solution. However, this could not be guaranteed when the exact solver is replaced with a heuristic one, since as opposed to an exact optimal solution to a single-objective problem, a heuristic near-optimal solution is obtained based on some stochastic operations and each run of the heuristic can lead to (at best) a slightly different solution. Therefore, it is possible that in later iterations of the QSM when the IPs get even tighter constrained, a heuristic solution is found which dominates the previously found and assumed to be ND solutions. This mainly has implications regarding the true progression of the QSM along the efficient frontier, as the solution returned by the `2D-NDP-Search` in each iteration provides the basis for defining the bounds on the first and second objectives in subsequent iterations of the algorithm. When using heuristic optimisers, the solution found in each iteration is only an approximation of the optimal solution, and thus the bounds defined based on it usually underestimate the true bounds and hence some ND points will be missed.

In addition to this, one major difficulty in hybridising heuristics with MPTs arises from the fact that LS-based heuristic algorithms for single-objective optimisation often find it very difficult to retrieve an initial feasible solution when extra objectives are progressively appended to the problem in form of additional constraints, and a large proportion of their computational time is just spent on finding such initial feasible solutions. As the upper bounds to these constraints progressively become tighter, sometimes the algorithm spends the entire allotted computational budget and even fails to find a feasible solution, and therefore only returns *Null*. However, if the very algorithm for the very bounded problem is provided with an initial feasible solution satisfying the assigned bounds, it is usually fast and easy for it to improve the provided solution; what the algorithm is indeed mainly intended to do.

We address these complications by replacing the original list of efficient solutions in the QSM, with the *UENDA* scheme introduced earlier, and implementing techniques for obtaining the required initial feasible solutions from the very archive for each bounded IP. The *UENDA* appears everywhere in the solution process and accompanies all lower-level functions used by the higher-level algorithm to identify and archive all ND solutions encountered during the search. That is, while the first IP optimiser is just focusing on minimising the third objective function, for instance, all newly generated solutions, even those which cannot contribute to the progress of the search towards the minimisation of the objective function of concern, are checked against the *UENDA,* and if non-dominated, are archived. With this scheme, the optimal solution to each IP under consideration is taken from the *UENDA* at the termination of the corresponding IP optimiser, instead of using the optimal solution found by the solver explicitly. Moreover, always prior to launching the first IP solver in the `2D-NDP-Search`, a "best" feasible initial solution, based on the bounds on the first and second objective functions, is selected from the *UENDA* using an `init-sol-select` function. This function sorts the *UENDA* based on the non-decreasing values of the third objective function and starts from the top of the list and picks the first solution that satisfies the upper bounds on the first and second objective functions as the initial solution to the first IP. If no such solution is found, the solution that minimises the aggregated difference between its first and second objective values, and the upper bounds on those objectives is selected as the initial solution. Even though these solutions do not satisfy the bounds on the first two objectives, they are as close as it gets to the bounds. Moreover, the first IP solver has some built-in features that are only activated when the initial solution provided is still unable to satisfy the bounds, and take over the main operators in the first IP optimiser for minimising the third objective, and try to convert the provided infeasible solution into a feasible one first. In case the *UENDA* is empty (usually only at the beginning of the search) an *init-sol-gen* heuristic that is described later is utilised. Observe that the solution to the first IP is used as an initial solution to the second IP. Finally, to define "more conservative" bounds on the first two objective functions for the forthcoming iterations of the algorithm, similar to the approach proposed by Becerra and Coello (2006), the upper bounds derived from the solution returned by the *2D-NDP-Search* can be increased by a confidence tolerance $\mathbb{t}_i, i \in \{1,2\}$.

For solving the first IP in the *2D-NDP-Search* we are proposing a Simulated Annealing (SA) algorithm with a new Exhaustive Neighbourhood Search (ENS), called the *SAENS* algorithm. The second IP, however, is solved using an Order-First-Split-Second Memetic Algorithm (*OFSS-MA*). Therefore, the resulting hybridisation is called the *HQSM-SA-MA* (Figure 3-2).

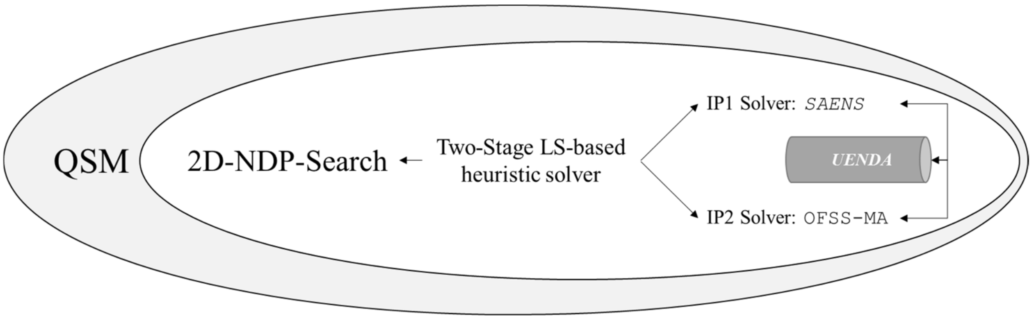


Figure 3-2 Schematic of the proposed *HQSM-SA-MA* algorithm

In the rest of this section, the first and the second IP optimisers are described.

3.4.2.1.1 IP-I optimiser: SAENS

Solving the first IP in the `2D-NDP-Search` is assigned to *SAENS*, which is specifically tailored to minimise the third objective of the SPRP, i.e. travel time of the tours; however, to deal with infeasible initial solutions some extra features are also built in the algorithm to convert an infeasible solution to the bounded SPRP into a feasible one.

A description of the *SAENS* is given in Algorithm 3-3. The initial solution to the algorithm (*init_sol*) is selected using the `init-sol-select` function explained earlier. We remind that the very first feasible solution at the beginning of the *HQSM-SA-MA*, when *UENDA* is still empty, is generated using the `init-sol-gen` heuristic. This simple heuristic initially generates a *Giant Tour (GT)* as a random permutation of all customers, which undergoes the `2-opt` heuristic for distance minimisation. The resulting GT then goes through the `LA-split` based on the third objective of the SPRP, i.e. total makespan, with no RP. Subsequently, the generated set of trips from `LA-split` undergoes the *RTO* algorithm for the minimisation of the third objective.

Common termination criteria for the standard SA are used, and a number of local search iterations are performed in each temperature. One of the special features of the proposed SA is how it explores the neighbourhood using *ENS* strategy (line 7), which will be explained shortly. Along with archiving ND solutions and updating the *UENDA* steadily, the *ENS* outputs both a new solution (*new_sol*) and a usually different current solution (*curr_sol*) from what it takes as input. Lines 8 to 21 of the algorithm are only operated if the current solution is infeasible with regard to the upper bounds (u_1, u_2) assigned on the first and the second objectives by the higher-level `2D-NDP-Search`. If the solution is infeasible with regard to the upper bound assigned on the first objective (line 8), first it is converted into a giant tour (*GT*) using the `Split-inv` function (line 9) which is the inverse of the split function. The resulting *GT* then goes through the `LA-Split` function (line 10), with the following RCs (*RemCon*) and RP (*RemPol*): if an arc in the auxiliary graph has a third objective value less than or equal to 1/3 (or 1/2) of T, then reduce its first objective value to zero (or 1/2 of its actual value). Note that in this study we consider $r = 3$. After this, the `BF-MTO` procedure is called (line 11) to find the optimal MT of the resulting *split_ sol*. If this leads to finding a solution *MT_ sol* with first or third objective values lower than the current solution, then it replaces *curr_ sol* (lines 12 to 14).

In case of the infeasibility of the current solution regarding the upper bound assigned on the second objective (line 16), this solution undergoes a completely different route-path optimisation (line 17) by getting the `Route-path-opt` function to minimise the second objective instead of the third. Observe that in IP-I optimiser, the `Route-path-opt` everywhere else optimises route-path and vehicle type operating the tour by only considering the third objective of the SPRP, and changing the objective of concern can lead to the selection of a completely different sequence of road-paths. Again, if the resulting solution *RP_ sol* has a first or third objective values lower than the current solution, then it replaces *curr_ sol* (lines 18 to 20).

Unlike the standard SA algorithm, in `SAESN` updating the solutions to see if we have a new *best_ sol*, or *curr_ sol* is not a straightforward task, because in order for retrieving a feasible solution as fast as possible, we might sometimes need to let the best solution get degraded, and accept a solution that is worse with regard to the third objective, but feasible with respect to the upper bounds on the first two objectives, as the new best solution. This analysis is assigned to a `sol-update` function (line 22). This function compares the third objective values of the *new_ sol* and the *curr_ sol*, with that of the *best_ sol*, and if any of them is better than the *best_ sol*, both the *best_ sol* and the *curr_ sol* are replaced by the corresponding *curr_ sol* or *new_ sol*. If neither *curr_ sol* nor *new_ sol* can replace *best_ sol* based on this condition, then still if *best_ sol* is infeasible with regard to at least one of the upper bounds assigned on one of the two first objectives, and either the *new_ sol* or the *curr_ sol* is feasible, *best_ sol* is replaced by them. If both are feasible, then the one which has a lower value in the third objective is selected to replace both *best_ sol* and *curr_ sol*. If none of the above conditions are satisfied, then while *best_ sol* remains unchanged, the chance of *new_ sol* replacing *curr_ sol* is investigated using the standard SA thresholds; that is, a *delta* parameter is calculated as follows: *delta = obj3(new_ sol) – obj3(curr_ sol)* (note that *obj3(curr_ sol)* means the cost of *curr_ sol* with regard to the third objective of SPRP). If *delta* ≤ *0*, then *cur_ sol* is replaced with *new_ sol*; otherwise, a random number (*rand*) between zero and one is generated, and if *rand* ≤ *exp(-delta/temp)* then *curr_ sol* is replaced by *new_ sol*.

Algorithm 3-3 SAENS

```
input UENDA, init_sol, $\delta, S, u_1, u_2$, alpha, MinTemp, $MaxIter_{SA}$, $TimeLimit_{SA}$, StartingTemp
best_sol ← init_sol
while time ≤ $TimeLimit_{SA}$ do
    curr_sol ← best_sol, temp ← StartingTemp
    while time ≤ TimeLimitSA and temp > MinTemp do
        for $i = 1$ to $MaxIter_{SA}$ do
            (UENDA, curr_sol, new_sol) ≔ ENS (UENDA, curr_sol, $\delta, S, u_1, u_2$)
            if obj1(curr_sol) > $u_1$ then
                GT ≔ Split-inv (curr_sol)
                (UENDA,split_sol) ≔ LA-Split (OBJ3, GT, RemPol,RemCon, UENDA)
                (UENDA,MT_sol) ≔ BF-MTO (split_sol, UENDA)
                if obj1(MT_sol) ≤ obj1(curr_sol) or obj3(MT_sol) ≤ obj3(curr_sol) then
                    curr_sol ← MT_sol
                end if
            end if
            if obj2(curr_sol) > $u_2$ then
                (RP_sol, UENDA) ≔ Route-path-opt (OBJ3, curr_sol, load(Curr_sol), $e_0$, UENDA)
                if obj1(RP_sol) ≤ obj1(curr_sol) or obj3(RP_sol) ≤ obj3(curr_sol) then
                    curr_sol ← RP_sol
                end if
            end if
            (curr_sol, best_sol) ≔ sol-update (new_sol, curr_sol, best_sol, temp, $u_1, u_2$)
        end for
        temp = alpha × temp
    end while
end while
UENDA ≔ best_eval (Obj3, best_sol, UENDA)
return UENDA
```

After the termination criteria are met and the *best_sol* is returned, a `best-eval` function is called to apply a final touch of improvement on the solution. This function takes a solution to the SPRP as input and provides the best evaluation of the solution by finding the optimal allocation of available resources to it with respect to the objective function of concern. To this end, this function first converts the solution into a GT using the `Split-inv` function and then, if the solution is already feasible with regard to the upper bounds on the other two objectives, applies the `LA-Split` function with no RP; otherwise, the same RP explained above is applied. Then, the route-paths and departure times of the tours in the solution and the vehicles types operating them are optimised using the *RTO* procedure. After this, the solution as it is returned by the `LA-Split` function, also undergoes the `BF-MTO` algorithm to find the optimal MTs. Observe that, as always the *UENDA* is archiving all ND solutions found in the meantime.

Note that, the *best_sol* is not explicitly returned by the SAENS as the solution for the first IP; instead the required solution is extracted from the *UENDA* (line 28). This is because it is possible that a solution with same third objective value as the *best_sol*, but better objective values for the first and the second objectives exist in the *UENDA*, and if no such solution is there, then definitely *best_sol* is in the *UENDA*, since it is non-dominated.

The proposed `ENS` which is described in algorithm 3-4 is inspired by the successful neighbourhood exploration strategy used by Bent and Van Hentenryck (2004) in applying a SA algorithm on the VRPTW. Bent and Van Hentenryck (2004) consider 5 well- known local search operators in their study; i.e. *2-opt*, *Or-opt*, *Relocation*, *Swap*, and *Crossover*. In each iteration of their SA algorithm, they focus on a sub-neighbourhood of a given neighbourhood by randomly choosing a move operator and a customer, and then considering all the possible moves for this customer using the selected operator, to see if any improvement could be found. While we also use the same 5 operators, our `ENS` differs with that of Bent and Van

Hentenryck (2004) in two respects. Firstly, while they only focus on a sub-neighbourhood of a given neighbourhood, in *ENS* the entire neighbourhood is searched exhaustively by exploring all possible moves of all customers based on a randomly selected operator (lines 4 to 18). Furthermore, unlike Bent and Van Hentenryck (2004) we keep updating *curr_ sol* every time the operator is applied. That is, during the *ENS* exploration after each iteration if *new_ sol* is better than *curr_ sol*, then *curr_ sol* is replaced by the *new_ sol* (lines 13 to 17), and in the next iteration the new *curr_ sol* is submitted as the input to the selected operator. While this sounds like an expensive search, it often returns a nicely improved solution that is indeed worth the computational burden and a few iterations of the algorithm usually suffice for finding a near optimal solution. Furthermore, it greatly contributes to the implicit identification of many useful ND solutions, as the entire operation is accompanied by the *UENDA*. We demonstrate the benefits of using the *ENS* in the computational study section of the paper.

Following the termination of the first IP optimiser, the best solution to this IP, with regard to the bounds on the objective functions, is extracted from *UENDA* and submitted to the second IP optimiser that is described next.

Given the description of the two stage solvers, the pseudo-code of the full *HQSM-SA-MA* algorithm is presented in Algorithm 3-5.

Algorithm 3-4 *ENS*

```
input UENDA, curr_ sol, $\delta, S, u_1, u_2$
OP ≔ Random (Operators)
Sltd ≔ a random integer between 1 and $n$
for $i = 1$ to $n$ do
    (UENDA, new_ sol) ≔ OP (UENDA, curr_ sol)
    if RTO threshold for new_ sol is met then
        GT≔ Split-inv (new_ sol)
        (UENDA,split_ sol) ≔ LA-Split (OBJ, GT, No RemPol, No RemCon, UENDA)
        if obj3(split_ sol) ≤ obj3(new_sol) then new_sol ← split_ sol end if
        Call the RTO for new_ sol while updating the UENDA
    end if
    if $i$ = sltd then sltd_ sol ← new_ sol end if
    if obj3(new_ sol) < obj3(curr_ sol) then
        curr_ sol ← new_ sol
    else if (obj1(curr_ sol) > $u_1$ or obj2(curr_sol) > $u_2$) and (obj1(new_sol) ≤ $u_1$ and obj2(new_sol) ≤ $u_2$) then
        curr_ sol ← new_ sol
    end if
end for
new_sol ← sltd_sol
return UENDA, curr_sol, new_sol
```

Algorithm 3-5 HQSM-SA-MA

```
Initialise the $UENDA$ to be empty and a double-ended linked list $D$ with $(+\infty,+\infty)$
while $D$ is not empty do
    $Right\_boundry\_not\_treated \leftarrow True$
    while $Right\_boundry\_not\_treated = True$ do
        Pop the front element of $D$ and denote it by $u$
        $(NDsol, UENDA)$ :=$2D$-$NDP$-$Search$ $(u, UENDA)$   // SAENS and OFSS-MA are used here
        if $NDsol = Null$ then
            $Right\_boundry\_not\_treated = False$
        Else
            if $u_1^f < NDsol_1 + \mathbb{t}_1 - \epsilon_1$ or $D$ is $empty$ then
                Add $(NDsol_1 + \mathbb{t}_1 - \epsilon_1, u_2)$ to the front of $D$
            end if
            Add $(u_1, NDsol_2 + \mathbb{t}_2 - \epsilon_2)$ to the front of $D$
        end if
    end while
    $Top\_boundry\_not\_treated \leftarrow True$
    while $Top\_boundry\_not\_treated = True$ do
        Pop the back element of $D$ and denote it by $u$
        $(NDsol, UENDA)$ :=$2D$-$NDP$-$Search$ $(u, UENDA)$   // SAENS and OFSS-MA are used here
        if $NDsol = Null$ then
            $Top\_boundry\_not\_treated = False$
        Else
            if $u_2^b < NDsol_2 + \mathbb{t}_2 - \epsilon_2$ or $D$ is $empty$ then
                Add $(u_1, NDsol_2 + \mathbb{t}_2 - \epsilon_2)$ to the back of $D$
            end if
            Add $(NDsol_1 + \mathbb{t}_1 - \epsilon_1, u_2)$ to the back of $D$
        end if
    end while
end while
$POS \leftarrow UENDA$
return $POS$
```

3.4.2.1.2 IP-II optimiser: OFSS-MA

The second IP in the *2D-NDP-Search* focuses on minimising a composite scalar objective of all the three objective functions of the SPRP and this is assigned to the *OFSS-MA* that uses a slightly modified version of the *SAENS* in its education and intensification phases. Fleet size and mix, and multi-trip optimisation contribute significantly to the minimisation of this objective and a modified version of the *best-eval* function described earlier plays a key role in this regard. As before, the *UENDA* appears in all levels of the algorithm and stores all ND solutions encountered during the search. The overall framework of the proposed *OFSS-MA* is shown in the flowchart in Figure 3-3.

As it can be seen in this figure, after the initialisation phase, until the stopping criterion of the algorithm, i.e. a maximum number of iterations (*MaxIter*), is not met, the four phases of parent selection and crossover, education, intensification, and survivor selection are carried out on each generation to return the best feasible solution to the second IP in the *2D-NDP-Search*. These steps are explained in more detail in the sequel.

To initialise the algorithm and generate the first population, we first refer to the *UENDA*. All solutions in the *UENDA* are sorted based on their third objective values. If the number of solutions in the *UENDA* is larger than n_p, then the first n_p solutions from the top of the *UENDA* are selected to be included in the initial population. This provides a nice kick-start for the second solver, as solutions in the *UENDA* are all of a high quality. Otherwise, if n_p is larger than the number of solutions in the *UENDA*, all solutions in the *UENDA* are included in the population and then the remaining required solutions are generated using an extension of the *init-sol-gen* heuristic with the composite objective function.

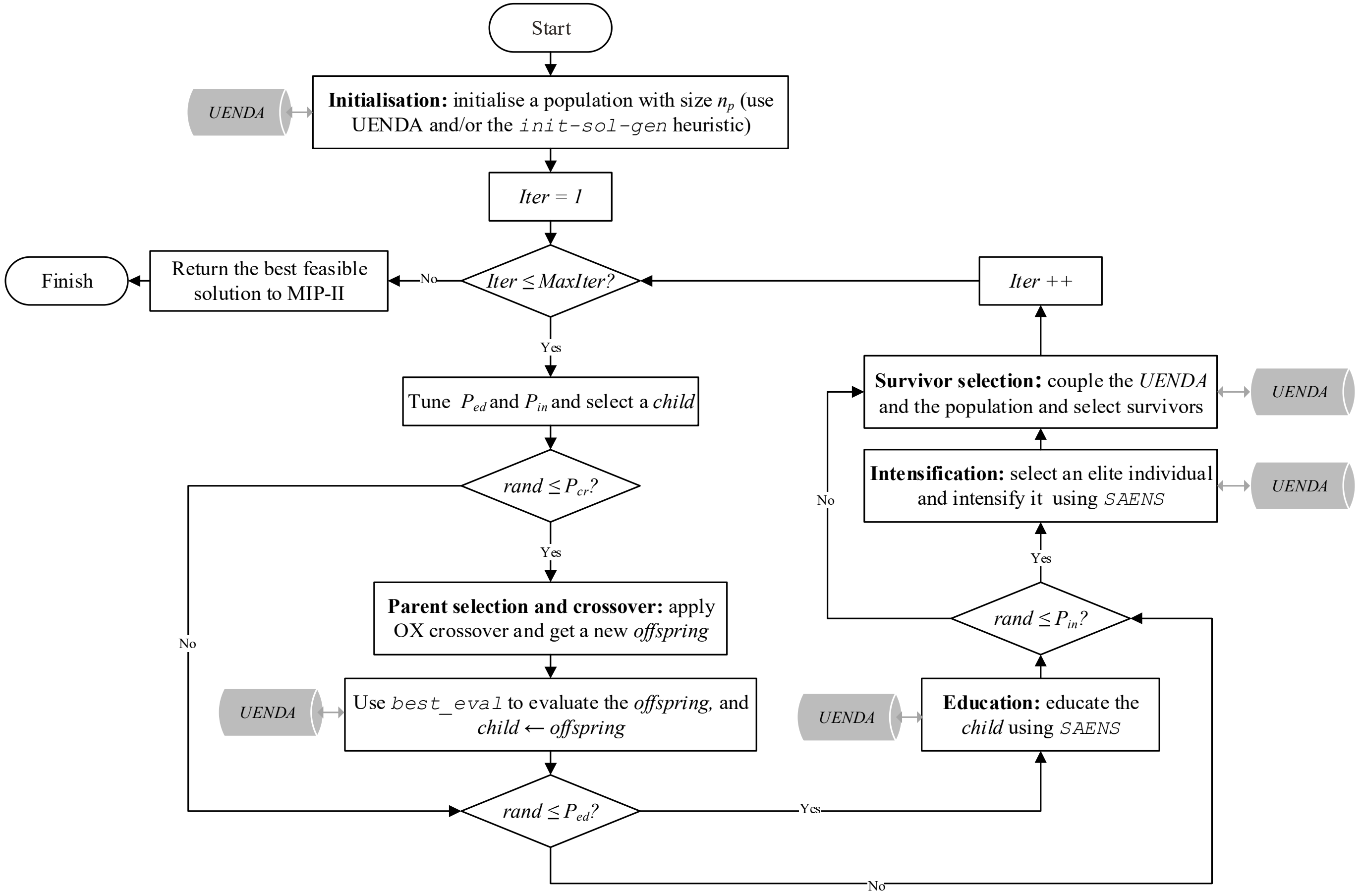

Start
UENDA
Initialisation: initialise a population with size n_p (use UENDA and/or the init-sol-gen heuristic)
Iter = 1
Finish
Return the best feasible solution to MIP-II
No
Iter ≤ MaxIter?
Iter ++
Yes
Survivor selection: couple the UENDA and the population and select survivors
UENDA
Tune P_{ed} and P_{in} and select a child
Intensification: select an elite individual and intensify it using SAENS
UENDA
rand ≤ P_{cr}?
No
Yes
Yes
Parent selection and crossover: apply OX crossover and get a new offspring
rand ≤ P_{in}?
No
UENDA
Use best_eval to evaluate the offspring, and child ← offspring
UENDA
Education: educate the child using SAENS
rand ≤ P_{ed}?
Yes
No

Figure 3-3 *OFSS-MA* for MIP-II in *2D-NDP-Search*

One solution from the initial population is always randomly selected as a *child* in the beginning of each iteration of the algorithm to ensure that if crossover is not taking place, there is a selected child in the population to undergo education if required. Subsequently, a random number is generated and if it is smaller than P_{cr} (crossover probability), parents are selected using the binary tournament method for crossover; i.e. two chromosomes are randomly selected from the population and the least-cost one becomes the first parent, and the same procedure is repeated to get the second parent. These chromosomes are solutions from the population that are converted into giant tours without trip delimiters using the `Split-inv` function. The classical OX crossover is used to generate two new offsprings; however, only one of them is randomly selected in our algorithm. In the OX crossover, two positions are randomly selected in the first parent and the substring between the selected positions is copied into the first offspring, at the same positions. The second parent is then swept cyclically from the second position onwards to fill the empty positions in the offspring. The second offspring is generated likewise by exchanging the roles of the two parents. The selected offspring is then evaluated using a modified `best-eval` function which changes the given giant tour into a high quality SPRP solution. To this end, the given chromosome is first split into a feasible set of vehicle trips using the `LA-Split` function with the same RC and RP described in the previous section. The route-path and departure times of the trips in the deduced solution are then optimised using the `TB-RTO` procedure and the corresponding cost is recorded. Then, the `BF-MTO` procedure is also called and the optimal MT scenario is found and compared with the solution found by the `TB-RTO` (without MTs) and the best solution is accepted as the performance of the chromosome. Observe that all lower-level optimisation problems are now carried out as single objective optimisation problems, which minimise the composite scalar objective of all the three objective functions of the

SPRP, and throughout the entire process all encountered ND solutions are archived in the *UENDA*.

The education and the intensification phases are both based on the `SAENS` algorithm, with the difference that the threshold verified block in the `ENS` (lines 7 to 10 in Algorithm 3-4) is replaced with the modified `best-eval` function described above. In the intensification phase, an elite solution is randomly selected from the top 1/3 of the population and is intensified using the SAENS algorithm. Notice that, at each iteration of the algorithm, education and intensification are applied on a candidate chromosome (an offspring or an elite solution) with probabilities P_{ed} and P_{in}, respectively. However, in certain circumstances these probabilities are tuned adaptively from inside the algorithm. That is, when the algorithm finds no feasible solutions in the population with regard to the upper bounds assigned on the objectives, it increases these probabilities (especially P_{in}) to benefit from the built-in features in the `SAENS` algorithm that are designed for changing an infeasible solution into a feasible one (see the previous section). Once a feasible solution is found, these probabilities retrieve their original values. Remember that, unlike when an exact solver is used in the QSM, there are two sets of bounds on the objectives for the second IP in the heuristic `2D-NDP-Search`; firstly, the values of each of the three objective functions should not exceed the corresponding values already found by the first IP, and secondly, the upper bounds, u_1 and u_2 which are not essentially satisfied by the solution to the first IP should be observed. We remind that when an exact solver is used, once these bounds are not satisfied by the solution of the first IP, it is guaranteed that the second IP is also infeasible. However, in a heuristic case this could be attributed to the inability of the first solver in finding a feasible solution to the first IP, and chances are that the second IP solver can find a feasible solution.

All offsprings and new solutions found through the crossover, education and the intensification phases are added to the population, and at the end of each iteration of the algorithm, in the survivor selection step, a coupling function puts all the solutions in the population and the *UENDA* into a same pool of solutions. The pool is then sought for feasible solutions with regard to the upper bounds on the objective functions, and these feasible solutions are sorted in the non-decreasing order of their composite scalar objective value. Infeasible solutions are then sorted after this set based on the non-decreasing order of their third objective value. After this the top n_p solutions in this sorted pool are selected as the surviving generation.

3.4.2.2 MOOH II: A Hybrid Multi-Objective Evolutionary Algorithm with a Generational-Target Attainment (HMOEA-GTA)

MOEAs are known to be very well-performing in solving multi-objective optimisation problems. They incorporate the concept of Pareto optimality to evolve a generation of solutions at multiple points along the Pareto front, without the need of linearly combining multiple objectives into a composite scalar objective function (Tan et al., 2006), and can generate several elements of the Pareto optimal set in a single run. The hybridisation of global search MOEAs with LS techniques, known as hybrid or memetic MOEAs, makes them more beneficial to real-world applications by driving the search towards the Pareto front more effectively and efficiently (Coello et al., 2007). The LS process in the decision space, and the selection of associated objective space points to explore and exploit constitutes the heart of the hybridised approach (Coello et al., 2007). In order to select points in objective space based upon LS in decision space, hybrid MOEA techniques are usually based on either: (i) only a single objective, (ii) the weighted vector methods, or (iii) the dominance methods (Coello et al., 2007). Our approach exploits somehow the advantages of all these three methods.

The proposed `HMOEA-GTA` solution methodology has a general structure similar to that of the `OFSS-MA` we proposed for solving IP-II in the `2D-NDP-Search` in the previous section; that is, we have the same steps of initialisation, parent selection and crossover, education, intensification, and survivor selection, and we recycle the *SAENS* algorithm as the LS in the education and intensification stages. However, here in the *ENS* used within the *SAENS* we relax the requirement that all customers should go through the randomly selected operator, and instead only five customers are randomly selected for this purpose. This way by using a faster neighbourhood search strategy, we are able to carry out sufficient EA iterations.

Similar to the case of the *HQSM-SA-MA*, the *UENDA* scheme plays a crucial role in the proposed *HMOEA-GTA* and appears literally everywhere in the algorithm. The main difference, however, is clearly in the fitness assignment schemes used in the global search MOEA and within the LS technique. In the sequel we focus on the main differences of the *HMOEA-GTA* algorithm with *OFSS-MA*; hence, all other aspects of the algorithm that are not given a detailed mention here are similar to the *OFSS-MA*.

The *HMOEA-GTA* uses the ND sorting criterion of Deb et al. (2002) in its global search MOEA. Based on this ND sorting criterion the population is divided into ND fronts and all individuals on the same front are given a similar fitness value (normally a rank), such that the lower is the front, the fitter is the solution. In order to carry out fitness assignment inside the LS, a Generational Target Attainment (*GTA*) scheme (along with the *UENDA*) is proposed. The *GTA* scheme works based on *ideal* and *nadir* values selected for each objective function in each generation as the algorithm evolves, where ideal values are considered as "targets to attain". Starting from the initial population the minimum value found for each objective times π, where $0 \leq \pi \leq 1$, is selected as the current ideal value for that objective ($f_k^+, k \in \{1,2,3\}$), and the maximum value is selected as the current nadir value ($f_k^-, k \in \{1,2,3\}$); then during the local search the fitness value of any newly generated solution (f^{new}) is calculated as its normalised closeness to the ideal solution as follows:

$$f^{new} = \sum_{k=1}^{3} \frac{f_k^{new} - f_k^{+}}{f_k^{-} - f_k^{+}} \tag{3-4}$$

This fitness assignment scheme is also used by the global search MOEA for breaking ties whenever in the parent selection and crossover step fitness sharing occurs and two candidate chromosomes have a same rank.

The main benefit of using the proposed *GTA* scheme is that it reinforces the exploitation ability of the algorithm and its convergence towards the true front by pursuing a target-oriented search. Moreover, compared with most scalarisation and weighted vector methods which are usually very sensitive to the weights assigned to the objectives, expression (3-4) that is used within *GTA*, assigns a normalised fitness value to each solution and does not require the determination of weights. It is worth reiterating that in our algorithm the LS is always accompanied by the *UENDA* which archives all ND solutions encountered during the search, and this partially resembles the role played by the dominance methods.

To ensure that the exploration ability of the algorithm is not affected significantly by the *GTA* scheme, a new approach for assigning objective functions to lower-level optimisation problems within LS is employed. Unlike in the *OFSS-MA* that we always assign a composite scalar objective of all the three objective functions of the SPRP to the lower-level problems, in the *HMOEA-GTA* three candidate objective functions: (i) fuel consumption, (ii) travel time, and (iii) the scalarisation of vehicle cost, fuel consumption and travel time, are put in a pocket with equal chances to be picked, and one of them is randomly selected based on roulette wheel selection for each lower-level optimisation problem. This approach has turned out to be contributing to the exploration ability of the proposed *HMOEA-GTA*, and in experimentations with different alternative schemes we have observed that it consistently leads to a much better performance than selecting an objective function deterministically.

In order to initialise the algorithm and generate the first population, the proposed *HMOEA-GTA* algorithm uses the *init-sol-gen* heuristic described earlier. Similar to the case of the proposed `OFSS-MA` in the previous section, at crossover, parents are converted into chromosomes without trip delimiters using the `Split-inv` function, and offsprings are evaluated using the previously described `best-eval` function, but based on the *GTA* fitness scheme described above.

Throughout the entire process of the `HMOEA-GTA`, all encountered ND solutions are archived in the *UENDA* and this set acts independently from the HMOEA population. At the end of each iteration of the algorithm, in the survivor selection step, a coupling function puts all the solutions in the population and the *UENDA* into a same set, and applies the ND sorting on the entire solutions in the set and selects the n_p solutions of the best ranks. Ties are broken based on the *GTA* fitness, and if any solution of rank one is left outside the survived population, it is reinserted back into the *UENDA*; otherwise, the *UENDA* is empty at the beginning of the next iteration.

3.4.2.3 MOOH III: A Simple Order-First-Split-Second MOEA (SOFSS-MOEA)

The main motivation behind introducing the `SOFSS-MOEA` is in fact to deal with a possible limitation of the `HMOEA-GTA` in terms of its computational cost. Indeed, due to the intensification and the education phases of the algorithm that are carried out by the `SAENS`, if one intends to exploit fully the generational evolution ability of MOEAs by conducting a sufficient number of EA iterations, the overall runtime of the algorithm is inevitably increased. Instead, as the third alternative MOOH to solve the SPRP we present here the `SOFSS-MOEA` which is much simpler to implement and can carry out a large number of MOEA iterations in a reasonable computational time.

The general steps of the proposed `SOFSS-MOEA` comprise population initialization, ND sorting of the individuals in the population based on the ND sorting criterion of Deb et al. (2002), parent selection, crossover and mutation, and recombination and survivor selection. Like the previous MOOHs, this algorithm is also accompanied by the *UENDA* scheme. The steps of `SOFSS-MOEA` are summarised in the sequel.

To generate the first population of solutions in the `SOFSS-MOEA` a simple heuristic called the `MOEA_Init_Gen`, with a structure similar to the `init-sol-gen` heuristic is used. In the first step of the `MOEA_Init_Gen` n_p (population size) GTs are generated as random permutation of all customers, where 2/3 of them undergo the 2-opt heuristic for distance minimisation. The resulting population of the GTs from the first step then goes through a 3-phase best evaluation function called the `3P_best_eval`. This function first splits a GT based on the first objective of the SPRP, i.e. vehicle hiring cost, using the `LA-split` function with the following RC and PP: if an arc in the auxiliary graph has a third objective value less than or equal to 1/3 (or 1/2) of T, then reduce its first objective value to zero (or 1/2 of its actual value). The resulting set of trips from `LA-split` then undergoes the `RTO` algorithm for the minimisation of the first objective, following which, the solution goes through the `BF-MTO` procedure. The attributes of this solution are stored and the algorithm proceeds to phases two and three, where the same steps are repeated based on the second and third objective functions of the SPRP, respectively, with the difference that no RC and RP is used for the `LA-split`. Note that throughout these steps and lower-level heuristics all ND solutions are identified and stored in the *UENDA*. Following the completion of the three phases in the `3P_best_eval`, the solutions from the phases are compared based on their composite scalar objective of all the three objective functions of the SPRP and the one that has the minimum of all the three is selected as the best performance of the GT under consideration.

Parent selection and crossover in the `SOFSS-MOEA` are done the same way as in the `HMOEA-GTA`, with the difference that the `3P_best_eval` is used whenever evaluation is required.

In the mutation step of the `SOFSS-MOEA`, three fast and simple heuristics are employed to improve a given solution. The first heuristic is based on the well-known `2-opt` heuristic for the TSP. Every route in a given solution (*Sol*) undergoes a complete `2-opt` (i.e. for all customers on the route). The resulting solution is checked in the optimal departure times of the tours and if better than the given *Sol*, replaces it; otherwise, the same *Sol* is returned. The second heuristic is called the `route-elim` heuristic, which is a routes elimination and reinsertion heuristic with two competing insertion schemes. In this heuristic, all vehicle tours in the *Sol* that have a smaller size (i.e. the number of customers served by the vehicle) than a decided parameter *minSize*, are eliminated from the solution, and all customers on these eliminated routes are put into a non-routed pocket and then are attempted to be reinserted into the remaining routes. If they could not be served by any of the routes, a new route is initiated to accommodate them. Two insertion algorithms compete for re-inserting the customers back into the routes. One of them is based on the best insertion point that positions the customer in its best position of all tours, and the second one is a new 2-opt re-insertion algorithm that adds the customer to the end of each route, and then applies 2-opt on the customer to find the best sequence of visits for the route when the new customer is added, and then this is repeated for all routes and the best insertion is returned. The performance of the two re-insertion algorithms is checked against one another and the one with the lowest cost is accepted. The quality of the proposed `route-elim` algorithm is sensitive to the value decided for the *minSize* parameter, and without user interference, the best value is selected regarding the characteristics of the instance under consideration, by changing this value in a well-defined range. Note that again new ND solutions might be encountered which are archived in the *UENDA*.

The last heuristic in the mutation phase, is the `node-elim` heuristic. This algorithm begins from the first customer in the first route, and ejects the customer from its current position and tries to reinsert it in its best position in any of the routes using the best reinsertion algorithm. The algorithm continues like this for all the subsequent customers and only terminates when no improvement is gained after n consequent ejections.

Finally, in the recombination and survivor selection step of the algorithm, the same coupling function used within the *HMOEA-GTA* is used. The *SOFSS-MOEA* has a much lighter mutation step than that of the *HMOEA-GTA*, and thus it is expected to be more agile.

3.5 Computational study

To analyse the performance of the proposed MOOHs experiments are carried out on three sets of test problems. The first set of experiments are based on a set of 25 small sized test instances specifically designed as the benchmark for evaluating the performance of MOOHs for the SPRP by Raeesi and Zografos (2019). These instances are defined on networks with 100 nodes out of which five nodes are selected as customer nodes. Raeesi and Zografos (2019) generate the full set of the ND solutions to these instances using a method based on the QSM that uses a PEP-based MILP formulation of the SPRP as the core optimisation problem. The solutions to these instances provide a basis for comparing the approximate fronts generated by the proposed MOOHs with the true fronts.

The second set of test instances are based on the publicly available network of Chicago's arterial streets (https://data.cityofchicago.org) with time-dependent speed observations (Figure 3-4). We use the graph and the traffic updates provided by Dokka and Goerigk (2017) based on this database. We have considered a planning horizon of 8 hours from 08:00 to 16:00 (i.e. 480 minutes) in their considered day, and generated a set of 10 test instances by randomly selecting 100 nodes of the graph as customer nodes. In all test instances, service times and the reloading time at the depot for vehicles executing an extra round of trip are assumed 20 minutes. Feasible time-windows and demands are induced for the customers using a procedure based on a nearest neighbour algorithm where a heavy-duty truck is dispatched to visit the nearest customer in each iteration of the algorithm, until capacity or time constraints are violated. Customers' demands are drawn randomly from the discrete uniform distribution on the interval [1000kg, 15000kg], and relatively wide time-windows covering up to 40% of the planning horizon are generated around the arrival time of the dispatched trucks. For all instances, synthetic driving cycles denoting the second-by-second speed variations and thus A/D rates are constructed from the available macroscopic traffic speed records using the model proposed in Raeesi and Zografos (2019), and corresponding RTM and UTM values for road-links in the network are computed and stored.

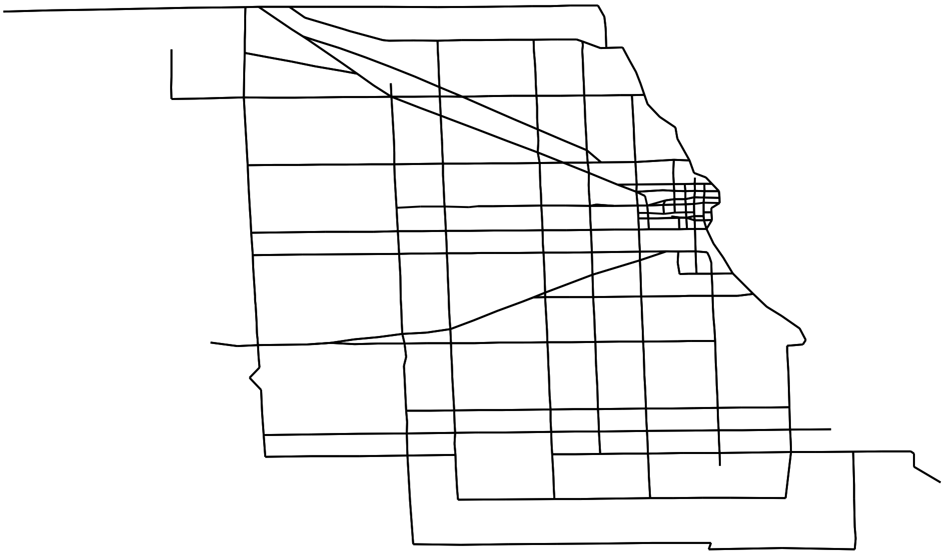


Figure 3-4 Chicago's arterial streets

The third set of test instances are large sized test instances of size 500 and 1000 where 10% of the nodes (i.e. 50 and 100 nodes) are selected as customer nodes. The set of the time-dependent road networks generated in these instances resemble real-life urban road networks and the way the time-dependent traffic congestion occurs in them during a given day. The desired raw graphs of these instances are created using the procedure proposed by Letchford et al. (2013). The centre of the generated road networks is assumed to be representing a city centre, which is usually the most congested part of an urban road network during rush hour, and as we get distant from the centre, traffic congestion gradually softens. Based on this assumption, each road-link in the network is classified as an inner, intermediate, or outer road, such that every road segment that has an end-point within 40% of the radius of the generated graphs is categorised as an inner road, all road segments with an endpoint between 40% and 70% of the radius are categorised as intermediate roads, and the remaining roads (farther than 70% of the radius) are categorised as outer roads. This classification of roads is essential for defining realistic time-dependent traffic congestion scenarios. In Figure 3-5, an example time-dependent road network with 1000 nodes where 100 of these are customers is illustrated. In this figure road segments in red, yellow, and green are of classes inner, intermediate and outer roads, respectively. Green nodes represent customer locations on the graph, and yellow nodes are network junctions.

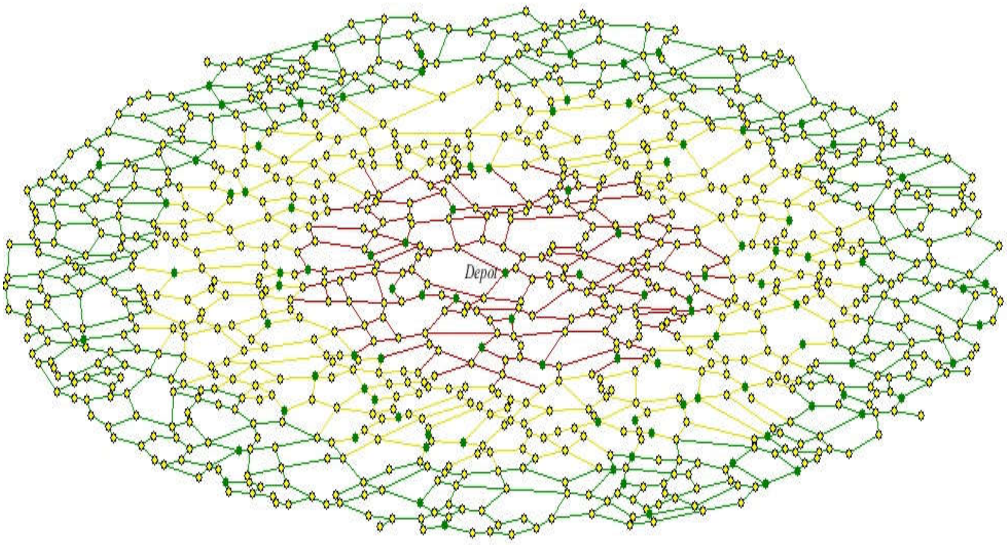


Figure 3-5 An example time-dependent road network with 1000 nodes

Determination of the time-dependent traffic congestion and its spread over the arcs over time has been carried out by incorporating observations from real world traffic patterns. One such observation is that peak hours do not occur to all road segments in the network at the same time. For example, early in the morning inner roads in major cities experience less traffic congestion compared to other road categories as many people are commuting from outer roads towards the intermediate ones and then towards the inner roads. Based on this, for each road category a certain traffic pattern during the day is defined, and 15-minute data series of average speed observations are generated randomly within the designated patterns. Figure 3-6 shows one of such randomly generated speed levels within the defined traffic patterns for a given instance. As it can be seen, in the beginning of the working day outer roads have the slowest congestion speed, but later in the day they constitute the fastest road-links in the road network, while inner roads are the most congested roads from around 09:00 a.m. onward with a morning traffic peak at around 10:00, and an afternoon one at around 15:00.

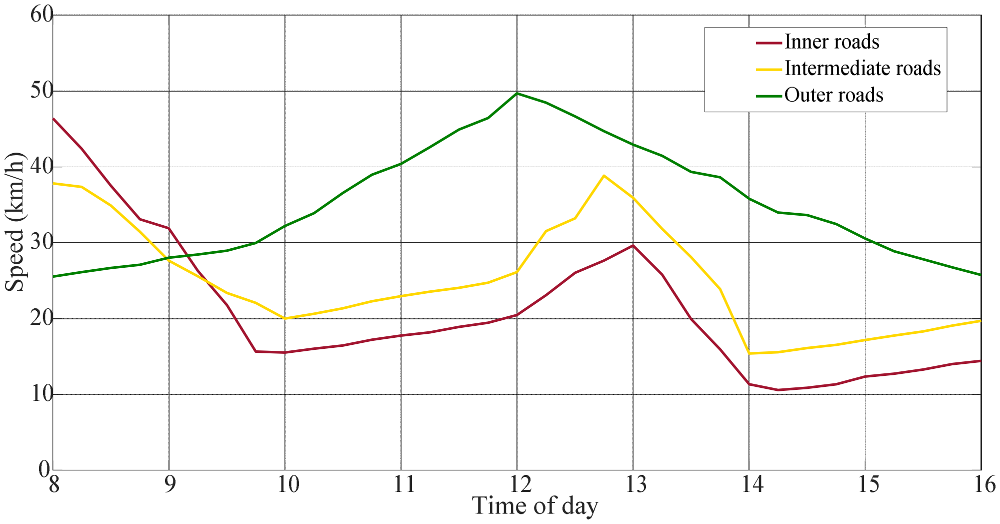


Figure 3-6 An example of randomly generated speed profiles for different road categories

Similar to the case of the Chicago instances, a time-horizon equal to 8 hours is used, and service times of 20 minutes are assumed for all customers. Driving cycles are also constructed for all road-links in these networks as explained earlier. In total 15 instances of size 500 with 50 customers, and 15 instances of size 1000 with 100 customers are generated. Note that while the problem features considered in this paper are considerably more complicated and comprehensive, the problem size of the instances compare favourably with the existing related studies. All the test instances developed in this paper along with the reported solutions in this section are available at http://www.lancaster.ac.uk/staff/raeesi/MOOHsforSPRP-DAT.zip, or could be requested from the authors via email.

In a pre-processing stage, the PEP is applied on all the 65 test instances considered by the paper and the resulting reduced networks are submitted to the proposed MOOHs for the generation of Pareto fronts.

All algorithms were implemented in MATLAB and run on a computer with Intel Core™ i5 3.20 GHz processor with 8 GB RAM. The description and the determined values for the parameters used in the algorithms are provided in Table 3-2. To fine-tune the parameters on all categories of instances considered, we have conducted a preliminary set of empirical analyses. While the chosen settings generally work well on the considered test problems, no claim is made that our choice of parameter values is the best possible.

Not that the total time budget of all algorithms for all instance sizes is set to 45 minutes, and all algorithms are run 10 times on each instance and all reported measures and runtimes are based on the average of the 10 runs.

Table 3-2
Description and the determined values for parameters in each algorithm

Algorithm	Parametres	Description	Values*
HQSM-SA-MA	ϵ_1	QSM parameter	1
	ϵ_2	QSM parameter	0.1
	$TimeLimit_{SA}$	Time budget given to SAENS for IP-1 (second)	5
	$MaxIter_{SA}$	Maximum number of allowed iterations in SAENS for IP-1	10
	$MaxIter_{MA}$	Maximum number of allowed iterations in OFSS-MA	5
	n_p	Population size in OFSS-MA	30
	$TimeLimit_{ed}$	Time budget given to SAENS for education in OFSS-MA (second)	2
	$MaxIter_{ed}$	Maximum number of allowed iterations in SAENS for education in OFSS-MA	3
	$TimeLimit_{in}$	Time budget given to SAENS for intensification in OFSS-MA (second)	2
	$MaxIter_{in}$	Maximum number of allowed iterations in SAENS for intensification in OFSS-MA	3
HMOEA-GTA	n_p	Population size	50
	$MaxIter_{HMOEA}$	Maximum number of allowed iterations in HMOEA-GTA	1000
	$TimeLimit_{ed}$	Time budget given to SAENS for education (second)	5
	$TimeLimit_{in}$	Time budget given to SAENS for intensification (second)	5
	$MaxIter_{ed}$	Maximum number of allowed iterations in SAENS for education	3
	$MaxIter_{in}$	Maximum number of allowed iterations in SAENS for intensification in OFSS-MA	3
SOFSS-MOEA	n_p	Population size	50
	$MaxIter_{SOFSS\text{-}MOEA}$	Maximum number of allowed iterations in *SOFSS-MOEA*	3000

Common parameters for all algorithms and instance sizes: P_{ed} = 0.5, P_{in} = 0.5, P_{cr} = 0.9, π = 0.75, δ = 0.95, *S=10, alpha = 0.92, MinTemp = 300, StartingTemp = 500*. Total time budget of all algorithms for all instance sizes is set to 45 minutes.

Unlike single objective heuristics, evaluating the performance of a MOOH is not a straightforward task, since approximated sets of ND solutions are yielded by MOOHs which are not easy to compare. Therefore, a set of performance metrics tailored for the evaluation of the quality of MOOHs are usually used. These metrics are described next.

3.5.1 Performance metrics

We use the following multi-objective performance metrics that measure the two criteria of convergence and uniform diversity (Durillo & Nebro, 2011):

- *Generational Distance (GD)* (Van Veldhuizen & Lamont, 1998). This performance metric measurs how far the solutions in the Pareto appoximation are from those in the optimal Pareto front and is defined as follows:

$$GD = \frac{\sqrt{\sum_{i=1}^{\mathcal{s}} d_i^2}}{\mathcal{s}} \tag{3-5}$$

  where $\mathcal{s}$ is the number of the solutions in the approximation and d_i is the Euclidean distance between each solution and the nearest member in the optimal Pareto front in the objective space. Clearly, if all the solutions in the approximated front are in the true front, then we have $GD = 0$.
- *Inverse Generational Distance (IGD)* (Zitzler et al., 2000). As a variant of GD, IGD is aslo computed using the same expression (3-5), with the difference that $\mathcal{s}$ in IGD represents the number of solutions in the true front and d_i is the Euclidean distance between each point of that true front and the nearest member in the approximation.
- *Hypervolume (HV)* (Zitzler & Thiele, 1999)*:* This metric calculates the volume enclosed by the set of ND points $\mathcal{S}$ in the objective space, such that for each point $i \in \mathcal{S}$, a hyper-cube v_i is computed with a reference point $\mathcal{W}$ and the point i as the diagonal corners of the hypercube. The reference

point can be found by constructing a vector of maximal objective functions values. A union of all hypercubes is determined as the hypervolume (*HV*):

$$HV = volume\left(\bigcup_{i=1}^{|S|} \boldsymbol{v}_i\right) \tag{3-6}$$

A larger value of *HV* is more desirable.

The calculation of these performance measures, however, requires a knowledge of the true front, which is not available in case of large sized instances. In such cases, following a common practice in the pertinent literature (Coello et al., 2007; Fonseca et al., 2005), we construct a *'Reference Set' (RS)* by putting all Pareto fronts found by the MOOHs over all runs in a pool of solutions, and then extracting all ND solutions from this pool. This set constitutes the basis of comparison for all proposed algorithms. Having RSs available, as another useful performance indicator, we use the HV% metric that represents the ratio of the HV of each algorithm to the HV of the RS in the corresponding instance.

3.5.2 Performance of the MOOHs

In this section, we begin by analysing the performance and the scalability of the proposed algorithms in solving large sized instances based on the Chicago road network, and the graphs with 500 and 1000 nodes with 50 and 100 customer locations, respectively. The summary performance of the proposed MOOHs against the true POSs of the small sized instances is presented afterwards.

Table 3-3 to Table 3-5 present the average runtime, and the average performance of the MOOHs under the performance metrics of GD, IGD, HV and HV%, for each instance of size $|N| = 500, |N_1| = 50$ (called $I_{500,50}$ instances hereafter), $|N| = 1000, |N_1| = 100$ (called $I_{1000,100}$ instances hereafter), and Chicago test instances with $|N_1| = 100$ (called $I_{chicago,100}$ instances hereafter), respectively. The HV% in these tables represents the ratio of the HV of each method to the HV of the RS in the corresponding instance.

It is observed in these tables that the *HQSM-SA-MA* is consistently able to provide a very good approximation of the RSs in all cases. However, the performance of the *HMOEA-GTA* degrades significantly with the increasing size of instances. The *SOFSS-MOEA*, on the other hand, has a rather satisfactory performance in case of the $I_{500,50}$ instances, and the $I_{chicago,100}$ instances, but its performance in case of the $I_{1000,100}$ instances is not as good. Note also that while the *HMOEA-GTA* uses the entire allocated computational budget in all cases (i.e. 45 minutes), the other two algorithms only consume the whole budget in case of the larger instances.

In Table 3-6, we summarise all results obtained from the application of the three proposed MOOHs on all test instances of the paper. The las two columns of this table, i.e. Avg. All and Avg. Large, show the average performance over all test instances and over large sized test instances, respectively.

It is obvious from the table that the proposed MPT-based algorithm, i.e. *HQSM-SA-MA*, is placed on top of the other two MOEA-based algorithms in almost all multi-objective performance metrics over the considered libraries of instances and can provide a consistently good solution quality while preserving the required scalability. The *HMOEA-GTA*, on the other hand, might be considered as the least favourable algorithm to solve the SPRP; while it can deliver almost the best performance in the case of the small sized instances (at a higher computational cost, though), its ability to deal with large sized SPRP instances is much degraded by the increasing size and difficulty of these instances. On the contrary, while the performance of the *SOFSS-MOEA* is not very promising in case of the small sized-test instances, it can deliver a much better performance than the *HMOEA-GTA* (but not as good as the *HQSM-SA-MA*) when the instance sizes increase.

Table 3-3

Multi-objective performance measures for $I_{500,50}$ test instances

Inst. #	*HQSM-SA-MA*					*HMOEA-GTA*					*SOFSS-MOEA*				
	Time m	GD	IGD	HV	HV%	Time m	GD	IGD	HV	HV	Time m	GD	IGD	HV	HV
1	31	0.02	0.02	0.78	86.21%	45	0.03	0.03	0.58	63.75%	18	0.03	0.03	0.66	72.78%
2	23	0.02	0.05	0.71	79.75%	45	0.05	0.06	0.49	54.59%	22	0.03	0.03	0.63	70.74%
3	12	0.04	0.03	0.56	69.40%	45	0.03	0.03	0.49	60.13%	21	0.02	0.03	0.68	84.02%
4	30	0.02	0.02	0.65	72.09%	45	0.04	0.03	0.52	57.13%	20	0.08	0.03	0.45	50.39%
5	23	0.03	0.04	0.55	66.40%	45	0.04	0.04	0.40	48.47%	20	0.02	0.03	0.51	61.03%
6	29	0.03	0.03	0.59	66.15%	45	0.03	0.03	0.52	58.08%	19	0.07	0.03	0.39	43.95%
7	11	0.03	0.02	0.82	86.14%	45	0.03	0.04	0.54	56.50%	21	0.13	0.04	0.52	54.81%
8	10	0.02	0.04	0.61	74.67%	45	0.03	0.03	0.42	51.25%	22	0.04	0.03	0.60	73.38%
9	38	0.01	0.01	0.84	90.34%	45	0.04	0.03	0.50	54.18%	21	0.09	0.06	0.26	27.55%
10	29	0.03	0.03	0.56	66.55%	45	0.03	0.03	0.53	63.19%	17	0.02	0.03	0.54	64.08%
11	37	0.01	0.01	0.83	88.35%	45	0.03	0.02	0.60	63.84%	23	0.07	0.04	0.38	40.67%
12	15	0.03	0.05	0.61	72.57%	45	0.04	0.05	0.48	57.18%	14	0.04	0.04	0.70	82.84%
13	30	0.02	0.03	0.73	82.88%	45	0.04	0.04	0.47	53.63%	18	0.05	0.04	0.51	58.32%
14	20	0.01	0.03	0.64	74.02%	45	0.06	0.06	0.30	34.77%	15	0.05	0.03	0.62	71.66%
15	30	0.02	0.03	0.74	82.95%	45	0.05	0.04	0.46	51.02%	16	0.05	0.03	0.62	69.57%
Avg.	24	0.02	0.03	0.68	77.23%	45	0.04	0.04	0.49	55.18%	19	0.05	0.03	0.54	61.72%

Table 3-4

Multi-objective performance measures for $I_{1000,100}$ test instances

Inst. #	*HQSM-SA-MA*					*HMOEA-GTA*					*SOFSS-MOEA*				
	Time m	GD	IGD	HV	HV%	Time m	GD	IGD	HV	HV	Time m	GD	IGD	HV	HV
1	46	0.03	0.04	0.73	74.44%	45	0.12	0.17	0.17	17.72%	34	0.11	0.10	0.41	41.34%
2	46	0.01	0.02	0.76	80.07%	45	0.09	0.07	0.25	26.33%	43	0.11	0.03	0.55	57.84%
3	46	0.03	0.05	0.49	51.74%	45	0.09	0.10	0.26	27.82%	30	0.09	0.05	0.55	58.22%
4	46	0.02	0.03	0.56	65.75%	45	0.05	0.04	0.38	44.25%	35	0.02	0.03	0.69	79.99%
5	46	0.02	0.04	0.83	87.68%	45	0.04	0.09	0.33	34.93%	42	0.10	0.10	0.32	33.64%
6	46	0.02	0.02	0.74	77.36%	45	0.12	0.10	0.22	22.51%	45	0.10	0.05	0.47	48.53%
7	46	0.03	0.04	0.76	76.53%	45	0.11	0.18	0.19	18.75%	36	0.06	0.06	0.66	66.19%
8	45	0.02	0.02	0.75	78.53%	45	0.11	0.11	0.22	22.67%	45	0.08	0.05	0.58	60.83%
9	46	0.01	0.02	0.81	87.80%	45	0.11	0.10	0.12	13.45%	35	0.04	0.03	0.57	62.13%
10	45	0.02	0.04	0.70	76.74%	45	0.05	0.04	0.36	39.03%	45	0.12	0.05	0.37	40.91%
11	45	0.02	0.02	0.77	79.69%	45	0.08	0.06	0.35	36.15%	45	0.13	0.06	0.32	33.10%
12	44	0.02	0.01	0.78	82.99%	45	0.10	0.08	0.23	24.76%	45	0.10	0.04	0.45	47.36%
13	45	0.01	0.04	0.81	87.20%	45	0.04	0.07	0.25	27.39%	34	0.07	0.04	0.63	67.82%
14	37	0.02	0.02	0.77	79.89%	45	0.12	0.11	0.18	18.46%	43	0.17	0.09	0.26	27.08%
15	42	0.02	0.02	0.81	81.77%	45	0.13	0.15	0.16	16.41%	29	0.11	0.08	0.44	44.89%
Avg.	45	0.02	0.03	0.74	77.88%	45	0.09	0.10	0.25	26.04%	39	0.09	0.06	0.48	51.32%

Table 3-5

Multi-objective performance measures for $I_{chicago,100}$ test instances

Inst. #	*HQSM-SA-MA*					*HMOEA-GTA*					*SOFSS-MOEA*				
	Time m	GD	IGD	HV	HV%	Time m	GD	IGD	HV	HV	Time m	GD	IGD	HV	HV
1	46	0.03	0.06	0.62	66.68%	45	0.06	0.07	0.33	35.36%	43	0.01	0.01	0.91	98.99%
2	46	0.02	0.02	0.77	80.62%	45	0.06	0.06	0.36	37.65%	45	0.04	0.02	0.64	66.89%
3	46	0.02	0.02	0.78	83.79%	45	0.04	0.05	0.39	42.19%	45	0.09	0.03	0.55	59.10%
4	46	0.02	0.04	0.60	72.89%	45	0.04	0.05	0.27	32.95%	45	0.01	0.02	0.65	78.87%
5	46	0.03	0.05	0.67	74.10%	45	0.04	0.06	0.42	46.23%	45	0.05	0.04	0.52	57.90%
6	46	0.03	0.07	0.63	69.06%	45	0.08	0.09	0.26	28.34%	45	0.04	0.03	0.63	68.48%
7	46	0.02	0.04	0.71	77.38%	45	0.06	0.06	0.29	31.55%	45	0.03	0.02	0.59	64.72%
8	46	0.01	0.02	0.68	74.93%	45	0.05	0.05	0.34	37.76%	45	0.02	0.01	0.71	78.24%
9	46	0.01	0.02	0.76	87.26%	45	0.04	0.05	0.38	43.92%	45	0.02	0.02	0.59	68.44%
10	46	0.03	0.05	0.70	76.09%	45	0.06	0.07	0.34	36.57%	45	0.05	0.02	0.60	65.08%
Avg.	46	0.02	0.04	0.69	76.28%	45	0.05	0.06	0.34	37.25%	45	0.04	0.02	0.64	70.67%

Table 3-6

Summary results of all algorithms over all instances

			Instance type and size					
Alg.	Performance measures		Small-sized	$I_{500,50}$	$I_{1000,100}$	$I_{chicago,100}$	Avg. All	Avg. Large
HQSM-SA-MA	Deviation from global minima	f1	1.43%	3.98%	2.66%	4.39%	3.35%	3.45%
		f2	1.29%	3.38%	1.54%	13.86%	3.79%	3.85%
		f3	0.07%	1.06%	1.01%	3.14%	1.44%	1.48%
	Runtime (min)		4.34	24.47	44.74	46.05	24.73	37.47
	NDsetSize		15.34	27.83	27.71	36.34	24.31	29.91
	TrueNDsize		5.80	5.10	4.32	4.36	5.08	4.62
	GD		0.02	0.02	0.02	0.02	0.02	0.02
	IGD		0.02	0.03	0.03	0.04	0.03	0.03
	HV		0.70	0.68	0.74	0.69	0.70	0.71
	HV%		90.13%	77.23%	77.88%	76.28%	82.19%	77.24%
HMOEA-GTA	Deviation from global minima	f1	1.00%	4.44%	4.10%	5.54%	4.41%	4.59%
		f2	0.25%	4.45%	6.44%	34.18%	9.95%	10.21%
		f3	0.22%	7.90%	14.22%	19.64%	13.49%	13.89%
	Runtime (min)		43.58	45.02	45.05	45.05	44.48	45.04
	NDsetSize		28.49	47.07	38.55	49.73	38.37	44.54
	TrueNDsize		15.21	1.82	1.00	0.10	6.52	1.08
	GD		0.01	0.04	0.09	0.05	0.04	0.06
	IGD		0.01	0.04	0.10	0.06	0.05	0.07
	HV		0.75	0.49	0.25	0.34	0.51	0.36
	HV%		96.6%	55.2%	26.0%	37.3%	61.63%	39.77%
SOFSS-MOEA	Deviation from global minima	f1	48.17%	6.97%	2.73%	8.23%	7.40%	5.24%
		f2	0.00%	0.89%	2.68%	1.00%	1.95%	2.00%
		f3	0.09%	3.16%	5.41%	9.70%	5.63%	5.80%
	Runtime (min)		6.84	19.23	39.15	44.71	22.98	33.07
	NDsetSize		6.80	22.80	19.27	33.01	17.40	24.03

TrueNDsize	3.40	3.20	0.43	7.54	3.30	3.25
GD	0.02	0.05	0.09	0.04	0.05	0.06
IGD	0.05	0.03	0.06	0.02	0.05	0.04
HV	0.36	0.54	0.48	0.64	0.47	0.54
HV%	46.89%	61.72%	51.32%	70.67%	54.99%	60.06%

These findings are in line with the similar findings by Becerra and Coello (2006) that also demonstrate the MPT-based heuristics perform better than MOEAs in approximating the true POS of hard optimisation problems.

In the next section, we carry out some RS analyses to gain useful insights.

3.5.3 Reference sets analyses

The RSs found in the case of the considered instances are analysed further in this section to represent better the trade-offs among the three objectives of the SPRP and to observe the overall fleet size and mix, multiple trips and the routing patterns of the solutions. For representation, the RS of a selected instance (i.e. instance 11 in $I_{500,50}$) is illustrated in the 3-dimensional (3D) space in Figure 3-7.

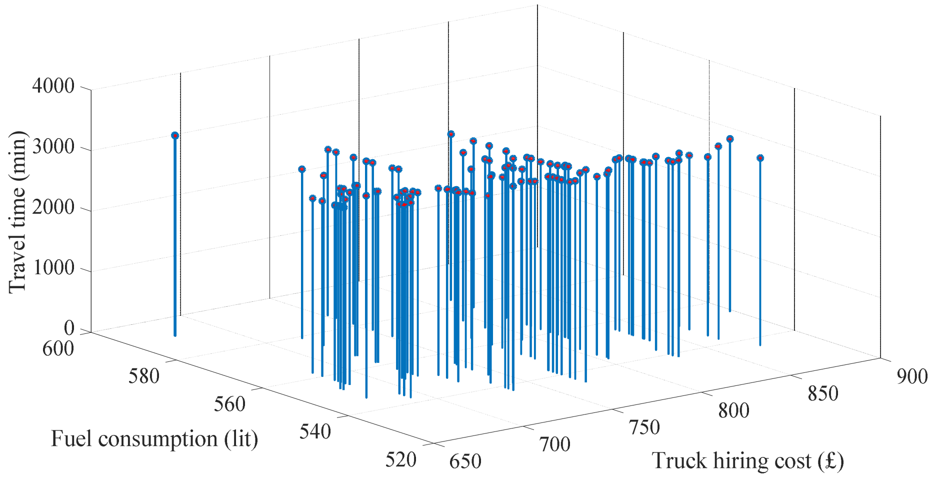
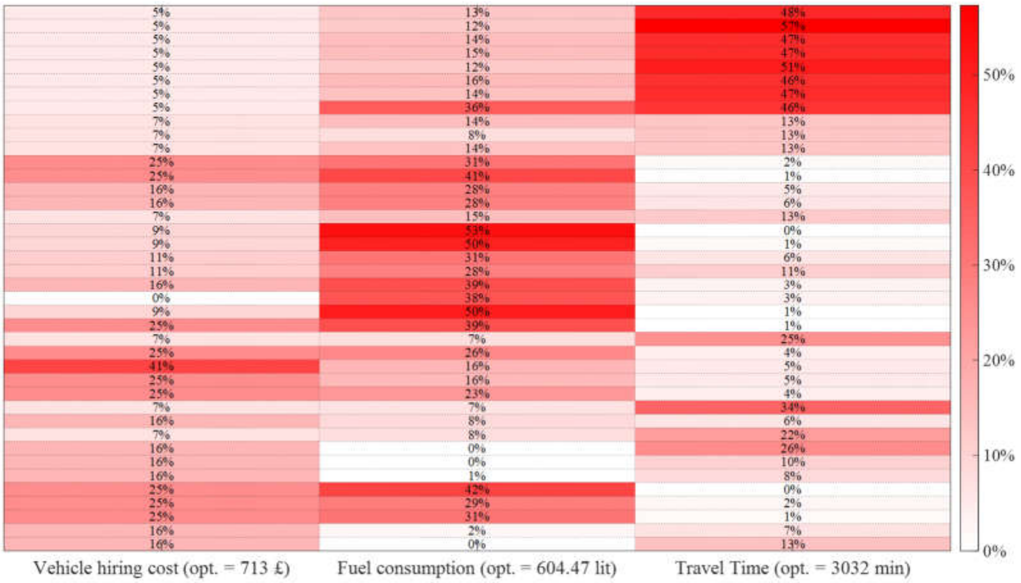


Figure 3-7 The RS of a selected SPRP instance

While visualisation of the Pareto front in case of multi-objective problems with more than two objective functions is not very easy, heat maps like the one presented in Figure 3-8, are appropriate visual aids for the decision maker to select a solution that provides a suitable compromise among the objective values. In this figure, the RS of instance 7 in $I_{500,50}$ is selected and the percentage deviations from the absolute minimum in case of each objective function for all the 40 solutions on the considered

RS is shown. While it can be observed that there exists a significant trade-off among the three objectives of the SPRP and the minimisation of one objective can significantly deteriorate the value of the other two, with the help of colour gradient, this figure makes it possible to visually locate the more balanced solutions. It is also worth noting that while the minimisation of vehicle hiring cost does not have a large negative impact on the travel time in the case of this instance, it has led to a significant increase in the fuel consumption objective. This common observation is mainly due to the fleet size and mix in emissions-efficient solutions where the use of a less environment-friendly truck of type heavy-duty is usually refrained and compensated by allocating several more light or medium duty trucks to the routes. However, due to capacity-efficiency, heavy-duty trucks can accommodate many customers in one go and lead to a smaller hiring cost and makespan.



Vehicle hiring cost (opt. = 713 £)	Fuel consumption (opt. = 604.47 lit)	Travel Time (opt. = 3032 min)
5%	13%	48%
5%	12%	57%
5%	14%	47%
5%	15%	47%
5%	12%	51%
5%	16%	46%
5%	14%	47%
5%	36%	46%
7%	14%	13%
7%	8%	13%
7%	14%	13%
25%	31%	2%
25%	41%	1%
16%	28%	5%
16%	28%	6%
7%	15%	13%
9%	53%	0%
9%	50%	1%
11%	31%	6%
11%	28%	11%
16%	39%	3%
0%	38%	3%
9%	50%	1%
25%	39%	1%
7%	7%	25%
25%	26%	4%
41%	16%	5%
25%	16%	5%
25%	23%	4%
7%	7%	34%
16%	8%	6%
7%	8%	22%
16%	0%	26%
16%	0%	10%
16%	1%	8%
25%	42%	0%
25%	29%	2%
25%	31%	1%
16%	2%	7%
16%	0%	13%

Figure 3-8 Heat map illustrating the ND solutions to a given SPRP instance

As an average of all the solutions on all RSs, Table 3-7 provides a summary report of a set of routing characteristics contained in the ND solutions. In this table, the avg. no. of MT light/medium/ heavy-duty rows indicate the average number of light, medium, and heavy duty trucks used for multiple-trips.

Table 3-7 Average characteristics of solutions on the RSs of the large sized instances

Characteristics	Instance type			Avg.
	$I_{500,50}$	$I_{1000,100}$	$I_{chicago,100}$	
Avg. no. of routes	9.88	19.17	19.26	16.10
Avg. no. of light-duty	0.15	0.11	0.69	0.32
Avg. no. of medium-duty	3.05	4.35	3.61	3.67
Avg. no. of heavy-duty	6.69	14.70	14.96	12.12
Avg. truck capacity utilisation	88.71%	86.73%	88.29%	87.74%
Avg. no. of MT light-duty	0.03	0.06	0.06	0.05
Avg. no. of MT medium-duty	0.90	0.58	0.39	0.62
Avg. no. of MT heavy-duty	0.72	0.36	1.37	0.82
Avg. no. of customers per route	5.12	5.24	5.37	5.25
Avg. makespan of each trip (min)	322.89	372.43	242.59	312.63
Avg. fuel consumption of each trip (lit)	61.87	80.80	34.83	59.16
Mean departure time from the depot	09:18	08:50	10:49	09:40

In this table, the rather large number of routes in the solutions, and consequently the rather small number of customers per route and the higher utilisation of heavy-duty trucks, is mainly due to the large size of the demands requested by the customers in the developed instances. This is confirmed by the row indicating the average truck capacity utilisation, which is indeed quite satisfactory. The table shows that as instances become larger, the heavy-duty trucks are the most preferred truck type in both cases of a single or multiple trips. The mean departure time from the depot, which is on average around 9:40 a.m., suggests that trucks are usually dispatched almost after the first morning peak congestion, and when the overall traffic speed starts to rise, to make the best use of the fastest period of the day. Note also in this table the smaller fuel consumption and makespan of each trip in the case of Chicago instances that indicates the travel times of network road-links in the case of these instances are comparatively higher.

3.5.4 Performance of the proposed neighbourhood exploration strategy

In this section, we analyse the performance of the neighbourhood exploration strategy introduced by the paper, i.e. the *ENS*. In order to do this, the first instance in $I_{500,50}$ is selected and 50 different feasible SPRP solutions are randomly generated.

These feasible solutions are then submitted as initial solutions to the neighbourhood exploration strategy of Bent and Van Hentenryck (2004) (denoted by BVH-NS), discussed in section 4 of the paper, and the *ENS*, for the improvement of total travel time. Both algorithms are given an equal time budget of 30 seconds and the improved solutions returned by them are compared. Figure 3-9 illustrates the result of these experiments.

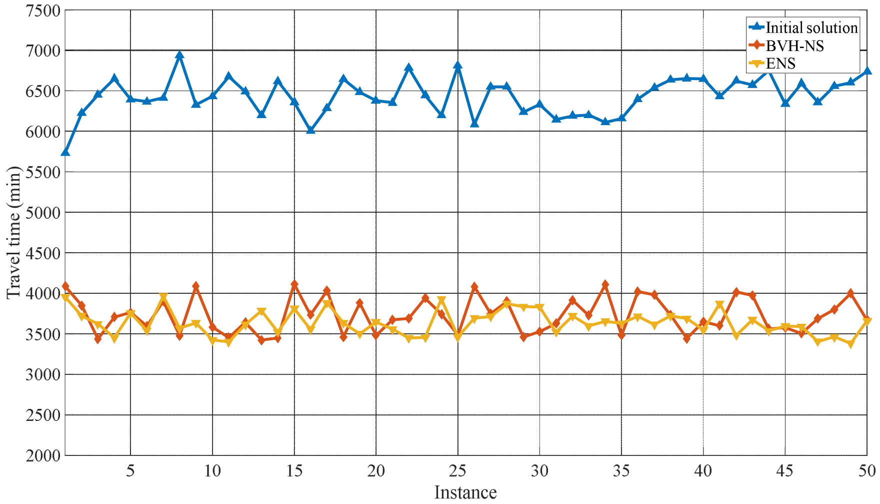


Figure 3-9 Comparison of the ENS and the BVH-NS

While both neighbourhood exploration strategies can very efficiently improve the initial solution by over 76%, the ENS is able to provide a better solution in 34 cases out of the entire 50 cases.

3.6 Discussion and concluding remarks

In this paper, we focused on solving a new realistic variant of the well-known PRP, called the SPRP, which is a tri-objective, time and load dependent, fleet size and mix PRP, with time windows, flexible departure times, and multiple-trips on congested urban road networks. The SPRP is a very difficult problem to solve as it contains several hard variants of the VRPTW, and entails addressing the problem of intermediate road-paths identification between truck visits on the original roadway graph, and solving several multi-objective lower level optimisation problems. In order to be able to approximate the true POS of practical real-life sized instances of the SPRP within a reasonable computational cost, the paper developed a multi-phase hybridised exact and heuristic solution framework comprising three different higher-level MOOHs that benefit from new lower-level heuristics designed to address the emerging optimisation problems that arise in the course of the SPRP solution evaluation. A new scheme of UENDA was incorporated into all algorithms, and a new neighbourhood exploration strategy was introduced in the paper.

The main MOOH proposed by the paper, i.e. the *HQSM-SA-MA*, hybridises successfully a recent efficient MPT to address tri-objective MOIP problems with a two-stage LS-based heuristic. With this approach, all difficult lower level multi-objective optimisation problems that must be addressed in the SPRP are reduced to easier single objective problems, and the search for ND solution progresses systematically along the non-inferior surface. The other two competing MOOHs developed by the paper are based on the well-known MOEAs, where a hybrid and simple MOEAs were implemented and compared with each other and the *HQSM-SA-MA*.

A new comprehensive library of large-sized SPRP test instances were developed by the paper and the performance of the proposed MOOHs were examined in addressing them. The computational results of the paper confirm the superior performance of the *HQSM-SA-MA* over the other two proposed MOEA-based MOOHs. The algorithm demonstrates a satisfactory scalability and is consistently able to obtain remarkable results on all test instances considered. Our findings confirm the findings by Becerra and Coello (2006) that also demonstrate MPT-based heuristics outperform MOEAs when tackling hard optimisation problems. The comparison of the other two MOEA-based heuristics, on the other hand, interestingly conveys that, as the problems become larger and more difficult to solve, the proposed simple MOEA is in general better performing than the hybrid MOEA, despite its more sophisticated design.

A more in-depth analysis of the obtained SPRP Pareto fronts reveals the significant trade-offs among the three objectives of the SPRP, and particularly show that the mere minimisation of the emissions can be so costly. Therefore, to enable decision makers to strike a balance between business and environmental objectives, it is always more realistic to provide them with a set of ND solutions that reflect the corresponding trade-offs efficiently.

While there are multiple research opportunities relevant to the study of VRPs directly on the roadway networks, future research can use the proposed benchmark test instances and solution algorithms as platform for investigating the effect of different perturbations in various characteristics of a logistics system like customers' demands, locations, and time-windows, depot location, and vehicle fleet characteristics, and carry out various scenario and what-if analyses. In addition to this, the proposed algorithms in this paper are SPRP dedicated and despite the fact that the SPRP per se includes different variants of the VRPTW, such as the TDVRPTW, the fleet size and mix VRPTW, the MT-VRPTW, and the MO-VRPTW, extra programming effort and parameter tuning is required to modify the algorithms to solve specific instances related to those variants. To address this limitation, in further research, the proposed algorithms can be extended to be used as unified general multi-objective solvers that can address these rich variants with minimum user interference.

4. THE ELECTRIC VEHICLE ROUTING PROBLEM WITH SYNCHRONISED AMBULANT BATTERY SWAPPING OR RECHARGING

4.1 Introduction

With the ever-increasing concerns about *Greenhouse Gas (GHG)* emissions from *Urban Freight Distribution (UFD)*, the European Commission has set a target for "essentially CO_2-free city logistics in urban centres by 2030" (European Commission, 2011). Meeting such target would inevitably entail facilitating the conversion of conventionally fuelled logistics fleet into *Electric Commercial Vehicles (ECVs)* with zero local emissions. This conversion of the fleet, however, is still so much constrained by ECVs' reduced driving range, long recharging time, and scarce and unevenly scattered *Charging Stations (CSs).* While reports (Committee on Climate Change, 2010) suggest a high uptake of ECVs will be possible by 2030 as electric *Light Goods Vehicles (LGVs)* will be then cost-saving compared to conventionally fuelled vehicles (Allen et al., 2017), for a smooth transition phase short term operational solutions are crucial.

The *Electric Vehicle Routing Problem with Time-Windows (EVRPTW)* (Schneider et al., 2014) is a variant of the VRP that aims at aiding companies operating on ECVs to overcome "range anxiety" by developing solutions that comprise the introduction of minimal detour in the vehicle route to visit CSs. The primary challenge in addressing the EVRPTWs, that distinguishes them from their *Green-VRP (G-VRP)* (Erdoğan & Miller-Hooks, 2012) counterpart, is in the significantly larger recharging time required to refill ECVs' batteries as compared to other alternative fuel vehicles. This limitation has implications mainly with regards to meeting customers' time-windows, and thus in the presence of realistic time windows the solutions yielded by EVRPTWs might be either infeasible or too expensive in terms of the number of ECVs required and the total distance travelled.

While partial recharging strategies instead of recharging the ECV battery fully upon arrival at a CS (Bruglieri et al., 2015; Desaulniers et al., 2016; Keskin & Çatay, 2016) can be helpful to a limited extent to address the large recharging time required, a more fundamental shift of paradigm is indeed needed to entice logistics companies to acquire ECVs in their fleet. One such attempt in recent years has been the idea of swapping the ECV's depleted battery with a fully charged one at a *Battery Swapping Station (BSS).* Leading EV companies such as the Chinese company *NIO®* are promising a 3-minute-long battery swapping experience and are making an investment to roll out at least 1100 power swap stations for battery swapping in main cities of China by 2020 (Manthey, 2017). This potential technological development has already motivated vehicle routing research and few studies have approached the problem as a *Battery Swap Station Location-Routing Problem (BSS-LRP)*, where the location of BSSs and the ECV routes considering their limited range must be determined (Hof et al., 2017; Yang & Sun, 2015). While BSS-LRPs might ultimately incur a smaller number of ECVs required to assign to delivery routes, and a shorter overall distance travelled, their application is inhibited by two major limitations. Firstly, opening a BSS can be significantly costly and the trade-off between acquiring more ECVs to operate the routes via visiting CSs and opening a BSS is yet unknown and very much problem dependent. The second issue is due to the nature of the business context in which ECVs are already operationalised; given their current payload restrictions, ECVs are mainly used in the small-package shipping industry by logistics companies like UPS, DHL, and DPD for last-mile deliveries (Schneider et al., 2014), and in such context customers locations changes on a daily basis and the location optimisation aspect of the BSS-LRP might not be deemed much pertinent.

To address the aforementioned limitations, in this study we introduce a new paradigm shift in EVRPTWs by exploiting new relevant technological developments that make mobile battery swapping or mobile rapid recharging possible. The development of a new fast battery swapping device installed on a *Battery Swapping Van (BSV)*, documented in patents Lu and Zhou (2013) and Gao et al. (2012), and justified and corroborated by the study of Shao et al. (2017), opens up new possibilities to logistics by providing an *'ambulant'* battery swapping mode, as opposed to the *'stationary'* battery swapping mode as in BSS-LRP. On par with this technology, the mobile recharging option developed by companies like *NIO®* and *FREEWIRE TECHNOLOGIES®* (Figure 4-1), implies many new possibilities, and of course unprecedented attractive research directions. For example, *NIO®* claims that their Power Mobile can travel anywhere to any EV in need of battery recharging and provide an extra 100 km with 10 min of charging.

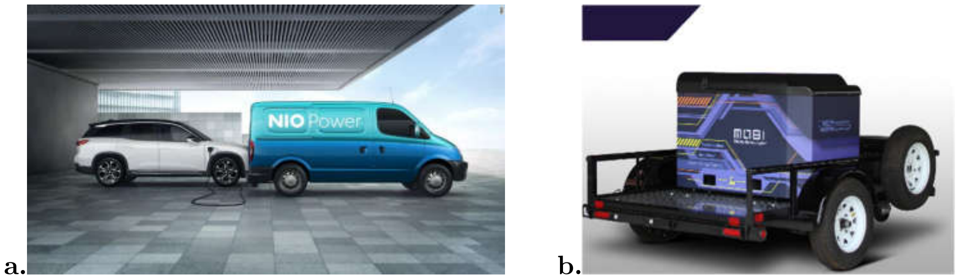


Figure 4-1 **(a)** NIO Power Mobile and **(b)** MOBI GEN TWS

Hence, in this study we introduce a new class of the EVRPTWs called the *Electric Vehicle Routing Problem with Synchronised Ambulant Battery Swapping/Recharging (EVRP-SABS)* that is motivated by these new technological developments. It is worth explaining our intentional choice of the word 'ambulant' over the word 'mobile' to name this class of the problem, as to connote the 'emergency' sense in the application of the BSVs in the design of delivery routes. In fact, we are aware that acquiring a BSV has its own acquisition cost and dispatching a BSV to visit an ECV in need of battery swapping/recharging incurs cost (e.g. proportional to the distance travelled). Moreover, keeping an inventory of spare electric batteries constitutes another cost element in the system. Hence, it is not sensible to entirely eliminate the option of recharging at a CS in the road network and replace it by mobile battery swapping/recharging, and instead this option is only kept as a last resort when a feasible solution via visiting CSs is not obtainable. Therefore, the proposed EVRP-SABS includes the EVRPTW with CSs and retains recharging at CSs as the primary solution to routing a fleet of ECVs.

Following the introduction of the EVRP-SABS, we focus on a key complication arising in the context of the EVRPTWs with CSs that corresponds to the optimal selection and placement of the CSs in energy infeasible delivery routes. To tackle this situation efficiently, we propose new combinatorial results that lead to the development of an exact *Eligible Paths Identification Procedure (EPIP)* that is used in a pre-processing stage to identify a priori all 'eligible' paths that pass through one or several CSs between a pair of required nodes and eliminate all redundant ones. The paper further develops closed form expressions for the precomputation of the required attributes of the identified paths and uses them in a significantly easier to solve EPIP-based formulation of the EVRP-SABS and its related variants, i.e. EVRPTW and GVRP, where all CSs are eliminated from the working multi-graph. It is demonstrated that by just putting the strengthened EPIP-based formulation into a standard branch-and-bound solver, one can solve and improve many of the test instances that were solved previously only using a sophisticated branch-price-and-cut algorithm (Desaulniers et al., 2016). Finally, to find near optimal solutions to practical-sized instances of the hard to solve EVRP-SABS, the paper proposes a *two-Stage MatHeuristic (2S-MatHeu)* algorithm that exploits the EPIP paths in its input and uses an exact *Dynamic Programming (DP)* algorithm at its core as the routine to solve the iteratively emerging route-path optimisation problem.

The contribution of this paper is multi-fold: (i) the EVRP-SABS is introduced and formulated as a *Mixed Integer Linear Programming (MILP)* model; in particular, the proposed formulation includes new synchronisation constraints that ensure a BSV and an ECV will be present at a designated time and location to perform a planned swap/recharge, (ii) an exact EPIP is proposed based on new combinatorial results that allow the identification of the set of the paths that must be retained between a pair of customers or a customer and the depot a priori, and closed form expressions are developed for the pre-computation of the paths attributes, (iii) a significantly strengthened EPIP-based formulation of the EVRP-SABS and the other related problem classes is developed and its performance against the existing formulations and algorithms is evaluated, and (iv) a new 2S-MatHeu algorithm is proposed to tackle practical sized instances of the EVRP-SABS; the proposed algorithm incorporates new neighbourhood exploration strategies in an *Intensified Large Neighbourhood Search (ILNS)* algorithm and uses a DP model for the route-path optimisation at its core.

In the remainder of this chapter, in section 4.2, a survey of the most pertinent literature is presented. Section 4.3 of the paper describes the EVRP-SABS formally and establishes the required notation, definitions and assumptions. Section 4.4 discusses the EPIP and the EPIP-based formulation. Section 4.5 introduces the proposed 2S-MatHeu. Section 4.6 presents the computational results of the EVRP-SABS and the proposed algorithms, while section 4.7 summarises the research conclusions.

4.2 Previous related work

The significant share of the road freight distribution in the global emissions of *Greenhouse Gases (GHGs)* and other environmental pollutants has motivated a surge of interest in the study of *Vehicle Routing and Scheduling Problems (VRSPs)* with environmental considerations in recent years. Research work in this area might be broadly categorised into: (i) the *Emissions minimising VRPs (EM-VRPs)* comprising the pollution routing problem (Bektaş & Laporte, 2011) and its variants (Androutsopoulos & Zografos, 2017; Demir et al., 2012; Franceschetti et al., 2017; Koç et al., 2014; Raeesi & Zografos, 2019), that aim at minimising the fuel consumption incurred by the delivery routes as a proxy for emissions, and (ii) the *Green-VRPs (G-VRPs)* that are concerned with routing a fleet of vehicles that run on a cleaner alternative fuel (Erdoğan & Miller-Hooks, 2012; Raeesi & O'Sullivan, 2014; Salimifard & Raeesi, 2014) or electric batteries (Bruglieri et al., 2015; Conrad & Figliozzi, 2011; Desaulniers et al., 2016; Hiermann et al., 2016; Schneider et al., 2014). There is also recent research that bridges these two categories by routing a mixed fleet of electric and conventional vehicles (Goeke & Schneider, 2015; Macrina et al., 2018). The interested reader is referred to Bektaş et al. (2019) for an up-to-date review of the key papers in the field. We also refer the reader to the paper by Pelletier et al. (2016) on goods distribution with ECVs that serves as a good starting point to discover the fundamentals of the ECV technology and its relevant economic and operational aspects. In what follows, a concise review of the papers that are most pertinent to this study is presented. This includes a brief discussion on a class of routing problems that may not be explicitly classified under VRSPs with environmental considerations but share some key features with the problem considered in this paper.

The EVRPTW (Schneider et al., 2014) can be viewed as a special case of the G-VRP (Erdoğan & Miller-Hooks, 2012) where capacity constraints and time-windows are added to the problem, and significantly larger refuelling (recharging) time is assumed. In the variant considered by Schneider et al. (2014) a minimum number of ECVs must be assigned to energy-feasible delivery routes (potentially visiting one or several CSs) that visit each customer exactly once during their pre-defined time-windows, such that the total capacity constraint of the ECV is not violated and the total distance travelled is minimised. Due to the limited driving range of ECVs, the core complication in the EVRPTW is related to the introduction of minimal detours in the vehicle routes to visit available CSs on the working graph to fully recharge their battery and carry on the delivery task. They formulate this problem as a MILP on an augmented graph of the customers and CSs which includes 'sufficient' dummy copies of the CSs to allow several visits to the same CS, but since the graph grows quickly in size with the increasing size of the instances, the formulation can only handle small-sized instances of 10, and few instances of 15 customer nodes and 5 CSs. An algorithm based on the hybridisation of a *Variable Neighbourhood Search (VNS)* algorithm with a *Tabu Search (TS)* heuristic is further developed by Schneider et al. (2014), where to handle the selection and placement of CSs into energy-infeasible routes, some of the exiting neighbourhood operators, such as 2-opt, relocate, and exchange are customised and used in their TS, and a new dedicated move operator called the *stationInRe* operator that performs insertions and removals of CSs is introduced.

To allow more flexibility in the design of the ECV delivery routes, Keskin and Çatay (2016) relax the full recharging restriction and allow partial recharging at a CS. To accommodate the decision on charging level upon arrival at a CS in the MILP proposed by Schneider et al. (2014), they introduce a new decision variable to represent the battery state of charge on departure from a CS. An *Adaptive Large Neighbourhood Search (ALNS)* algorithm that comprises new heuristics for station removal and station insertion is employed to solve the proposed problem. Their station insertion procedure in particular tries to repair an energy-infeasible solution by using one of the three different insertion algorithms they propose. Their computational results demonstrate that with partial recharging instead of full recharging, the solution to a few test instances can be improved.

Other variants of the EVRPTW considering different recharging strategies, recharging functions and fleet composition have been also explored in the literature. Felipe et al. (2014) propose a heuristic to solve a variant in which in addition to the decision on the charging level at a CS, the technology used for recharging e.g. regular or fast recharging is considered. Montoya et al. (2017) argue that the recharging level of the battery is a non-linear function of the recharging time and study the EVRP (without time windows) with a nonlinear recharging function. Hiermann et al. (2016) consider the fleet size and mix in the EVRPTWs where the available vehicle types in the fleet differ in terms of their capacity, battery size and acquisition cost. Goeke and Schneider (2015) study the EVRPTW with a mixed fleet of ECVs and conventional internal combustion commercial vehicles. A distinctive feature of their study is that instead of simply assuming energy consumption is a linear function of the distance travelled, they utilise an energy consumption model that takes speed, road slope and vehicle payload into account. In the same vein, Basso et al. (2019) incorporate into the routing decision an improved and more accurate energy consumption estimation model comprising detailed topography and speed profiles.

As regards exact solutions to the EVRPTW, Desaulniers et al. (2016) develop a branch-price-and-cut algorithms for the problem that is able to solve instances with up to 100 customers and 21 recharging stations. To deal with their *Column Generation (CG)* subproblem that corresponds to a variant of the *Elementary Shortest-Path Problem with Resource Constraints (ESPPRC)*, they develop mono-directional, as well as, bi-directional labelling algorithms, and they extend these algorithms to deal with four different variants of the EVRPTW that are distinguished from one another on the basis of the recharging strategy at a CS employed, and the recharging frequency over a delivery route allowed.

In the proposed work in this study, we demonstrate that through new analytical findings and using a simple EPIP in a fast pre-processing stage, it is possible to work on a multi-graph of customer nodes only and eliminate the CSs altogether from consideration. The benefits of such an approach is multi-fold and can address several limitations in the state-of-the-art literature on EVRP, and

GVRP, in general. Firstly, there is no need to formulate the problem on an augmented graph with dummy copies of CSs that soon becomes intractable. We will demonstrate that by just putting a strengthened EPIP-based formulation of the problem in CPLEX, not only many of the problem instances that are intractable for existing formulations are handled very efficiently, but also it is possible to solve and improve some of the instances with 100 customers and 21 CSs that have been only possible to approach using a sophisticated branch-price-and-cut algorithm previously (Desaulniers et al., 2016). Secondly, using the set of EPIP paths we address a shorcoming of the heuristic algorithms in the papers discussed above, corresponding to the exact evaluation of a sequence of customer visits in an ECV route. Indeed, to the best of our knowledge, in all heuristic algorithms for the EVRPTW the problem of the selection and placement of a CS in an energy-infeasible ECV route can be only dealt with in a rather stochastic fashion through the application of random neighbourhood exploration operators. Whereas, using the EPIP paths, one is only solving a VRPTW on a multi-graph instead of an EVRPTW with CSs, and any heuristic developed for the VRPTWs is applicable, and thus exact evaluation of ECV routes is possible.

While the problem introduced by this paper, i.e. the EVRP-SABS, has not been previously studied, due to the existence of spatial and temporal synchronisation requirement of an ECV with a BSV in the model, it has some similarities with a class of VRPs known as the *two-Echelon VRP with Satellite Synchronisation (2E-VRP-SS)*. In 2E-VRPSSs (Anderluh et al., 2017; Crainic et al., 2009; Grangier et al., 2016) the required temporal synchronisation of the vehicles in the first echelon with the vehicles of the second echelon at an intermediate site, called satellites, resembles the kind of synchronisation one must establish for a planned battery swap in EVRP-SABS. The main complication that arises in establishing such synchronisation is due to the fact that unlike the standard VRPs where vehicles are independent of one another, in VRPs with temporal synchronisation constraints a change in one route may affect other routes, and in the worst case, a change in one route may render all other routes infeasible. This problem is known as the ‘interdependence problem’ (Drexl, 2012), and to address it, Grangier et al. (2016) propose to represent the time constraints as a directed acyclic graph called a precedence graph. They use this graph in their route scheduling and feasibility algorithm which is placed at the heart of their proposed ALNS algorithm for the problem. Anderluh et al. (2017) consider temporal and spatial synchronisation between cargo bikes and vans, and propose a heuristic based on a greedy randomized adaptive search procedure with path relinking for the problem.

In this paper, we exploit the proposed EPIP to address the synchronisation requirement efficiently by decomposing the problem into two stages, where any need for a battery swapping is recognised in the first stage of the algorithm that solves an EVRPTW on the EPIP multi-graph, and the second stage algorithm only has to deal with a very simple VRPTW that can be solved to optimality.

4.3 The EVRP-SABS: formal description and formulation

In this section, we first provide a formal description of the EVRP-SABS and discuss the notation, definitions and key assumptions used in this paper. Next, a small illustrative example of the problem is presented to establish a case for it, and following that, the mathematical formulation of the problem is provided. It is worth noting that without any loss of generality in the rest of this paper we focus on the ambulant battery swapping rather than ambulant recharging. It is evident from the context that all arguments throughout the paper hold true in the case of ambulant recharging, too.

4.3.1 Formal description of the problem

The EVRP-SABS is defined on a complete, directed graph $G = (N, A)$, where N is the set of network nodes and $A = \{(i,j)|i,j \in N, i \neq j\}$ is the set of directed arcs. The set $N = \{D \cup C \cup R\}$ is comprised of the depot $D = \{0, n+m+1\}$, with $\{n+m+1\}$ being a dummy copy of $\{0\}$, customer nodes $C = \{1,2,\dots,n\}$, and CSs $R = \{n+1, n+2, \dots, n+m\}$. Each customer $i \in C$ is associated with a certain demand q_i to be delivered within its pre-determined hard time window, denoted by $[e_i, l_i]$, with service time s_i. The depot working hours, which define the planning horizon, is denoted by $T = [e_0, l_0]$. To each arc $(i,j) \in A$, a distance d_{ij}, and a travel time t_{ij} is attributed. There is a fleet of homogeneous ECVs and a fleet of homogenous electric BSVs that are all fully charged and located in the central depot. To each ECV a maximum payload Q_e, a battery capacity B_e, and an energy consumption rate per unit distance travelled r_e is attributed. Each BSV, on the other hand, can carry a maximum number of batteries Q_b, has a battery capacity B_b, and an energy consumption rate r_b. Moreover, the following key assumptions are made in the proposed EVRP-SBS:

- The tasks of ECVs and BSVs are not interchangeable. That is, ECVs are only meant to deliver the requests of customers, and BSVs are only used when battery swapping is required by an ECV, and they cannot be used for delivery.
- ECVs are allowed to visit CSs for recharging their batteries for the difference between their present charge level and B_e (full recharge). Recharging time is assumed proportional to the amount of energy recharged at a recharging rate of g. While it is not restrictive to also allow BSVs to visit CSs, it is reasonable to ban them from doing so.
- Battery swapping must be carried out at a customer location and realistically it cannot be done simultaneously with the ECV providing service at the customer. Hence, battery swapping can only start once ECV service is over.

The arrival time of the BSV at the swapping site must be therefore synchronised with the ECV's service finish time. However, the BSV can arrive earlier and wait till swapping starts. It is assumed that swapping takes $\mathscr{s}$ time units. Note that fixing the spatial aspect of the required spatiotemporal synchronisation between an ECV and a BSV to a customer location is not a restrictive assumption, and it is possible to introduce other separate designated points of swapping to G.

- BSVs are not allowed to require a battery swap from other BSVs during their trip.
- An ECV can ask for a battery swap for as many times required during its trip, and there is no restriction for a BSV to serve the same ECV several times.

The aim of the EVRP-SABS is to determine an optimal composition of ECVs and BSVs in the fleet to operate routes that start and finish at the depot, and serve every customer exactly once within their pre-defined time-windows, without violating vehicles' payload and battery capacities and working day limits, such that the following two objectives are minimised lexicographically: (i) the total number of ECVs and BSVs required, and (ii) the total travelled distance of ECVs and BSVs.

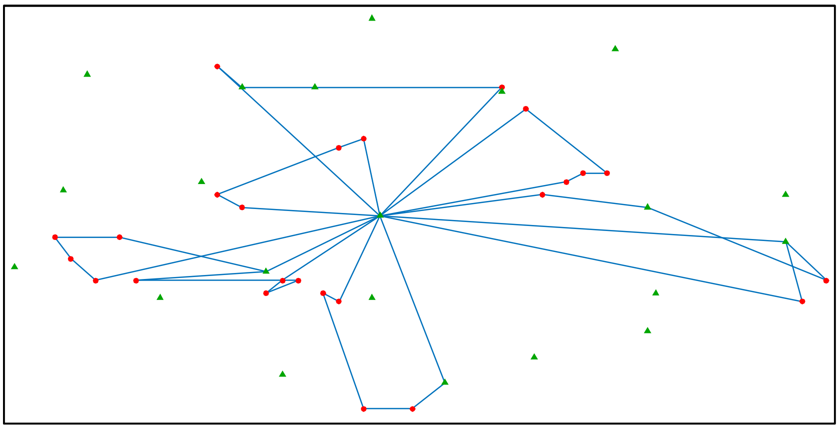
Prior to the presentation of an initial formulation for the EVRP-SABS, for illustration, an example of the EVRP-SABS is provided in the next subsection.

4.3.2 An illustrative example

An instance of the EVRPTW with 25 customer nodes and 21 CSs is selected from Desaulniers et al. (2016) (i.e. instance C106) to illustrate the EVRP-SABS against the EVRPTW. To modify this instance such that battery swaps might be required by ECVs we need to either increase the recharging time or decrease the battery capacity, and hence we have only multiplied the g value by three. The optimal delivery routes resulted by solving the resulting problem as an EVRPTW versus an EVRP-SABS are illustrated in Figure 4-2.a and Figure 4-2.b, respectively. The optimal solution yielded by the EVRPTW formulation (Figure 4-2.a) requires 8 ECVs, incurs a total distance of 778.07, and involves 7 visits to CSs. The EVRP-SABS solution, on the other hand (Figure 4-2.b), requires 6 ECVs that travel a total distance of 597.63, visit 3 times the CSs and call for 3 battery swaps. To provide the requested swaps, one BSV is needed to travel a distance equal to 45.71. Hence, the EVRP-SABS requires 7 vehicles in total (6 ECVs + 1 BSV) to travel a total distance of 643.35.

It is also interesting to further investigate the planned route of the BSV. The total 3 battery swaps requested in this example, are only called for by two ECVs, such that the BSV executes one battery swap for ECV1 and goes to a designated customer location to visit ECV2 after it completes its service at the customer in that location; in the meantime, following its first requested swap, ECV1 has completed service at 6 more customer locations and needs another round of battery swap which is provided by the same BSV.

a.



b.

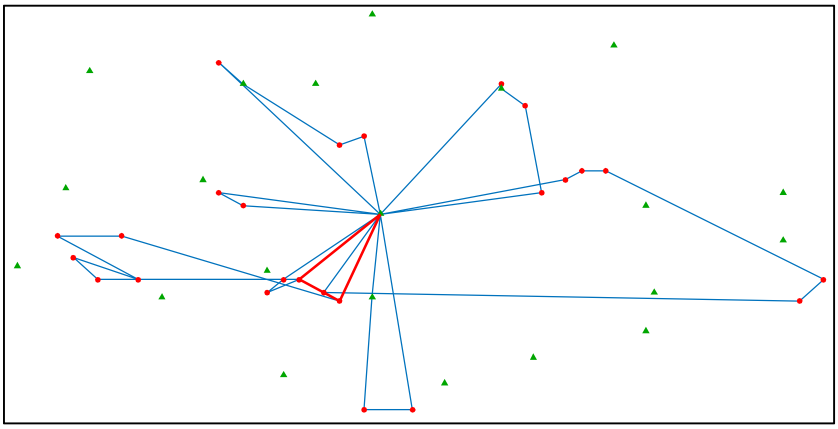


Figure 4-2 Optimal routes returned by **(a)** EVRPTW and **(b)** EVRP-SABS

It is important to note that in this example a feasible solution for the EVRPTW was obtainable. It is possible that in real life cases one cannot even find a feasible solution to the EVRPTW by just visiting CSs. Indeed, all exiting benchmark instances of the EVRPTW are forced feasible by design. For example, in the description of their proposed benchmark instances, Schneider et al. (2014) state that “the detours for visits to recharging stations and the recharging times incurred make it impossible to comply with the customer time windows given in the original Solomon instances, i.e., some instances become infeasible because no possibility exists to reach certain customers within their original time window”. Note that, using CSs only, instances with a fixed size fleet might be also infeasible. Hence, the key justification for the EVRP-SABS is indeed its ‘necessity’ rather than its ‘benefits’ only.

In the next subsection, we provide an initial formulation of the EVRP-SABS that is built upon the original EVRPTW formulation by Schneider et al. (2014). Obviously, such formulation is not able to handle problem instances like the one presented in this section. Later in the paper, we will discuss how this initial formulation can be strengthened using new combinatorial results to solve such instances within a reasonable computational time.

4.3.3 Mathematical formulation of the problem

Prior to discussing the mathematical formulation of the problem, the following auxiliary sets are defined for ease of reference: $N_1 = C \cup R'$, $N_2 = \{0\} \cup C$, $N_3 = C \cup \{n+m+1\}$, $N_4 = \{0\} \cup C \cup R'$, $N_5 = C \cup R' \cup \{n+m+1\}$, $N_6 = \{0\} \cup R'$, and $\mathcal{A} = \{(i,j) | i \in N_2, j \in N_3, i \neq j\}$. In these sets, R' is a sufficiently large set of dummy nodes generated to allow several visits to each CS in the set R. Note that following these definitions, the set of directed arcs A could be better written as $A = \{(i,j) | i \in N_4, j \in N_5, i \neq j\}$.

The MILP formulation of the EVRP-SABS works with the following decision variables:

- x_{ij}: Binary variable equal to 1 iff arc $(i,j) \in A$ is traversed by an ECV.
- z_{ij}: Binary variable equal to 1 iff arc $(i,j) \in \mathcal{A}$ is traversed by a BSV.
- y_i: Continuous variable denoting the time of arrival of an ECV at node $i \in N$.
- w_i: Continuous variable denoting the time of arrival of a BSV at node $i \in N \backslash R'$.
- f_i: Continuous variable denoting the remaining load on an ECV upon arrival at node $i \in N$.
- h_i: Integer variable denoting the number of the remaining fully-charged batteries on the BSV upon arrival at node $i \in N \backslash R'$.
- u_i: Continuous variable denoting the remaining battery charge level of an ECV on arrival at node $i \in N$.
- v_i: Continuous variable denoting the remaining battery charge level of a BSV on arrival at node $i \in N \backslash R'$.

The mathematical formulation of the EVRP-SABS is given by (4-1)-(4-20).

$$Min \sum_{(i,j) \in A} d_{ij} x_{ij} + \sum_{(i,j) \in \mathcal{A}} d_{ij} z_{ij} \qquad (4\text{-}1)$$

Subject to:

$$\sum_{j \in N_5} x_{ij} = 1, \qquad \forall i \in C \tag{4-2}$$

$$\sum_{j \in N_5} x_{ij} \leq 1, \qquad \forall i \in R' \tag{4-3}$$

$$\sum_{j \in N_4} x_{ji} - \sum_{j \in N_5} x_{ij} = 0, \qquad \forall i \in N_1 \tag{4-4}$$

$$\sum_{j \in N_2} z_{ji} - \sum_{j \in N_3} z_{ij} = 0, \qquad \forall i \in C \tag{4-5}$$

$$\begin{aligned} y_i + (t_{ij} + s_i)x_{ij} + \mathscr{s} \sum_{\sigma \in N_3} z_{i\sigma} - (l_0 + \mathscr{s})(1 - x_{ij}) \leq y_j, \quad \forall i \in N_2, j \\ \in N_5, i \neq j \end{aligned} \tag{4-6}$$

$$\begin{aligned} y_i + t_{ij}x_{ij} + g(B_e - u_i) - (l_0 + gB_e)(1 - x_{ij}) \leq y_j, \qquad \forall i \in R', j \\ \in N_5, i \neq j \end{aligned} \tag{4-7}$$

$$w_i \leq y_i + s_i, \qquad \forall i \in N_2 \tag{4-8}$$

$$y_i + (t_{ij} + s_i + \mathscr{s})z_{ij} - l_0(1 - z_{ij}) \leq w_j, \qquad \forall i \in N_2, j \in N_3, i \neq j \tag{4-9}$$

$$e_i \leq y_i \leq l_i, \qquad \forall i \in N \tag{4-10}$$

$$0 \leq f_j \leq f_i - (q_i x_{ij}) + Q_e(1 - x_{ij}), \qquad \forall i \in N_4, j \in N_5, i \neq j \tag{4-11}$$

$$0 \leq f_0 \leq Q_e \tag{4-12}$$

$$0 \leq h_j \leq h_i - z_{ij} + Q_b(1 - z_{ij}), \qquad \forall i \in N_2, j \in N_3, i \neq j \tag{4-13}$$

$$0 \leq h_0 \leq Q_b \tag{4-14}$$

$$\begin{aligned} 0 \leq u_j \leq u_i - (r_e d_{ij} x_{ij}) + (B_e \sum_{\sigma \in N_2 : j \in C} z_{\sigma j}) + B_e(1 - x_{ij}), \\ \forall i \in C, j \in N_5, i \neq j \end{aligned} \tag{4-15}$$

$$0 \leq u_i \leq B_e, \qquad \forall i \in N \tag{4-16}$$

$$0 \leq u_j \leq B_e - (r_e d_{ij} x_{ij}) + (B_e \sum_{\sigma \in N_2 : j \in C} z_{\sigma j}), \quad \forall i \in N_6, j \in N_5, i \neq j \tag{4-17}$$

$$u_i \geq \sum_{j \in N_5} r_e d_{ij} x_{ij}, \qquad \forall i \in C \tag{4-18}$$

$$0 \leq v_j \leq v_i - (r_b d_{ij} z_{ij}) + B_b(1 - z_{ij}), \qquad \forall i \in N_2, j \in N_3, i \neq j \tag{4-19}$$

$$0 \leq v_i \leq B_b, \qquad \forall i \in N \backslash R' \tag{4-20}$$

Expressions (4-1) is the objective function that minimises the total distance travelled by the ECVs and the BSVs assigned to the routes. Constraints (4-2) to (4-5) are routing constraints, constraints (4-6) to (4-10) are scheduling and temporal synchronisation constraints, constraints (4-11) to (4-14) are capacity constraints, and finally, constraints (4-15) to (4-20) are battery level control and swapping determination constraints. These constraints are further detailed below.

Constraints (4-2) indicate that each customer must be visited exactly once by an ECV for delivery. Constraints (4-3) ensure that a CS is not visited more than once by an ECV. Recall that extra visits to a same CS has been made possible by extending the set R to R'. Constraints (4-4) and (4-5) together guarantee that if a vehicle (i.e. an ECV or a BSV) enters a node, it should exit the node. Constraints (4-6) determine the arrival time of an ECV at each node taking into account the arrival time at the upstream node, its service time, and possibly its required time for a requested swap by a BSV. Constraints (4-7) do so when the upstream node is a CS and take into account the time required to fully charge the battery. Constraints (4-8) and (4-9) together are synchronisation constraints and ensure that a planned swap service by a BSV takes place after service at the customer is completed by the ECV. Constraints (4-10) are time-windows constraints.

Constraints (4-11) and (4-12) ensure demand fulfilment while guaranteeing that the capacity of the ECVs is not violated, and constraints (4-13) and (4-14) do the same for BSVs. Constraints (4-15) to (4-18) determine the battery charge level of ECVs after visiting a customer and/or a CS. Note that both constraints (4-15) and (4-16) are needed to determine the battery level of an ECV if a battery swapping by a BSV is scheduled. The mere use of constraints (4-15) is not sufficient, as in the case of non-zero u_i, by battery swapping the battery capacity will be exceeded without (4-16). Finally, constraints (4-19) and (4-20) ensure that the battery charge of a BSV never falls below 0.

In the next section, we discuss new important combinatorial results that make the development of much more compact and strengthened formulations for the EVRP-SABS, as well as the EVRPTW and the GVRP possible. These results are also very useful for exact solution evaluation in the course of a heuristic solution algorithm.

4.4 The exact Eligible Paths Identification Procedure (EPIP)

A major complication arising in the context of the EVRPTWs with CSs corresponds to the exact evaluation of an energy infeasible sequence of customer visits (an ECV route) that must be made feasible by imposing optimal visits to available CSs in the network. To get a picture of the source of this complication, consider only the number of paths that can connect an origin customer $\boldsymbol{o}$ to a destination customer $\boldsymbol{d}$ by passing through one or several of the available CSs in the graph. Indeed, given that there exists m CSs in G, there are $\sum_{i=1}^{m} a_i$ (where $a_i = a_{i-1}(m - i + 1)$, and $a_0 =$ 1) different distinct paths between $\boldsymbol{o}$ and $\boldsymbol{d}$ that pass through at least one CS. That is, for a value of m as small as 6, there are 1,956 different paths between $\boldsymbol{o}$ and $\boldsymbol{d}$; needless to say, that this increases exponentially; e.g. when $m = 10$, the number of paths increases to 9,864,100 paths. With this complication the problem formulation gets very soon intractable when introducing also dummy copies of CSs into the model. Note that this complication is not only present when one intends to use the MILP formulation in a standard branch-and-bound solver to find an optimal solution, but also it appears, for example, in the lower level Shortest Path Problem with Resource Constraint (SPPRC) that emerges when developing CG-based exact solutions, or when evaluating new solutions generated in the course of a heuristic algorithm.

It is, however, possible to identify many of the paths that cannot be part of an optimal solution to the EVRPTW with CSs or the EVRP-SABS and eliminate them from consideration. In this section we introduce new combinatorial results to do so efficiently using an exact Eligible Paths Identification Procedure (EPIP). We will show that by just putting an EPIP-based version of the problem formulation in a solver, EVRPTW instances that are well-beyond the previously accessible problem sizes and get as large as 100 customer nodes with 21 CSs, could be solved to optimality.

Prior to presenting these results, some terms that are used in the rest of this section are defined below:

- *Required nodes:* required nodes (N_r) are the nodes on G that represent the location of the depot and the customers; i.e. $N_r = D \cup C$.
- *Direct edge:* a direct edge ϵ_{ij}, or simply an edge, is hereafter an arc $(i,j) \in A | i,j \in N_r, i \neq j$.
- *CS-path:* a CS-path p_{ij}, is a sequence of arcs in A that passes through at least one CS on G and connects a pair of required nodes $i,j \in N_r$, $i \neq j$; i.e. $p_{ij} = [(i,1),(1,2),\dots,(\ell,j)], 1..\ell \in R$. By convention, let us assume that $\wp_{ij}$ is the set of all possible CS-paths between a pair of required nodes, i.e. $\wp_{ij} = \{p_{ij,1}, p_{ij,2}, \cdots, p_{ij,\ell}\}$ (remember that identifying this set can be intractable).
- *Battery Charge Level (BCL):* The BCL indicates the remaining battery charge level of an ECV when departing from node $i \in N_r$ and is denoted by u_i.

- *Path attributes:* to each CS-path $p_{ij,i} \in \mathscr{p}_{ij}$, several attributes such as distance and travel time could be attached, that we henceforth call path attributes.

Most of the arguments and results presented in the sequel rely on the BCL at the origin node of a pair of required nodes i and j, and hence a notion called *'BCL-dependency'* is introduced below:

Definition 1 BCL-dependent and BCL-independent path attributes: *a given path attribute of a CS-path $p_{ij,i} \in \mathscr{p}_{ij}$ is called a BCL-dependent path attribute if for its computation a knowledge of u_i is required, and its value depends on the value of u_i; on the contrary, if the value of the path attribute is unaffected by u_i it is called a BCL-independent path attribute.*

We use a small example, shown in Figure 4-3, to illustrate better this notion of the BCL-dependency and the forthcoming arguments of this section. Assume the number on each arc in this figure represents both the distance and the travel time of each arc. Also, assume $B_e = 10$, $r_e = 1$, and $g = 2$. Now, considering the CS-path (i, k, j) (which we call $p_{ij,1}$), it is clear that, while its distance $d(p_{ij,1})$ is BCL-independent, and regardless of the BCL at node i, is equal to 10, its travel time $\tau^{u_i}(p_{ij,1})$ is BCL-dependent and is determined based on the value of u_i. For example, for $u_i = 7$, we have $\tau^7(p_{ij,1}) = 22$, while for $u_i = 4$, we have $\tau^4(p_{ij,1}) = 28$.

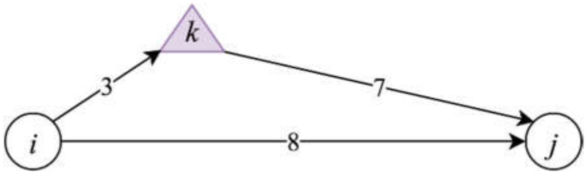


Figure 4-3 A small example of two required nodes and one CS

Proposition 1 *Regardless of the number of CSs visited on the CS-path* $p_{ij,i} \in \mathscr{p}_{ij}$*, knowing* u_i *is sufficient to compute* $\tau^{u_i}(p_{ij,i})$ *using the closed form expression* $\tau^{u_i}(p_{ij,i}) = \mathbb{t}_{ij,i} + g(B_e - u_i)$*, where* $\mathbb{t}_{ij,i}$ *is the travel time of* $p_{ij,i}$ *when* $u_i = B_e$*.*

To demonstrate the application of the closed form expression presented in Proposition 1, we refer back to the example in Figure 4-3. In this example, $\mathbb{t}_{ij,1} = 16$, and hence for $u_i = 7$, we have $\tau^7(p_{ij,1}) = 16 + 2(10 - 7) = 22$.

Definition 2 Endpoint Battery Discharge Level (EBDL): *EBDL is a BCL-independent path attribute for a given CS-path* $p_{ij,i} \in \mathscr{p}_{ij}$*, i.e.* $\pi(p_{ij,i})$*, that denotes the discharged level of the ECV's battery upon arrival at the endpoint of the CS-path* $p_{ij,i}$*.*

Proposition 2 *Regardless of the number of CSs visited on the CS-path* $p_{ij,i} \in \mathscr{p}_{ij}$*, we always have* $\pi(p_{ij,i}) = r_e d_{\ell j}$*, where* $d_{\ell j}$ *is the distance of the last arc in* $p_{ij,i}$*.*

In the case of the example in Figure 4-3, we have $\pi(p_{ij,1}) = 7$.

Definition 3 Minimum Required BCL (MR-BCL): *MR-BCL is a BCL-independent path attribute for a given CS-path* $p_{ij,i} \in \mathscr{p}_{ij}$*, i.e.* $\varphi(p_{ij,i})$*, that denotes the minimum value of* u_i*, below which it is not possible to traverse the CS-path* $p_{ij,i}$*. Hence,* $\varphi(p_{ij,i}) = r_e d_{i1}$*, where* d_{i1} *denotes the distance of the first arc in* $p_{ij,i}$*.*

Obviously, in the case of the example in Figure 4-3, we have $\varphi(p_{ij,1}) = 3$.

Lemma 1 *Given two different CS-paths* $p_{ij,1}, p_{ij,2} \in \mathscr{p}_{ij}$*, if* $\tau^{B_e}(p_{ij,1}) \leq \tau^{B_e}(p_{ij,2})$*, then* $\tau^{u_i}(p_{ij,1}) \leq \tau^{u_i}(p_{ij,2}), \forall u_i \in [\varphi(p_{ij,1}), B_e]$*.*

Proof. The condition that $\tau^{B_e}(p_{ij,1}) \leq \tau^{B_e}(p_{ij,2})$ means that the travel time of $p_{ij,1}$ when $u_i = B_e$, i.e. $\mathbb{t}_{ij,1}$, is less than or equal to the travel time of $p_{ij,2}$ when $u_i =$

B_e, i.e. $\mathbb{t}_{ij,2}$. Hence, for any given $u_i \in [\varphi(p_{ij,1}), B_e]$, we have $\mathbb{t}_{ij,1} + g(B_e - u_i) \leq \mathbb{t}_{ij,2} + g(B_e - u_i)$, and thus, $\tau^{u_i}(p_{ij,1}) \leq \tau^{u_i}(p_{ij,2})$. □

Lemma 1 presents a useful result in the development of any path identification algorithm as it implies that the comparison between the travel time of different CS-paths would be sufficient at only one BCL, i.e. the fully charged battery level. Note, however, that this lemma does not cover values of $u_i < \varphi(p_{ij,1})$, as for such values path $p_{ij,1}$ could not be traversed. As a result, if $\varphi(p_{ij,2}) < \varphi(p_{ij,1})$, there are some u_i for which path $p_{ij,2}$ is preferred over path $p_{ij,1}$ as regards the travel time attribute.

Definition 4 Eligibility vector: *To every CS-path $p_{ij,\iota} \in \mathscr{p}_{ij}$ a 4-dimensional (4D) vector of attributes, corresponding to $E(p_{ij,\iota})\colon= [d(p_{ij,\iota}), \tau^{B_e}(p_{ij,\iota}), \pi(p_{ij,\iota}), \varphi(p_{ij,\iota})]$, is attributed that is called its eligibility vector.*

Definition 5 Eligibility vector dominance: *The eligibility vector $E(p_{ij,1})$ of a CS-path $p_{ij,1} \in \mathscr{p}_{ij}$ is said to dominate another eligibility vectors $E(p_{ij,2})$ of a CS-path $p_{ij,2} \in \mathscr{p}_{ij}$ (denoted by $E(p_{ij,1}) \preccurlyeq E(p_{ij,2})$) iff $d(p_{ij,1}) \leq d(p_{ij,2}), \tau^{B_e}(p_{ij,1}) \leq \tau^{B_e}(p_{ij,2}), \pi(p_{ij,1}) \leq \pi(p_{ij,2})$ and $\varphi(p_{ij,1}) \leq \varphi(p_{ij,2})$. Consequently, if $E(p_{ij,1})$ is not dominated by the eligibility vector of any other CS-path in $\mathscr{p}_{ij}$, it is said to be a non-dominated eligibility vector.*

Definition 6 An Eligible Path: *A CS-path $p_{ij,\iota} \in \mathscr{p}_{ij}$ with a non-dominated eligibility vector $E(p_{ij,\iota})$ is called an eligible path, and any other CS-path connecting required nodes $i, j \in N_r$ with dominated eligibility vector is called a redundant path. We denote the set of all eligible paths between a pair of required nodes $i, j \in N_r$ by $\mathfrak{p}_{ij}$.*

Theorem 1 *Suppose $\mathcal{G} = (N_r, \mathcal{P})$, with $\mathcal{P} = \bigcup_{i,j \in N_r} \mathcal{P}_{ij}$, and $\mathcal{P}_{ij} = \{\mathfrak{p}_{ij} \cup \epsilon_{ij} | i, j \in N_r\}$, is a multi-graph constructed from required nodes only and the direct edges* *and eligible paths between them. Then, any optimal solution found for an instance of the EVRPTW with CSs on G, could be found on $\mathcal{G}$.*

Proof. Suppose $\mathcal{S} = \mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_k$ indicates an optimal solution to an instance of the EVRPTW with CSs that is solved on G, where each $\mathcal{R}_{\mathscr{b}}, \forall \mathscr{b} \in \{1, \dots, k\}$, denotes an ECV route in $\mathcal{S}$. Each route $\mathcal{R}_{\mathscr{b}} \in \mathcal{S}$ could be viewed as a path that starts at the depot 0, visits a set of customers $C_{\mathscr{b}} \subseteq C$ (and possibly some intermediate CSs), and terminates at the depot $n + m + 1$. Since $\mathcal{S}$ is an optimal solution to the problem, each route $\mathcal{R}_{\mathscr{b}} \in \mathcal{S}$ is resource feasible and has a smaller total distance than any other path visiting the same sequence of customers in $C_{\mathscr{b}}$. Hence, we must prove that the elimination of redundant paths between each pair of required nodes i and j in $\mathcal{R}_{\mathscr{b}}$ has no implications regarding the resource-feasibility and optimality of $\mathcal{R}_{\mathscr{b}}$. To do so, suppose that a set of labels are maintained at each required node along $\mathcal{R}_{\mathscr{b}}$, each one corresponding to a partial path issued from 0 and containing the cumulative consumption level of each resource at the end of the corresponding partial path. The three resources we need to keep track of are distance, time and the BCL, with the latter two being constrained. Note that vehicle capacity is irrelevant as $\mathcal{R}_{\mathscr{b}}$ is already capacity feasible and elimination of CS-paths has no effect on its capacity feasibility. For both of the constrained resources time and BCL, at each required node $\mathscr{g}$ in $\mathcal{R}_{\mathscr{b}}$, resource windows could be defined, where for the time resource, we have customers' time-windows $[e_{\mathscr{g}}, l_{\mathscr{g}}]$, and for the BCL resource we have BCL-windows $\left[\min_{p_{\mathscr{g}\mathscr{h},i} \in \mathcal{P}_{\mathscr{g}\mathscr{h}}} \varphi(p_{\mathscr{g}\mathscr{h},i})\,, B_e\right]$. Note that the CS-path with $\min_{p_{\mathscr{g}\mathscr{h},i} \in \mathcal{P}_{\mathscr{g}\mathscr{h}}} \varphi(p_{\mathscr{g}\mathscr{h},i})$ already exists in $\mathfrak{p}_{\mathscr{g}\mathscr{h}}$, and elimination of redundant paths has had no effect on the resource windows. Hence, let the label $L_{\mathscr{g}} = [\mathbb{I}_{\mathscr{g}}^{dist}, \mathbb{I}_{\mathscr{g}}^{time}, u_{\mathscr{g}}]$ denote the consumption level of distance, time, and the BCL up to the required node $\mathscr{g}$ in $\mathcal{R}_{\mathscr{b}}$, respectively. Note that the larger values of $u_{\mathscr{g}}$ are desirable. The initial label is $L_0 = [0, e_0, B_e]$, and the extension of a label along a

CS-path $p_{\mathscr{g}\mathscr{h},i} \in \mathscr{p}_{\mathscr{g}\mathscr{h}}$ from $\mathscr{g}$ to $\mathscr{h}$ in $\mathcal{R}_{\mathscr{b}}$ is performed using the following *Resource Extension Functions (REFs)* (note that we just need to focus on CS-paths rather than direct edges as they are already present in both $\mathcal{G}$ and G):

$$\mathbb{l}_{\mathscr{h}}^{dist} = \mathbb{l}_{\mathscr{g}}^{dist} + d(p_{\mathscr{g}\mathscr{h},i}) \tag{4-21}$$

$$\mathbb{l}_{\mathscr{h}}^{time} = \mathbb{l}_{\mathscr{g}}^{time} + \mathbb{t}_{\mathscr{g}\mathscr{h},i} + g(B_e - u_{\mathscr{g}}) \tag{4-22}$$

$$u_{\mathscr{h}} = B_e - r_e d_{\ell\mathscr{h}} \tag{4-23}$$

Using these REFs we can prove that the labels extended by any redundant path along $\mathcal{R}_{\mathscr{b}}$ are always dominated by the labels extendend using eligible paths. To do so, assume $p_{\mathscr{g}\mathscr{h},1}$ is an eligible path that is present in $\mathcal{G}$, whereas $p_{\mathscr{g}\mathscr{h},2}$ is a redundant path discarded from $\mathcal{G}$. Since the eligibility of $p_{\mathscr{g}\mathscr{h},1}$ and the redundancy of $p_{\mathscr{g}\mathscr{h},2}$ implies that $E(p_{\mathscr{g}\mathscr{h},1}) \preccurlyeq E(p_{\mathscr{g}\mathscr{h},2})$ (Definition 5), it is easy to see that for any given $\mathbb{l}_{\mathscr{g}}^{dist}$, we always have $\mathbb{l}_{\mathscr{g}}^{dist} + d(p_{\mathscr{g}\mathscr{h},1}) \leq \mathbb{l}_{\mathscr{g}}^{dist} + d(p_{\mathscr{g}\mathscr{h},2})$, and also we always have $B_e - r_e d_{\ell\mathscr{h},1} \geq B_e - r_e d_{\ell\mathscr{h},,2}$. Moreover, based on Lemma 1, as long as $u_{\mathscr{g}} \geq \varphi(p_{\mathscr{g}\mathscr{h},1})$, the fact that $\tau^{B_e}(p_{\mathscr{g}\mathscr{h},1}) \leq \tau^{B_e}(p_{\mathscr{g}\mathscr{h},2})$ (Definition 5), implies that $\mathbb{t}_{\mathscr{g}\mathscr{h},1} + g(B_e - u_{\mathscr{g}}) \leq \mathbb{t}_{\mathscr{g}\mathscr{h},2} + g(B_e - u_{\mathscr{g}})$, and hence $\mathbb{l}_{\mathscr{g}}^{time} + \mathbb{t}_{\mathscr{g}\mathscr{h},1} + g(B_e - u_{\mathscr{g}}) \leq \mathbb{l}_{\mathscr{g}}^{time} + \mathbb{t}_{\mathscr{g}\mathscr{h},1} + g(B_e - u_{\mathscr{g}})$. On the other hand, for values of $u_{\mathscr{g}} < \varphi(p_{\mathscr{g}\mathscr{h},1})$ when path $p_{\mathscr{g}\mathscr{h},1}$ is infeasible, path $p_{\mathscr{g}\mathscr{h},2}$ is also definitely infeasible as $E(p_{\mathscr{g}\mathscr{h},1}) \preccurlyeq E(p_{\mathscr{g}\mathscr{h},2})$ implies $\varphi(p_{\mathscr{g}\mathscr{h},1}) \leq \varphi(p_{\mathscr{g}\mathscr{h},2})$. □

Corollary 1 *Any optimal solution found for an instance of the EVRP-SABS on G, could be found on $\mathcal{G}$.*

Proof. In the EVRP-SABS in addition to ECV routes, BSV routes are also planned on G. However, BSV routes only use ϵ_{ij} between every pair of required nodes i and j (as they are banned from visiting CSs), and these are already present in $\mathcal{G}$. □

Theorem 1. and Corollary 1 suggest that we can identify all eligible paths between required nodes a priori in a pre-processing stage and solve the problem on a reduced multi-graph. With this transformation, instead of solving the EVRPTW with CSs, one can solve a much easier VRPTW on a multi-graph.

The identification of the eligible paths could be done rather quickly as it must be done on a very small graph of only two customers and CSs; however, before introducing an algorithm for doing so, we exploit a property, that is observed in all existing VRPTW and EVRPTW benchmark test instances, to speed up the procedure by reducing the 4D eligibility vector $E(p_{ij,\boldsymbol{i}}) := [d(p_{ij,\boldsymbol{i}}), \tau^{B_e}(p_{ij,\boldsymbol{i}}), \pi(p_{ij,\boldsymbol{i}}), \varphi(p_{ij,\boldsymbol{i}})]$ to a 3D eligibility vector $\mathcal{E}(p_{ij,\boldsymbol{i}}) := [\tau^{B_e}(p_{ij,\boldsymbol{i}}), \pi(p_{ij,\boldsymbol{i}}), \varphi(p_{ij,\boldsymbol{i}})]$. This property corresponds to the linear dependency between travel time and distance for all arcs $(i, j) \in A$ in the form $t_{ij} = d_{ij}/\bar{v}$, where $\bar{v}$ could be viewed as the average speed in the network (in Solomon benchmark problems (1987), test instances developed by Schneider et al. (2014) and also used in Desaulniers et al. (2016), and all test instances considered in this paper, $\bar{v} = 1$). An important implication of this dependency assumption that we use is that $d_{\boldsymbol{i}\boldsymbol{j}} \leq d_{\boldsymbol{k}\ell} \Longleftrightarrow t_{\boldsymbol{i}\boldsymbol{j}} \leq t_{\boldsymbol{k}\ell}, \forall(\boldsymbol{i}, \boldsymbol{j}), (\boldsymbol{k}, \ell) \in A$. Hence, the following Lemma allows us to use $\mathcal{E}(p_{ij})$ instead of $E(p_{ij})$ as eligibility vector:

Lemma 2 *If $\mathcal{E}(p_{ij,1}) \preccurlyeq \mathcal{E}(p_{ij,2})$ for two CS-paths $p_{ij,1}, p_{ij,2} \in \boldsymbol{p}_{ij}$, then $E(p_{ij,1}) \preccurlyeq E(p_{ij,2})$.*

Proof. To prove the lemma, we need to show that if $\mathcal{E}(p_{ij,1}) \preccurlyeq \mathcal{E}(p_{ij,2})$, then $d(p_{ij,1}) \leq d(p_{ij,2})$. Suppose $p_{ij,1} = [(i, \boldsymbol{v}_1), \dots, (\boldsymbol{v}_{\ell 1}, j)], \boldsymbol{v}_1 .. \boldsymbol{v}_{\ell 1} \in R$, and $p_{ij,2} = [(i, \boldsymbol{v}_2), \dots, (\boldsymbol{v}_{\ell 2}, j)], \boldsymbol{v}_2 .. \boldsymbol{v}_{\ell 2} \in R$. We may write the distance and the travel time (at full BCL) of each of these CS-paths as: $d(p_{ij,\boldsymbol{i}}) = d_{i\boldsymbol{v}_{\boldsymbol{i}}} + d_{\boldsymbol{v}_{\boldsymbol{i}} \dots \boldsymbol{v}_{\ell \boldsymbol{i}}} + d_{\boldsymbol{v}_{\ell \boldsymbol{i}} j}$ and $\tau^{B_e}(p_{ij,\boldsymbol{i}}) = t_{i\boldsymbol{v}_{\boldsymbol{i}}} + t_{\boldsymbol{v}_{\boldsymbol{i}} \dots \boldsymbol{v}_{\ell \boldsymbol{i}}} + t_{\boldsymbol{v}_{\ell \boldsymbol{i}} j} + gr_e(d_{i\boldsymbol{v}_{\boldsymbol{i}}} + d_{\boldsymbol{v}_{\boldsymbol{i}} \dots \boldsymbol{v}_{\ell \boldsymbol{i}}}), \forall \boldsymbol{i} \in \{1{,}2\}$, respectively, where $d_{\boldsymbol{v}_{\boldsymbol{i}} \dots \boldsymbol{v}_{\ell \boldsymbol{i}}}$ and $t_{\boldsymbol{v}_{\boldsymbol{i}} \dots \boldsymbol{v}_{\ell \boldsymbol{i}}}$ denote the total distance and travel time of the arcs between the first and the last CS on $p_{ij,\boldsymbol{i}}$. If there is only one CS on $p_{ij,\boldsymbol{i}}$, then both

$d_{\mathscr{v}_i\ldots\mathscr{v}_{\ell i}}$ and $t_{\mathscr{v}_i\ldots\mathscr{v}_{\ell i}}$ are 0. We use a proof by contradiction, where we assume despite $\mathcal{E}(p_{ij,1}) \preccurlyeq \mathcal{E}(p_{ij,2})$, we have (4-24) below:

$$d(p_{ij,1}) > d(p_{ij,2}) \tag{4-24}$$

The condition that $\mathcal{E}(p_{ij,1}) \preccurlyeq \mathcal{E}(p_{ij,2})$, yields:

$$\begin{aligned} &t_{i\mathscr{v}_1} + t_{\mathscr{v}_1\ldots\mathscr{v}_{\ell 1}} + t_{\mathscr{v}_{\ell 1}j} + gr_e(d_{i\mathscr{v}_1} + d_{\mathscr{v}_1\ldots\mathscr{v}_{\ell 1}}) \\ &\qquad\qquad \leq t_{i\mathscr{v}_2} + t_{\mathscr{v}_2\ldots\mathscr{v}_{\ell 2}} + t_{\mathscr{v}_{\ell 2}j} + gr_e(d_{i\mathscr{v}_2} + d_{\mathscr{v}_2\ldots\mathscr{v}_{\ell 2}}) \end{aligned} \tag{4-25}$$

On the other hand, based on the linear dependency assumption between travel time and distance, following (4-24) we have:

$$t_{i\mathscr{v}_1} + t_{\mathscr{v}_1\ldots\mathscr{v}_{\ell 1}} + t_{\mathscr{v}_{\ell 1}j} > t_{i\mathscr{v}_2} + t_{\mathscr{v}_2\ldots\mathscr{v}_{\ell 2}} + t_{\mathscr{v}_{\ell 2}j} \tag{4-26}$$

Considering (4-25) and (4-26), for (4-25) to hold true, we must have:

$$gr_e(d_{i\mathscr{v}_1} + d_{\mathscr{v}_1\ldots\mathscr{v}_{\ell 1}}) \leq gr_e(d_{i\mathscr{v}_2} + d_{\mathscr{v}_2\ldots\mathscr{v}_{\ell 2}}) \tag{4-27}$$

Which means:

$$d_{i\mathscr{v}_1} + d_{\mathscr{v}_1\ldots\mathscr{v}_{\ell 1}} \leq d_{i\mathscr{v}_2} + d_{\mathscr{v}_2\ldots\mathscr{v}_{\ell 2}} \tag{4-28}$$

At the same time, as another implication of $\mathcal{E}(p_{ij,1}) \preccurlyeq \mathcal{E}(p_{ij,2})$, we know $\varphi(p_{ij,1}) \leq \varphi(p_{ij,2})$, which means $r_e d_{\mathscr{v}_{\ell 1}j} \leq r_e d_{\mathscr{v}_{\ell 2}j}$, and hence:

$$d_{\mathscr{v}_{\ell 1}j} \leq d_{\mathscr{v}_{\ell 2}j} \tag{4-29}$$

The combination of (4-28) and (4-29) results in $d(p_{ij,1}) \leq d(p_{ij,2})$, which is in contradiction with (4-24). □

Based on these results we can propose an implementation of the EPIP which in practice must search for tri-objective shortest paths between a pair of required nodes on a very small auxiliary graph of the given origin and destination and CSs only. However, using an intuitive rule it is still possible to speed up the implementation further by searching for bi-criterion shortest paths instead. Indeed, we have extensively observed in our experiments that almost always when we only look for CS-paths with non-dominated $[\tau^{B_e}(p_{ij}), \pi(p_{ij})]$, the CS-path with minimum $\varphi(p_{ij})$ already exists in the returned set, and this means we do not need to carry out any further search, as it is provable that any other path will have a dominated eligibility vector. In case this is not satisfied, it is very simple to find the ‘next’ CS-path with minimum $\varphi(p_{ij})$ iteratively until we see it in the set.

Hence, the implementation of the EPIP is given in Algorithm 4-1. This algorithm finds the set of eligible paths between a pair of required nodes $\mathcal{O}, \mathcal{D} \in N_r$ using a modified extension of the label setting algorithm proposed by Martins (1984) for the multi-criteria shortest path problem (we adapt the implementation proposed by Ehrgott (2005)). In this algorithm $G' = (N', A')$ is an auxiliary graph where $N' = \{\{\mathcal{O}, \mathcal{D}\} \cup R\}$, and $A' = \{(i, j)|i, j \in N', i \neq j\}$. A label L of a node $i \in N'$ is denoted using a tuple $L = [\tau_i, \pi_i, u_i, J, \ell_J, k]$, where τ_i stores the travel time attribute of the path represented by the label up to node i, π_i is its EBDL attribute, u_i is the BCL at the node i, J represents the node from which the label was obtained, ℓ_J indicates the identifier of the label in the list of labels at node J from which L was obtained, and k is the identifier of the current label in the list of labels at node i. Note that domination rules used in lines 13 and 14 of the algorithm, are based on the first two components of the label, i.e. τ_i and π_i. The last while loop of the algorithm (lines 20 to 24) is only executed if the path with minimum MR-BCL does not exist already in $\mathfrak{p}_{\mathcal{O}\mathcal{D}}$. It is worth mentioning that any shortest path algorithm could be used in line 20 for the identification of the minimum MR-BCL path.

Algorithm 4-1 The EPIP

```
**Input** $G'$, origin node $\mathcal{O} \in N'$, destination node $\mathcal{D} \in N'$, $[e_{\mathcal{O}}, l_{\mathcal{O}}], s_{\mathcal{O}}, [e_{\mathcal{D}}, l_{\mathcal{D}}]$ $d_{ij}, t_{ij}, \forall (i,j) \in A'$, $B_e, r_e, g$
Initialise $\mathfrak{p}_{\mathcal{OD}} = \{\}, \mathcal{TL} = \{\}$, and $\mathcal{PL} = \{\}$.
Create label $L_{\mathcal{O}} = [e_{\mathcal{O}}, 0, B_e, 0{,}0{,}1]$ at node $\mathcal{O}$ and let $\mathcal{TL}$:= { $L_{\mathcal{O}}$}.
while $\mathcal{TL} \neq \emptyset$ do
    Let label $L_i = [\tau_i, \pi_i, u_i, \boldsymbol{\jmath}, \ell_{\boldsymbol{\jmath}}, \boldsymbol{k}]$ of node $i$ be the lexicographically smallest label in $\mathcal{TL}$.
    Remove $L_i$ from $\mathcal{TL}$ and add it to $\mathcal{PL}$.
    for all $j \in N'$ such that $(i,j) \in A'$ do
        if $i = \mathcal{O}$ then $\tau_j = \tau_i + t_{ij}$ else $\tau_j = \tau_i + g(B_e - u_i) + t_{ij}$ end if
        if $j = \mathcal{D}$ then $\pi_j = \pi_i + r_e d_{ij}$ else $\pi_j = 0$ end if
        $u_j = B_e - r_e d_{ij}$
        if  $e_{\mathcal{O}} + s_{\mathcal{O}} + \tau_j \leq l_{\mathcal{D}}$ and $\pi_j \leq B_e$ then
            Create label $L' = [\tau_j, \pi_j, u_j, i, \ell_i, \boldsymbol{t}]$ as the next label ($\boldsymbol{t}$th label) at node $j$ and add it to $\mathcal{TL}$.
            Delete all temporary labels of node $j$ dominated by $L'$.
            Delete $L'$ if it is dominated by another label of node $j$.
        end if
    end for
end while
Use the predecessor labels in $\mathcal{PL}$ to recover all efficient paths and add them to $\mathfrak{p}_{\mathcal{OD}}$.
$\vec{\mathfrak{a}} \leftarrow$ the shortest outgoing arc from $\mathcal{O}$
while $\vec{\mathfrak{a}}$ is not the first arc in any of the paths in $\mathfrak{p}_{\mathcal{OD}}$ do
    Find the CS-path $\mathrm{p}_{\mathcal{OD}}$ with minimum $\varphi(\mathrm{p}_{\mathcal{OD}})$ and add it to $\mathfrak{p}_{\mathcal{OD}}$
    Remove $\vec{\mathfrak{a}}$ from $G'$
    $\vec{\mathfrak{a}} \leftarrow$ the shortest outgoing arc from $\mathcal{O}$
end while
return $\mathfrak{p}_{\mathcal{OD}}$
```

In the next sub-section, we show how these results could be used to develop a strengthened formulation of the EVRPTW with CSs and the EVRP-SABS. It is worth mentioning that all the results presented in this section could be simply generalised to the case of G-VRP, as well.

4.4.1 An EPIP-based formulation of the problem

The alternative EPIP-based formulation is defined on the multi-graph $\mathcal{G} = (N_r, \mathcal{P})$ (refer to Theorem 1). Let us place ϵ_{ij} on top of the eligible paths in $\mathcal{P}_{ij}, \forall i, j \in N_r$, and refer to each member of the set $\mathcal{P}_{ij}$ by (i, j, p), where $p \in \{1, .., |\mathcal{P}_{ij}|\}$. Hence, $(i, j, 1)$ is always the direct edge (i, j) in G. As a generalisation of the closed form expression for the BCL-dependent travel time attribute, we can use the expression of the form $\alpha_{ijp} u_i + \beta_{ijp}$ for each $(i, j, p) \in \mathcal{P}_{ij}$, where parameters α_{ijp} and β_{ijp} are parameters that could be computed as follows and used as model input:

$$\alpha_{ijp} = \begin{cases} 0, & p = 1 \\ -g, & otherwise \end{cases} \tag{4-30}$$

$$\beta_{ijp} = \begin{cases} t_{ij}, & p = 1 \\ \mathbb{t} + \ gB_e, & otherwise \end{cases} \tag{4-31}$$

We also define here another BCL-dependent attribute for each path, called the Charge Gained and Gone (CGG) attribute. CGG takes into account the BCL at the origin of the path and any refuelling over the path, and in practice denotes the difference between the BCL at the origin node and the BCL upon the arrival at the destination using an expression of the form $\gamma_{ijp} u_i + \delta_{ijp}$ for each $(i, j, p) \in \mathcal{P}_{ij}$, where parameters γ_{ijp} and δ_{ijp} are parameters that could be pre-computed as follows and used as model input:

$$\gamma_{ijp} = \begin{cases} 0, & p = 1 \\ 1, & otherwise \end{cases} \tag{4-32}$$

$$\delta_{ijp} = \begin{cases} r_e d_{ij}, & p = 1 \\ r_e d_{\ell j} - B_e, & otherwise \end{cases} \tag{4-33}$$

Note that CGG is not essentially non-negative.

Now recycling some of the notations used in section 3.3 of the paper, we redefine two of the previous decision variables to use in the EPIP-based formulation as follows:

- x_{ijp}: Binary variable equal to 1 iff path $(i,j,p) \in \mathcal{P}$, is traversed by an ECV.
- z_{ij1}: Binary variable equal to 1 iff path $(i,j,1) \in \mathcal{P}$ is traversed by a BSV.

The extension of the formulation in (4-1)-(4-20) to the alternative EPIP-based formulation using these variables is presented in (4-34)-(4-52) below. Note also that it is easy to deduce an EPIP-based formulation for the EVRPTW and the GVRP from the proposed formulation. We use an implementation of the EPIP-based EVRPTW against an exact CG-based solution algorithm for the problem in the 'computational results' section of the paper to demonstrate the added value of the proposed formulation.

$$Min \sum_{(i,j,p)\in\mathcal{P}} d_{ijp}x_{ijp} + \sum_{(i,j,1)\in\mathcal{P}} d_{ij1}z_{ij1} \tag{4-34}$$

Subject to:

$$\sum_{j\in N_3}\sum_{p\in\mathcal{P}_{ij}} x_{ijp} = 1, \qquad \forall i \in C \tag{4-35}$$

$$\sum_{j\in N_3} z_{ij1} \le 1, \qquad \forall i \in C \tag{4-36}$$

$$\sum_{i\in N_3}\sum_{p\in\mathcal{P}_{ji}} x_{jip} - \sum_{i\in N_2}\sum_{p\in\mathcal{P}_{ij}} x_{ijp} = 0, \qquad \forall j \in C \tag{4-37}$$

$$\sum_{i\in N_3} z_{ji1} - \sum_{i\in N_2} z_{ij1} = 0, \qquad \forall j \in C \tag{4-38}$$

$$y_i + \alpha_{ijp}u_i + (\beta_{ijp}+s_i)x_{ijp} + \mathscr{s}\sum_{j\in N_2} z_{ji1} - (l_0+\mathscr{s})(1-x_{ijp}) \le y_j, \qquad \forall (i,j,p) \in \mathcal{P} \tag{4-39}$$

$$w_i \le y_i + s_i, \qquad \forall i \in N_r \tag{4-40}$$

$$y_i + \alpha_{ij1}v_i + (\beta_{ij1}+\mathscr{s}+s_i)z_{ij1} - l_0(1-z_{ij1}) \le w_j, \qquad \forall (i,j,1) \in \mathcal{P} \tag{4-41}$$

$$e_i \le y_i \le l_i, \qquad \forall i \in N_r \tag{4-42}$$

$$0 \leq f_j \leq f_i - (q_i \sum_{p \in \mathcal{P}_{ij}} x_{ijp}) + Q_e(1 - \sum_{p \in \mathcal{P}_{ij}} x_{ijp}), \qquad \forall i \in N_2, j \in N_3, i \neq j \tag{4-43}$$

$$0 \leq f_0 \leq Q_e \tag{4-44}$$

$$0 \leq h_j \leq h_i - z_{ij1} + Q_b(1 - z_{ij1}), \qquad \forall (i,j,1) \in \mathcal{P} \tag{4-45}$$

$$0 \leq h_0 \leq Q_b \tag{4-46}$$

$$0 \leq u_j \leq u_i - \gamma_{ijp} u_i - \delta_{ijp} x_{ijp} + (B_e \sum_{\jmath \in N_2 : j \neq n+m+1} z_{\jmath j1}) + B_e(1 - x_{ijp}), \qquad \forall (i,j,p) \in \mathcal{P} \tag{4-47}$$

$$\varphi_{0jp} x_{0jp} \leq u_0 \leq B_e, \qquad \forall (0,j,p) \in \mathcal{P} \tag{4-48}$$

$$\sum_{j \in N_m} \sum_{p \in \mathcal{P}_{ij}} \varphi_{ijp} x_{ijp} \leq u_i \leq B_e, \qquad \forall i \in C \tag{4-49}$$

$$0 \leq v_j \leq v_i - \gamma_{ij1} u_i - \delta_{ij1} z_{ij1} + B_b(1 - z_{ij1}), \qquad \forall (i,j,1) \in \mathcal{P} \tag{4-50}$$

$$\varphi_{0j1} z_{0j1} \leq v_0 \leq B_b, \qquad \forall (0,j,1) \in \mathcal{P} \tag{4-51}$$

$$\sum_{j \in N_m} \varphi_{ij1} z_{ij1} \leq v_i \leq B_b, \qquad \forall i \in C \tag{4-52}$$

While the interpretation of most of the constraints is equivalent to those in (4-1)-(4-20), constraints (4-48) and (4-52) respectively denote that the BCL of an ECV and a BSV at departure from the origin of a selected path is larger than the MR-BCL of the corresponding path.

4.5 The 2S-MatHeu algorithm for the EVRP-SABS

The EVRP-SABS is very difficult to solve to optimality in case of realistically sized test instances and hence development of a tailored heuristic solution algorithm is important to tackle practical problem sizes within a reasonable computational time. In this section, we propose a 2S-MatHeu solution algorithm for the EVRP-SABS that is based upon the proposed EPIP. This algorithm decomposes the problem into two easier to solve problems and deals with each one in two separate stages. In the first stage, the algorithm tries to find a feasible solution to the problem without requesting any swapping services and by just visiting the available CSs. If such solution exists, the algorithm terminates and there is no need to invoke the second stage solver. Otherwise, owing to the EPIP and its resultant graph $\mathcal{G}$, the first stage solver still returns a solution specifying exactly at which customer node(s) and at what time a BSV should be present for a swapping service. With this information returned by the first stage solver, the second stage solver needs only to solve a very small and simple VRPTW to route the BSV(s) that could be even solved to optimality.

It is important to note that an independent decomposition of the two levels of routing that must be determined for the ECVs and the BSVs can create issues regarding the interdependence problem that is inherent to problems with spatiotemporal synchronisation requirements. To avoid this, the *'communication'* between the two levels must be preserved through an appropriate *'medium'*. In this study we employ and examine with the intuition that a smaller number of swapping services requested by ECVs in the first level would potentially lead to a smaller number and total distance of the BSVs in the second level. Hence, the first stage problem will minimise the total number of swapping services requested and the total distance of the ECVs in a lexicographical order.

The details of each of these stages and the associated algorithms are discussed in the sequel.

4.5.1 The first stage problem: the EVRPTW with CSs

The objective of the first stage solver is to find a feasible solution to the problem without requesting any swapping services by just visiting the available CSs, and this corresponds to solving an EVRPTW with CSs. In the event that no feasible solution to the problem in hand in this stage could be found without requesting battery swaps, the algorithm is still instructed to return a solution enhanced by information regarding the minimum number of battery swapping required at the customer nodes visited by the ECV routes. This is made possible by placing a tailored EPIP-based DP at the heart of the first stage solver, which is described in the sequel.

As a result of the application of the proposed EPIP on a given instance of EVRP-SABS in the pre-processing phase, the first stage problem of the EVRPTW with CSs can be dealt with as a VRPTW on the multi-graph $\mathcal{G}$ (where all CSs are eliminated). Not only this problem transformation allows an exact evaluation of any ECV route that requires visits to CSs, but also, with few algorithmic enhancements that are shortly described, it is possible to derive very useful information on the need for BSV visits and send this information to the second stage solver. An optimal evaluation of an ECV route on $\mathcal{G}$, on the other hand, corresponds to solving a Fixed Sequence Arc Selection Problem (FSASP) (Garaix et al., 2010) to optimality. Garaix et al. (2010) propose a DP to solve FSASP, which can be extended and used to evaluate an ECV route on $\mathcal{G}$. However, if there is no feasible evaluation for the ECV without requesting a battery swap, this DP can only return null. We enrich this DP by also exploring the possibility of battery swaps when extending labels.

The proposed EPIP-based DP is hence presented in Algorithm 4-2. This algorithm takes the multi-graph $\mathcal{G}$, an ECV route (a sequence of customer visits) $\mathcal{R} = \{c_0, c_1, \dots, c_{\mathcal{N}}\}$ (where c_0 and $c_{\mathcal{N}}$ are the depot), and the time-windows and service times of the customers on the route in its input (line 1), and returns an optimal evaluation (deduced from labels at the destination node $\mathcal{L}_{\mathcal{N}}$) of the given route such that: (i) the total number of swapping services needed, and (ii) the total distance of the routes are minimised lexicographically. As discussed before, priority is given to the minimisation of the number of swapping services required as it is expected that a larger number of requested swapping services should lead to a larger number of BSVs needed and BSV distance travelled in the second stage. Along with $\mathcal{L}_{\mathcal{N}}$, the algorithm returns also information about the customers that require a swapping service in $\mathcal{C}_{\mathcal{N}}$, and the time at which these customers need the service to be available in $\mathcal{T}_{\mathcal{N}}$ (line 34). To this end, the algorithm retains and extends a set of labels $\mathcal{L}_{c_i}$, $\mathcal{C}_{c_i}$, and $\mathcal{T}_{c_i}$ at each node c_i along $\mathcal{R}$. Each label $\ell \in \mathcal{L}_{c_i}$ is a tuple of length 4, where ℓ_1 stores the accumulated distance, ℓ_2 stores the accumulated travel time, ℓ_3 stores the available BCL, and ℓ_4 stores the total number of swaps requested up to the current node in $\mathcal{R}$. Each label $\mathscr{g} \in \mathcal{C}_{c_i}$ and $\mathscr{f} \in \mathcal{T}_{c_i}$, on the other hand, is an open-ended list of customers requiring swaps and their requested service time, respectively. The first set of labels at c_0 is initiated in line 2 of the algorithm and it is extended in lines 3 to 32 of the algorithm. The distinctive feature of the proposed DP that particularly leads to extra information regarding the need to swapping services corresponds to lines 25 to 30 of the algorithm, where the restriction on the available BCL is lifted and it is assumed that the ECV is ready to depart the node using a fully charged battery as a result of a potential battery swapping service by a BSV.

The working of the proposed EPIP-based DP is illustrated using a small example in Figure 4-4. Figure 4-4.a shows an ECV route $\mathcal{R} = \{c_0, c_1, c_2, c_3\}$ visiting 2 customers c_1 and c_2 on G. Identified by the EPIP, there is only one eligible CS-path between consecutive visits c_1 and c_2 in this example. The number above each arrow in this figure denotes the distance (=travel time) of each arc, and we assume that the service time at all customers on the route is equal to zero, and while there is no time-windows on c_1, there is a time-window on c_2 corresponding to [0,11]. Moreover, suppose that $B_e = 10, r_e = 1, \mathscr{s} = 1$ and $g = 1$. Figure 4-4.b shows the resulting multi-graph $\mathcal{G}$, and all parameters related to the attributes of paths A to D in this multi-graph are computed based on the previous expressions and presented in Table 4-1.

Algorithm 4-2 The EPIP-based DP

```
Input $\mathcal{G}$, $\mathcal{R}$, $B_e$, $r_e$, and $e_{\mathscr{c}_i}, l_{\mathscr{c}_i}, s_{\mathscr{c}_i} \forall \mathscr{c}_i \in \mathcal{R}$
Initialise $\mathcal{L}_0 = \{0,0, B_e, 0\} = \{\}, \mathcal{T}_0 = \{\}$, and $\mathcal{C}_0 = \{\}$;
for $\mathscr{q} = 0$ to $\mathcal{N} - 1$ do
    foreach: label $\ell \in \mathcal{L}_{\mathscr{q}}$ do
        for $\mathscr{p} = 1$ to $|\mathcal{P}_{\mathscr{c}_{\mathscr{q}}\mathscr{c}_{\mathscr{q}+1}}|$ do
            if $\ell_3 \geq \varphi(\mathcal{P}_{\mathscr{c}_{\mathscr{q}}\mathscr{c}_{\mathscr{q}+1},\mathscr{p}})$ then
                $\mathbb{b} = \max\{\ell_2 + s_{\mathscr{c}_{\mathscr{q}}} + \alpha_{\mathscr{c}_{\mathscr{q}}\mathscr{c}_{\mathscr{q}+1},\mathscr{p}}\ell_3 + \beta_{\mathscr{c}_{\mathscr{q}}\mathscr{c}_{\mathscr{q}+1},\mathscr{p}}, e_{\mathscr{c}_{\mathscr{q}+1}}\}$;
                if $\mathbb{b} \leq l_{\mathscr{c}_{\mathscr{q}+1}}$ then
                    dominated := false, $\mathbb{a} = \ell_1 + d_{\mathscr{c}_{\mathscr{q}}\mathscr{c}_{\mathscr{q}+1},\mathscr{p}}$
                    $\mathbb{c} = \ell_3 - \gamma_{\mathscr{c}_{\mathscr{q}}\mathscr{c}_{\mathscr{q}+1},\mathscr{p}}\ell_3 - \delta_{\mathscr{c}_{\mathscr{q}}\mathscr{c}_{\mathscr{q}+1},\mathscr{p}}$, $\mathbb{d} = \ell_4$,
                    $\mathbb{e} = \mathscr{f}, \mathbb{f} = \mathscr{g}$,    // $\mathscr{f}$ and $\mathscr{g}$ are the $\ell$th lablels in $\mathcal{T}_{\mathscr{q}}$ and $\mathcal{C}_{\mathscr{q}}$,
                    $\ell' = \{\mathbb{a}, \mathbb{b}, \mathbb{c}, \mathbb{d}, \}$;
                    foreach: label $\ell'' \in \mathcal{L}_{\mathscr{q}+1}$ do
                        if $\ell'_1 \leq \ell''_1$ and $\ell'_2 \leq \ell''_2$ and $\ell'_3 \geq \ell''_3$ and $\ell'_4 \leq \ell''_4$ then
                            $\mathcal{L}_{\mathscr{q}+1} := \mathcal{L}_{\mathscr{q}+1} \backslash \{\ell''\}, \mathcal{T}_{\mathscr{q}+1} := \mathcal{T}_{\mathscr{q}+1} \backslash \{\mathscr{f}''\}, \mathcal{C}_{\mathscr{q}+1} := \mathcal{C}_{\mathscr{q}+1} \backslash \{\mathscr{g}''\}$;
                        elseif $\ell''_1 \leq \ell'_1$ and $\ell''_2 \leq \ell'_2$ and $\ell''_3 \geq \ell'_3$ and $\ell''_4 \leq \ell'_4$ then
                            dominated := true, break;
                        end if
                    end for
                    if dominated = false then
                        $\mathcal{L}_{\mathscr{q}+1} := \mathcal{L}_{\mathscr{q}+1} \cup \{\ell'\}, \mathcal{T}_{\mathscr{q}+1} := \mathcal{T}_{\mathscr{q}+1} \cup \{\mathbb{f}\}, \mathcal{C}_{\mathscr{q}+1} := \mathcal{C}_{\mathscr{q}+1} \cup \{\mathbb{g}\}$ ;
                    end if
                end if
            end if
            if $B_e \geq \varphi(\mathcal{P}_{\mathscr{c}_{\mathscr{q}}\mathscr{c}_{\mathscr{q}+1},\mathscr{p}})$ then
```

```
26              Repeat lines 7 to 22 with following modifications:
27              In line 7: $\mathbb{b} = \max\{\ell_2 + s_{\mathscr{c}_{\mathscr{q}}} + \mathscr{s} + \alpha_{\mathscr{c}_{\mathscr{q}}\mathscr{c}_{\mathscr{q}+1},\mathscr{p}} B_e + \beta_{\mathscr{c}_{\mathscr{q}}\mathscr{c}_{\mathscr{q}+1},\mathscr{p}}, e_{\mathscr{c}_{\mathscr{q}+1}}\}$
28              In line 10: $\mathbb{c} = B_e - \gamma_{\mathscr{c}_{\mathscr{q}}\mathscr{c}_{\mathscr{q}+1},\mathscr{p}} B_e - \delta_{\mathscr{c}_{\mathscr{q}}\mathscr{c}_{\mathscr{q}+1},\mathscr{p}},$
29              In line 11: $,\mathbb{e} = \mathscr{f} \cup \{\ell_2 + s_{\mathscr{c}_{\mathscr{q}}}\},\ \mathbb{f} = \mathscr{g} \cup \{\mathscr{c}_{\mathscr{q}}\};$
30           end if
31        end for
32     end for
33  end for
34  return $\mathcal{L}_{\mathcal{N}}, \mathcal{T}_{\mathcal{N}}$, and $\mathcal{C}_{\mathcal{N}}$.
```

Table 4-1 Parameters related to the attributes of the paths in Figure 4-4

Paths	d_{ijp}	φ_{ijp}	α_{ijp}	β_{ijp}	γ_{ijp}	δ_{ijp}
A	2	2	0	2	0	2
B	4	4	0	4	0	4
C	5	2	-1	17	1	-7
D	8	8	0	8	0	8

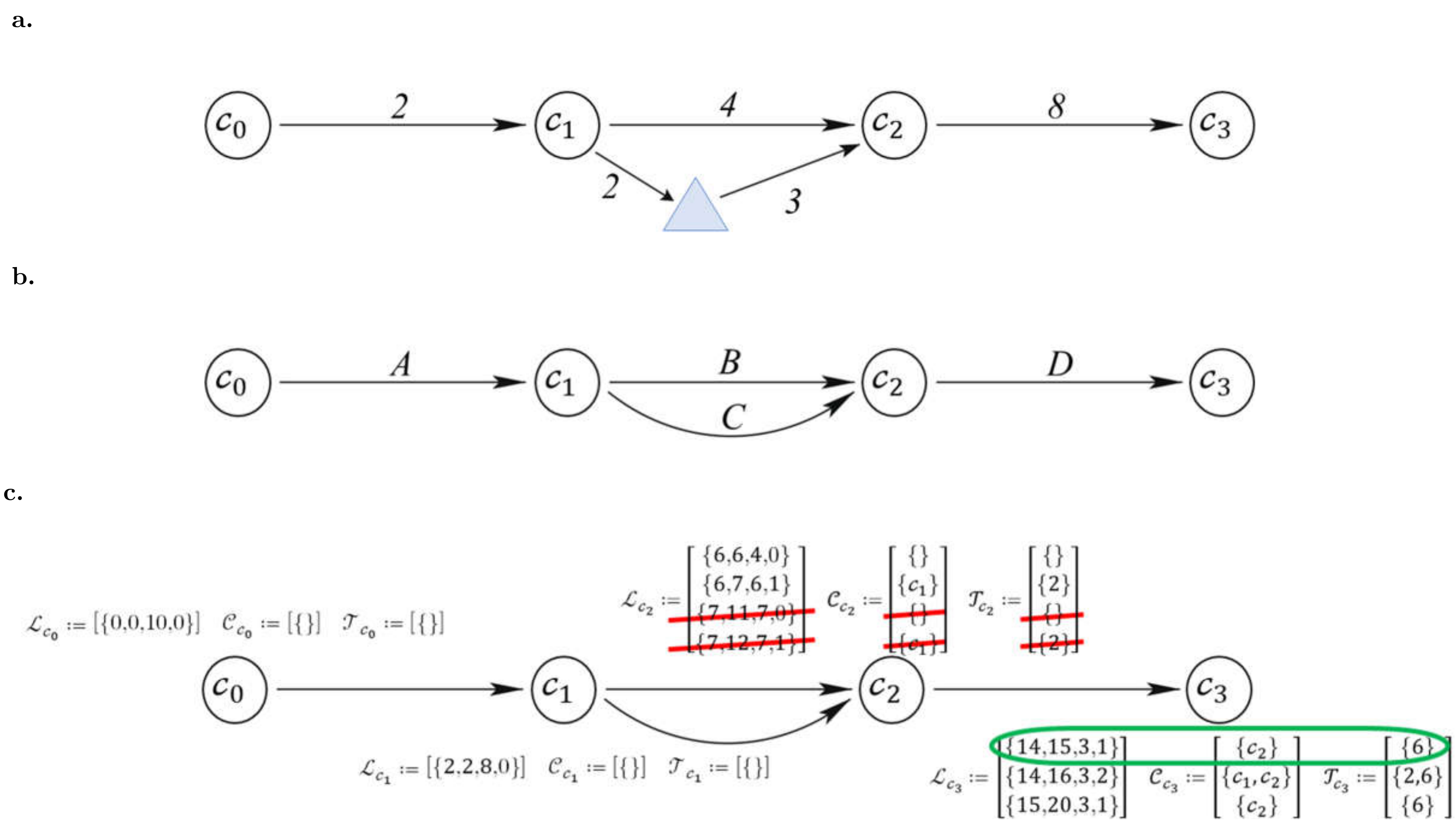

a.
c_0
2
c_1
4
c_2
8
c_3
2
3
b.
c_0
A
c_1
B
c_2
D
c_3
C
c.
$\mathcal{L}_{c_0} := [\{0,0,10,0\}]$ $\mathcal{C}_{c_0} := [\{\}]$ $\mathcal{T}_{c_0} := [\{\}]$
$\mathcal{L}_{c_1} := [\{2,2,8,0\}]$ $\mathcal{C}_{c_1} := [\{\}]$ $\mathcal{T}_{c_1} := [\{\}]$
$\mathcal{L}_{c_2} := \begin{bmatrix} \{6,6,4,0\} \\ \{6,7,6,1\} \\ \cancel{\{7,11,7,0\}} \\ \cancel{\{7,12,7,1\}} \end{bmatrix}$ $\mathcal{C}_{c_2} := \begin{bmatrix} \{\} \\ \{c_1\} \\ \cancel{\{\}} \\ \cancel{\{c_1\}} \end{bmatrix}$ $\mathcal{T}_{c_2} := \begin{bmatrix} \{\} \\ \{2\} \\ \cancel{\{\}} \\ \cancel{\{2\}} \end{bmatrix}$
$\mathcal{L}_{c_3} := \begin{bmatrix} \{14,15,3,1\} \\ \{14,16,3,2\} \\ \{15,20,3,1\} \end{bmatrix}$ $\mathcal{C}_{c_3} := \begin{bmatrix} \{c_2\} \\ \{c_1,c_2\} \\ \{c_2\} \end{bmatrix}$ $\mathcal{T}_{c_3} := \begin{bmatrix} \{6\} \\ \{2,6\} \\ \{6\} \end{bmatrix}$
c_0
c_1
c_2
c_3

Figure 4-4 An illustrative example for the EPIP-based DP

Figure 4-4.c shows the extension of the labels along the given ECV route. At the final node, we must choose between the first and the last label at c_3, i.e. $\{14{,}15, 3, 1\}$ and $\{15, 20, 3, 1\}$ resulted respectively from resource feasible paths A-B-D and A-C-D, with a requested swapping service only at c_2 at time 6. Clearly, A-B-D has a smaller total distance and hence is selected. Note that despite its shorter distance, the middle label $\{14{,}16, 3, 2\}$ is not preferred over the last label due to its larger number of swapping services required.

It is worth noting that while in this study we tend to solve the core problem of FSASP to optimality, to speed up the overall algorithm, several alternative approaches could be used which will be left as perspectives here. One such approach is to use a fast heuristic algorithm like the one proposed in Lai et al. (2016) to solve the iteratively emerging FSASP problem approximately for newly generated solutions, and only run the exact algorithm at local optima. It is also possible to introduce a built-in memory with limited size in the algorithm to store frequently evaluated routes (or partial paths with their labels) in a hash table with a key. With this memory, before evaluating any route the algorithm will check with the hash table if an evaluation could be directly obtained without solving the incurred FSASP.

Placing the proposed EPIP-based DP at the heart of the first stage solver as the routine for solution evaluation, any of the many available solution algorithms for the VRPTW could be used to deal with EVRPTW with CSs as a VRPTW on a multi-graph. We use an *Intensified LNS (ILNS)* algorithm for this purpose. The LNS developed by Shaw (1998), and its adaptive extension, i.e. ALNS, developed by Ropke and Pisinger (2006), is a conceptually simple metaheuristic and has proven successful in solving different variants of routing problems, particularly the VRPTW. The LNS is based on large rearrangements in a current solution by applying several removal and re-insertion heuristics and hence moving from one area within the feasible region to another using rather large steps. For brevity, we avoid elaborating on the details of the LNS and ALNS here and refer the reader to the original studies of Shaw (1998) and Ropke and Pisinger (2006) for that purpose.

In order to achieve a better exploitation capability, we equip further the proposed algorithm with an intensification procedure based on a Simulated Annealing (SA) metaheuristic with a new neighbourhood exploration strategy. This algorithm is invoked upon finding local optima to seek the possibility of going downhill further.

An overview of the proposed ILNS algorithm is given in Algorithm 4-3. In the first step of the proposed algorithm (line 1) a feasible solution is generated using a simple heuristic. This heuristic puts all customers into a non-routed pocket and initiates an empty ECV route in the beginning, and then in each iteration extracts a customer from the pocket and tries to insert it at its best location in the current route, or if impossible, in a new route until the pocket is empty. Following the generation of the initial solution, the proposed ILNS takes a fixed number of iterations ($maxIter_{LNS}$) to return a near optimal solution S_{best}. In each iteration, a removal heuristic is selected from a set of available removal heuristics and is applied on the current solution S_{curr} to remove a certain number of customers from the routes in the solution (line 6). We are using three removal heuristics: (i) Shaw removal (Shaw, 1998), (ii) random removal (Ropke & Pisinger, 2006), and (iii) worst removal (Ropke & Pisinger, 2006), all with equal chances to be selected. For all these removal heuristics, the number of customers to remove is determined by selecting a random integer in the interval $[4, n\rho_{rem}]$, where $\rho_{rem} \in [0,1]$ is a user defined parameter. In addition to this parameter, there is a $\rho_{shaw} \in \mathbb{R}_+$ parameter for the Shaw removal which controls determinism in the relatedness function (see Shaw, 1998), and there is a $\rho_{worst} \in \mathbb{R}_+$ parameter for the worst removal that controls the degree of randomisation (see Ropke and Pisinger, 2006).

The destroyed solution S_{new} after applying the selected removal heuristic, and the pocket containing removed customers are then submitted to a selected re-insertion algorithm to repair S_{new} and retrieve a possibly new S_{curr} (line 7). We use two insertion heuristics adopted from Ropke and Pisinger (2006) for this purpose: (i) regret-2 heuristic, and (ii) regret-2 heuristic with noise (see Ropke and Pisinger, 2006 for details). There is only one parameter here (i.e. η) associated with the second heuristic to control the amount of noise.

Following the application of the destroy and repair mechanisms in lines 6 and 7 of the algorithm, the intensification procedure is invoked if the new resulting S_{curr} is 'better' than the existing S_{best} (line 8). It is important to note that in our algorithm a solution with smaller number of battery swaps required is always prefered over a solution with smaller total distance but larger number of swaps required. Hence, whenever we compare the cost of two solutions this is observed. After updating S_{best} (line 9), while S_{best} can be improved, the intensification procedure is repeatedly applied on S_{best}.

The structure of the intensification procedure is similar to the successful SA algorithm proposed by Bent and Van Hentenryck (2004) for VRPTW, and mainly differs in its neighbourhood exploration strategy, which is indeed the special feature of their SA. In each iteration of their SA algorithm, Bent and Van Hentenryck (2004) choose randomly a move operator and a customer, and then consider all the possible moves for this customer using the selected operator to see if any improvement could be found. While we use the same 5 well- known local search operators that Bent and Van Hentenryck (2004) use in their study; i.e. 2-opt, Or-opt, Relocation, Swap, and Crossover, we explore a wider sub-neighbourhood by selecting λn customers instead of only one customer, where the rate $\lambda \in [0,1]$ is a user defined parameter. The closer is the selected λ to 1, the wider will be the explored sub-neighbourhood, and thus the better might be the ultimate solution, but also the slower would be the overall algorithm. Note that the SA algorithm used in the intensification procedure requires 5 other input parameters corresponding to $timeLimit_{INT}$, $\mathcal{C}_{INT}$, $tempLimit_{INT}$, $startingTemp_{INT}$, and $maxIter_{INT}$ (see Bent and Van Hentenryck, 2004 for details).

In lines 15 to 19 of the algorithm, S_{curr} is compared with S_{candid}, and if S_{curr} cannot improve S_{candid}, a SA-wise acceptance criterion is used to examine if S_{candid} could be updated. Note that the temperature $temp_{LNS}$ in the algorithm starts out at $startingTemp_{LNS}$ (line 3) and is decreased at the end of every iteration using the expression $temp_{LNS} \leftarrow temp_{LNS} \times \mathcal{C}_{LNS}$ (line 20), where $0 < \mathcal{C}_{LNS} < 1$ is the cooling rate.

The solution returned at the termination of the proposed ILNS algorithm, i.e. S_{best}, (line 22), is either a feasible solution that contains no requests to battery swaps, or otherwise it is a solution that includes at least one requested swapping service by a known customer at a known time. In the former case there is no need to go to the second stage of the algorithm, but in the latter case the second stage solver is run which is described next.

Algorithm 4-3 *ILNS*

```
Generate an initial solution and denote it by $S_{candid}$;
$S_{best} \leftarrow S_{candid}$;
$temp_{LNS} \leftarrow startingTemp_{LNS}$;
for $iter = 1$ to $maxIter_{LNS}$ do
    $S_{curr} \leftarrow S_{candid}$;
    $S_{new} \leftarrow$ Select a removal heuristic and apply it on $S_{curr}$;
    $S_{curr} \leftarrow$ Select an insertion heuristic and apply it on $S_{new}$;
    if $cost(S_{curr}) < cost(S_{best})$ then
        $S_{best} \leftarrow S_{curr}$;
        while $true$ do
            $S_{new} \leftarrow$ Apply the intensification procedure on $S_{curr}$;
            if $cost(S_{new}) < cost(S_{best})$ then $S_{best} \leftarrow S_{new}$ and $S_{curr} \leftarrow S_{new}$ else break end if
        end while
    end if
    if $cost(S_{curr}) < cost(S_{candid})$ then
        $S_{candid} \leftarrow S_{curr}$;
    else if $rand < exp(-(cost(S_{curr}) - cost(S_{candid}))/temp_{LNS})$ then
        $S_{candid} \leftarrow S_{curr}$;
    end if
    $temp_{LNS} \leftarrow temp_{LNS} \times \mathcal{C}_{LNS}$
end for
return $S_{best}$
```

4.5.2 The second stage problem: the VRPTW

As discussed above and illustrated using the example in Figure 4-4, the solution to the first stage problem (i.e. S_{best}) clearly identifies the spatial and temporal characteristics of the required synchronisation between the BSV and the ECV with a depleted battery. This valuable information could be used to deduce a very small and simple to solve VRPTW for the BSV(s). To describe the resulting VRPTW, assume $\mathbb{C} = \{1, \dots, \mathbb{c}\}$ denotes the set of the customers that require a battery swapping service as deduced by S_{best}. Also, suppose that $\mathbb{t}_i$ denotes the swapping service start time at customer $i \in \mathbb{C}$ in S_{best} (this is the time when ECV has finished serving customer $i \in \mathbb{C}$). Then, the second stage problem is defined on a graph $\mathbb{G} = (\mathbb{N}, \mathbb{A})$, where $\mathbb{N} = D \cup \mathbb{C}$, and $\mathbb{A} = \{(i,j) | i, j \in \mathbb{N}, i \neq j\}$. Recall that each BSV can arrive at a customer before $\mathbb{t}_i$, but it needs to wait until swapping can take place. Also, note that the second stage problem is no longer solved on a multi-graph as BSVs are banned from visiting CSs.

The second stage problem is hence a conventional VRPTW with an additional constraint on the BSV charge availability as given below:

$$Min \sum_{(i,j) \in \mathbb{A}} d_{ij} z_{ij} \tag{4-53}$$

Subject to:

$$\sum_{j \in N_3} z_{ij} = 1, \qquad \forall i \in C \tag{4-54}$$

$$(\mathbb{t}_i + \mathscr{s}) z_{ij} + d_{ij} z_{ij} - l_0 (1 - z_{ij}) \leq w_j, \qquad \forall i \in N_2, j \in N_3, i \neq j \tag{4-55}$$

$$w_i = \mathbb{t}_i + \mathscr{s}, \qquad \forall i \in C \tag{4-56}$$

and (4-5), (4-13), (4-14) and (4-19).

The second stage VRPTW is very 'small' as $\mathbb{C}$ is only a small fraction of C, and it is very 'simple' to solve as time-windows are very tight. Therefore, this problem can be simply solved to optimality using an off-the-shelf solver.

We demonstrate the performance of the proposed 2S-MatHeu algorithm in the next section of the paper.

4.6 Computational results

In this section, we present the numerical experiments conducted to gain insights on the newly proposed problem of the EVRP-SABS and to evaluate the effectiveness of the proposed EPIP and the 2S-MatHeu algorithm. The section begins by introducing the developed EVRP-SABS benchmark instances and is then followed by the evaluation of the proposed EPIP-based formulation in finding exact solutions to the EVRP-SABS and its closely related class of the EVRPTW with CSs. Small EVRP-SABS instances with up to 25 customers and 21 CSs are solved to optimality using the proposed EPIP-based formulation and it is shown that without the application of the proposed EPIP, and using the original formulation, the problem becomes very soon intractable. Moreover, the exact solution to these small test instances provides a benchmark for the evaluation of the proposed 2S-MatHeu algorithm. We also apply an EPIP-based formulation of the EVRPTW on the instances developed by Schneider et al. (2014) in this section, and demonstrate that by just putting the formulation directly into a standard branch-and-bound solver, one can solve and improve several of the instances (up to 100 customer nodes) that were only possible to solve previously using a sophisticated CG-based solution algorithm (Desaulniers et al., 2016). The experiments are completed by evaluating the proposed 2S-MatHeu against the optimal and near optimal solutions found to small-sized instances, and by its application on EVRP-SABS instances with 100 customers.

All the experiments were performed on a computer with Intel Core™ i5 3.40 GHz processor with 8 GB RAM. The branch-and-bound solver of CPLEX™ 12.6.3 was used as the exact solver, and all other algorithms were coded in MATLAB™.

4.6.1 Generation of EVRP-SABS test instances

The EVRP-SABS test instances developed in this paper are created by applying several modifications on the EVRPTW instances by Schneider et al. (2014) such that battery swaps might be potentially required. The test problems in Schneider et al. (2014) are developed based on the well-known benchmark instances for the VRPTW proposed by Solomon (1987) which comprises six sets of test problems (C1, R1, RC1, C2, R2, and RC2). Instances in the sets C1 and C2 are with clustered geographical data, instances in R1 and R2 are generated by a random uniform distribution, and instances in RC1 and RC2 are semi-clustered instances that contain a mix of randomly generated data and clusters. Problem sets in the first group (i.e. R1, C1, and RC1) have a short scheduling horizon, whereas the second group instances (i.e. R2, C2, and RC2) have a longer scheduling horizon. To extend these instances to their intended EVRPTW test problems, Schneider et al. (2014) introduce to each one the locations of a set of 21 CSs, one at the depot, and the other 20 ones at randomly selected locations, such that every customer can be reached from the depot using at most two different CSs. Since the detours for visits to CSs and the resulting recharging times make it impossible to comply with the customer time windows given in the original Solomon instances, they generate new time-windows to obtain feasible EVRPTW instances.

We apply three main modifications on the test problems proposed in Schneider et al. (2014) as follows: (i) the time-windows in their instances are reverted to the original Solomon instances, (ii) an inverse recharging rate which is three times the recharging rate they use for each instance is used, and (iii) battery capacity has been reduced to the extent that the feasibility of instances is not affected. These extra configurations make the problems harder to solve, and potentially lead to the use of BSVs in the solution. To ensure feasibility, however, BSVs' battery capacity has been set more than twice the battery capacity of an ECV in each instance, but the consumption rate of a BSV has been set equal to that of the ECVs. Each BSV is assumed to be able to carry a maximum of 5 batteries, and swapping service time across all instances is assumed to be 3 time units. In general, EVRP-SABS instances of sizes 5, 10, 15, 25, and 100 customers are generated, where instances with 25 and 100 customer locations comprise 21 CS locations.

4.6.2 The performance of the proposed EPIP

To evaluate the proposed EPIP, we initially solve EVRP-SABS instances of size 5, 10, and 15 to optimality (or near optimality) using the original formulation of the EVRP-SABS given in (4-1)-(4-20), and compare it with the solution of the EPIP-based formulation in (4-34)-(4-52). Since the original formulation cannot handle instances with more than 15 customers, we also solve some of the instances with 25 customers using the EPIP-based formulation for benchmarking purposes. This provides a basis for the performance evaluation of the proposed 2S-MatHeu algorithm in approximating the optimal solutions in the next section. Note that the solver is given a maximum of 3600 seconds for each instance, and upon this termination criterion, if there is any MIP gap, it is reported.

The results of the experiments on instances with 5, 10 and 15 customers are presented in Table 4-2. In this table the headings denote the following: V_T: total number of vehicles (ECVs and BSVs) used in the solution; D_T: total distance travelled by all vehicles (ECVs and BSVs); V_E: total number of ECVs used in the solution; D_E: total distance travelled by ECVs; V_B: total number of BSVs used in the solution; D_B: total distance travelled by BSVs; S: total number of battery swaps requested, C: total number of visits to CSs; and t (s): total computing time in seconds. The table also reports the average number of CS-paths retained between every pair of required nodes after the application of the EPIP under the column with 'Avg. paths' heading.

While the original formulation is not able to solve instances with 25 customers and 21 CSs, we present the obtained solutions to these problems using the EPIP-based formulation in Table 4-3. In addition to demonstrating the effectiveness of the proposed EPIP, the solution to these instances provide a basis for the evaluation of the performance of the proposed 2S-MatHeu solution algorithm for the EVRP-SABS in the next sub-section.

In order to further investigate the effectiveness of the proposed EPIP in problem classes that are closely related to the EVRP-SABS (e.g. the EVRPTW and the GVRP), we demonstrate that by just putting an EPIP-based formulation of the EVRPTW into CPLEX, one can obtain results that are not only very well comparable with a sophisticated CG-based algorithm proposed by Desaulniers et al. (2016), but are also improving several of their results in terms of the number of vehicles required. In Desaulniers et al. (2016) four variants of the EVRPTW resulting from the combination of the adopted recharging strategy (i.e. full or partial) and recharging frequency (single or multiple recharge per route) are considered and for each variant, an exact branch-price-and-cut algorithms is presented. They mainly concentrate on the first group of test problems developed by Schneider et al. (2014) (i.e. test sets R1, C1, and RC1), which are characterized by narrow time windows, and show that they can solve instances with up to 100 customers and 21 recharging stations. We also use the same set of 25, 50 and 100 test instances they use and show that the EPIP-based formulation can handle a majority of them. Note that the EPIP-based formulation can tackle the second variant of the EVRPTW they consider, i.e. multiple recharges per route, with full recharges only. Similar to their study we have applied a maximum computational time of 3600 s on CPLEX.

Table 4-2 Comparison of the original formulation for the EVRP-SABS with the EPIP-based formulation

No.	Inst.	Original formulation				EPIP-based formulation										
		V_{T}	D_{T}	t (s)	MIP gap	Avg. paths	V_{E}	D_{E}	V_{B}	D_{B}	S	C	V_{T}	D_{T}	t (s)	MIP gap
1	C101-5	4	334.34	1.68	0.00	0.30	3	228.18	1	106.16	2	0	4	334.34	1.13	0.00
2	C103-5	2	220.67	1.13	0.00	1.60	1	159.06	1	61.61	1	1	2	220.67	1.28	0.00
3	C206-5	2	294.49	0.69	0.00	2.10	1	229.72	1	64.78	1	3	2	294.49	0.95	0.00
4	C208-5	2	257.96	0.69	0.00	1.60	1	165.77	1	92.18	2	3	2	257.96	0.80	0.00
5	R104-5	4	209.00	0.75	0.00	1.80	3	161.81	1	47.19	2	2	4	209.00	0.47	0.00
6	R105-5	3	235.83	1.03	0.00	0.90	2	180.26	1	55.57	1	2	3	235.83	0.08	0.00
7	R202-5	2	215.86	0.89	0.00	2.75	1	146.03	1	69.83	2	2	2	215.86	0.38	0.00
8	R203-5	2	340.50	3.23	0.00	1.90	1	257.54	1	82.9616	2	7	2	340.50	0.14	0.00
9	RC105-5	4	366.45	0.64	0.00	0.45	3	252.03	1	114.42	2	3	4	366.45	0.05	0.00
10	RC108-5	5	469.22	3600.00	0.22	1.40	4	418.23	1	50.99	1	7	5	469.22	0.14	0.00
11	RC204-5	2	289.14	4.19	0.00	2.50	1	204.65	1	84.50	2	3	2	289.14	0.20	0.00
12	RC208-5	2	228.20	0.73	0.00	1.65	1	177.62	1	50.59	1	4	2	228.20	0.16	0.00
13	C101-10	5	488.59	1.14	0.00	0.60	4	398.78	1	89.80	2	3	5	488.59	0.23	0.00
14	C104-10	3	370.99	35.55	0.00	2.72	2	273.92	1	97.08	4	2	3	370.99	65.06	0.00
15	C202-10	4	351.76	8.91	0.00	1.59	3	311.64	1	40.12	2	6	4	351.76	0.58	0.00
16	C205-10	2	461.27	694.76	0.00	1.37	1	355.31	1	105.96	4	3	2	461.27	1.38	0.00
17	R102-10	-	-	3600.00	-	1.10	4	323.09	1	96.42	3	5	5	419.51	3.44	0.00
18	R103-10	3	228.40	3600.00	0.00	2.09	2	162.35	1	64.06	2	0	3	226.40	13.49	0.00
19	R201-10	2	373.36	3600.00	0.32	1.74	1	302.75	1	70.61	3	7	2	373.36	1.51	0.00
20	R203-10	2	334.25	83.95	0.00	3.39	1	232.68	1	101.57	2	4	2	334.25	48.77	0.00
21	RC102-10	5	485.08	4.25	0.00	0.56	4	414.40	1	70.68	1	3	5	485.08	0.19	0.00
22	RC108-10	4	502.10	482.77	0.00	1.33	3	369.73	1	132.37	3	2	4	502.10	219.95	0.00
23	RC201-10	3	383.82	5.13	0.00	1.83	2	325.51	1	58.31	1	5	3	383.82	1.41	0.00
24	RC205-10	3	462.77	61.00	0.00	1.67	2	394.97	1	67.80	2	5	3	462.77	1.91	0.00
25	C103-15	5	516.40	3600.00	0.31	2.15	3	338.81	2	175.47	5	2	5	514.28	1167.33	0.00
26	C106-15	4	481.84	49.88	0.00	0.73	3	342.21	1	139.63	4	2	4	481.84	4.33	0.00
27	C202-15	4	456.72	18.00	0.00	1.78	3	355.91	1	100.81	3	4	4	456.72	1.77	0.00
28	C208-15	3	335.71	25.73	0.00	1.61	2	288.68	1	47.02	2	4	3	335.71	2.94	0.00
29	R102-15	5	383.26	1107.44	0.00	2.69	4	338.19	1	45.07	2	3	5	383.26	48.41	0.00

30	R105-15	5	460.89	121.55	0.00	0.83	4	372.16	1	88.73	3	4	5	460.89	2.08	0.00
31	R202-15	5	704.35	54.65	0.00	1.89	2	494.60	3	209.75	7	13	5	704.35	45.83	0.00
32	R209-15	3	368.88	3600.00	0.18	3.14	2	318.88	1	50	1	6	3	368.88	319.84	0.00
33	RC103-15	7	635.71	4.75	0.00	1.03	5	459.89	2	175.8185	4	4	7	635.71	1.52	0.00
34	RC108-15	-	-	3600.00		1.42	4	438.00	1	103.0326	3	4	5	541.04	3600.00	0.41
35	RC202-15	3	398.94	3600.00	0.23	2.90	2	357.70	1	41.231	1	5	3	398.94	95.56	0.00
36	RC204-15	3	438.25	3600.00	0.24	4.16	2	346.58	1	91.671	2	7	3	438.25	3600.00	0.20

Table 4-3 Solutions to instances with 25 customers and 21 CSs using the EPIP-based formulation

No.	Inst.	EPIP-based formulation										
		Avg. paths	V_E	D_E	V_B	D_B	S	C	V_T	D_T	t (s)	MIP gap
1	C101-25	1.01	6	693.96	2	212.74	2	8	8	906.70	30.14	0.00
2	C102-25	1.50	5	534.58	2	213.29	2	6	7	747.87	566.66	0.00
3	C105-25	1.02	8	761.48	1	105.86	1	4	9	867.35	84.95	0.00
4	C106-25	1.34	8	787.07	0	0.00	0	0	8	787.07	67.66	0.00
5	C205-25	2.40	3	431.54	1	51.37	1	2	4	482.91	8.53	0.00
6	C208-25	2.93	3	470.12	0	0.00	0	0	3	470.12	43.47	0.00
7	R101-25	0.73	10	699.15	3	258.66	3	7	13	957.81	1.05	0.00
8	R105-25	1.28	8	662.13	1	99.59	1	4	9	761.72	488.81	0.00
9	RC101-25	0.93	9	865.39	1	125.6398	1	3	10	991.03	42.98	0.00
10	RC102-25	1.77	9	815.68	1	119.10	1	3	10	934.78	3600.00	0.04
11	RC201-25	2.82	2	948.21	0	0.00	0	0	2	948.21	115.56	0.00
12	RC206-25	3.88	3	593.81	1	72.62	1	3	4	666.43	3299.53	0.00

The results of this comparison is presented in Table 4-4. In this table, solutions obtained using the EPIP-based formulation are compared with the solutions reported by Desaulniers et al. (2016) (under the heading Des. et al) for EVRPTW instances with 25, 50 and 100 customers and 21 CSs. EPIP-based solutions in italic are matching with Des. et al solutions, and solutions in bold are improving their solutions. Note that the improvements are either in the total number of ECVs required, or they are solutions to problems that had remained unsolved using the branch-price and-cut algorithm of Desaulniers et al. (2016). It must be mentioned that the algorithm proposed by Desaulniers et al. (2016) does not include features to minimise total number of ECVs and can only minimise distance.

Table 4-4 The EPIP-based formulation for the EVRPTW instances with 25, 50 and 100 customers and 21 CSs

Inst.	25				50				100			
	Des. et al		EPIP-		Des. et al		EPIP-based		Des. et al		EPIP-based	
	k	d	k	d	k	d	k	d	k	d	k	d
C101	7	626.90	**5**	**708.90**	9	783.59	**7**	**904.71**	12	1053.83	12	1093.98
C102	5	526.24	*5*	*526.24*	8	784.67	9	788.39	12	1022.58	-	-
C103	4	345.41	*4*	*345.41*	7	656.67	7	677.38	-	-	-	-
C104	4	449.75	**4**	**449.53**	5	582.68	6	600.00	-	-	**13**	**1226.23**
C105	6	541.35	**4**	**620.13**	9	736.76	8	777.59	12	1033.93	12	1062.39
C106	5	562.27	**4**	**619.38**	9	754.95	9	763.98	12	1027.25	-	-
C107	6	505.73	**4**	**628.98**	7	708.70	*7*	*708.70*	12	1025.63	-	-
C108	5	508.27	*5*	*508.27*	8	725.97	*8*	*725.97*	-	-	-	-
C109	4	473.41	**4**	**475.28**	7	677.02	7	686.36	-	-	**12**	**1030.12**
R101	9	662.15	9	662.80	12	939.87	**11**	**961.76**	20	1601.76	**18**	**1639.9**
R102	6	452.90	**5**	**470.18**	10	866.67	-	-	18	1454.91	-	-
R103	6	494.45	*6*	*494.45*	9	803.16	10	818.99	-	-	**17**	**1350.61**
R104	4	351.99	*4*	*351.99*	-	-	**7**	**633.56**	-	-	**14**	**1254.76**
R105	6	584.41	*6*	*584.41*	10	842.41	*10*	*842.41*	15	1340.18	-	-
R106	5	480.06	*5*	*480.06*	9	793.95	9	797.55	14	1229.21	-	-
R107	5	416.33	5	417.23	8	691.35	-	-	-	-	**17**	**1422.08**
R108	4	429.19	-	-	6	543.49	7	585.04	-	-	**14**	**1251.22**
R109	5	462.05	*5*	*462.05*	8	789.35	-	-	-	-	-	-
R110	4	419.49	4	427.21	7	713.43	-	-	-	-	-	-
R111	4	382.86	4	382.86	7	745.12	-	-	-	-	**16**	**1336.75**
R112	4	397.24	4	397.24	6	602.81	7	659.12	-	-	**15**	**1352.16**
RC101	7	737.98	**6**	**791.60**	11	1074.13	*11*	*1074.13*	-	-	**17**	**1706.17**
RC102	7	648.34	7	649.31	10	897.21	10	922.79	16	1531.77	-	-
RC103	6	560.73	*6*	*560.73*	8	829.42	9	873.25	-	-	-	-
RC104	4	516.28	*4*	*516.28*	7	689.97	7	693.97	-	-	**14**	**1465.81**
RC105	6	589.68	*6*	*589.68*	10	983.94	-	-	15	1482.19	16	1650.58
RC106	5	557.08	*5*	*557.08*	8	887.99	9	911.92	-	-	-	-
RC107	4	497.48	*4*	*497.48*	7	786.18	8	811.81	-	-	-	-
RC108	4	479.58	*4*	*479.58*	-	-	**7**	**751.36**	-	-	-	-

4.6.3 The performance of the proposed 2S-MatHeu

In this section, we first compare the solutions obtained using the proposed 2S-MatHeu with the optimal (or near optimal) solutions found for the EVRP-SABS instances of size 5, 10, 15, and 25 in the previous section, and then report the solutions to instances with 100 customers and 21 CSs. To determine the value of the parameters used inside 2S-MatHeu, we have conducted a preliminary set of empirical analyses and chosen the following values: ρ_{rem}= 0.4, ρ_{shaw} = 6, ρ_{worst} = 3, η = 0.025, λ = n/150, $maxIter_{LNS}$ = 30n, $startingTemp_{LNS}$ = 100, $coolingRate_{LNS}$ = 0.997, $maxIter_{INT}$ = number of routes in the solution, $timeLimit_{INT}$ = 0.5, $startingTemp_{INT}$ = 500, and $coolingRate_{INT}$ = 0.992.

Table 4-5 presents the solutions obtained by applying the proposed 2S-MatHeu algorithm on instances with 5, 10, 15, and 25 customers. While in over 81% of the cases the solution returned by the 2S-MatHeu matches exactly with the optimal solution, and in one case (instance number 36) it improves the solution returned by CPLEX after 3600 seconds, in 9 cases the 2S-MatHeu solution does not match with the exact solution. Further investigation of the reason for this, however, can be insightful for future solution developments for the EVRP-SABS. In Figure 4-5, a visual comparison between these non-matching solutions is presented in terms of ECV and BSV distance, ECV and BSV numbers assigned to routes, and total number of batteries required by the solution. This figure reveals that this non-matching is due to the 'interdependence problem' (Drexl, 2012) discussed earlier that is inherent to the EVRP-SABS . While using the "total number of batteries" as the medium of communication between the two stages in our proposed 2S-MatHeu algorithms turns out to be successful in most of the cases, it is not always guaranteed to lead to the best solution. Figure 4-5 shows that in case of all these instances the solution returned by the 2S-MatHeu always returned an equal or smaller number of batteries required (what the first stage problem was intended to do). In the case of these instances, the minimisation of the number of batteries has usually led to a larger ECV distance, or even BSV distance. The figure also suggests that even in the case that the number of batteries in case of the 2S-MatHeu and the exact do match, the minimisation of the ECV distance in the first stage can lead to a larger BSV distance in the second stage. For example, in case of inst #26, while in both exact and the 2S-MatHeu 4 batteries are needed, the minimisation of the ECV distance in the first stage of the 2S-MatHeu (15% smaller than the ECV distance in the exact) has led to 5% increase in the total BSV distance and incurred need to an extra BSV for completing the delivery. However, it must be noted that the overall difference between the solutions returned by the proposed 2S-MatHeu and the exact solutions in case of these instances and in terms of the total number of vehicles needed and distance travelled is only marginally different.

Finally, the proposed 2S-MatHeu algorithm has been applied on the instances with 100 customers and 21 CSs. The result of this experiment is presented in Table 4-6. These results indicate that in case of instances in the sets R2 and RC2, which have a longer scheduling horizon, swapping service has never been requested and optimisation has been completed in the first stage. This also shows the effectiveness of 2S-MatHeu in driving the search towards solutions with no need to swapping services. In case of the other instances, the ratio of number of batteries swapped and the number of BSVs utilised indicates that on average each BSV delivers 1.9 batteries over its delivery route.

In summary, results presented in this section show that the performance of the 2S-MatHeu as an initial algorithm to tackle the newly introduced problem of EVRP-SABS sounds acceptable and can provide a benchmark for future solution developments for the problem.

Table 4-5 2S-MatHeu solutions to the EVRP-SABS instances with 5, 10, 15, and 25 customers

No.	Inst.	The 2S-MatHeu							
		V_E	D_E	V_B	D_B	S	V_T	D_T	t (s)
1	C101-5	3	228.18	1	106.16	2	4	334.34	0.16
2	C103-5	1	159.06	1	61.61	1	2	220.67	0.19
3	C206-5	1	229.72	1	64.78	1	2	294.49	0.14
4	C208-5	1	165.77	1	92.18	2	2	257.96	0.11
5	R104-5	3	174.39	1	36.77	1	4	211.16	0.39
6	R105-5	2	180.26	1	55.57	1	3	235.83	0.27
7	R202-5	1	146.03	1	69.83	2	2	215.86	0.15
8	R203-5	1	257.54	1	82.96	2	2	340.50	0.27
9	RC105-5	3	252.03	1	114.42	2	4	366.45	0.65
10	RC108-5	4	418.23	1	50.99	1	5	469.22	0.37
11	RC204-5	1	218.88	1	72.11	1	2	290.99	0.16
12	RC208-5	1	177.62	1	50.59	1	2	228.20	0.53
13	C101-10	4	398.78	1	89.80	2	5	488.59	0.41
14	C104-10	2	272.32	1	126.37	3	3	398.69	1.64
15	C202-10	3	311.64	1	40.12	2	4	351.76	0.69
16	C205-10	1	353.01	1	129.10	4	2	482.11	1.44
17	R102-10	4	323.09	1	96.42	3	5	419.51	0.73
18	R103-10	2	162.35	1	64.06	2	3	226.40	0.82
19	R201-10	1	302.75	1	70.61	3	2	373.36	2.64
20	R203-10	1	232.68	1	101.57	2	2	334.25	3.99
21	RC102-10	4	414.40	1	70.68	1	5	485.08	0.84
22	RC108-10	3	369.73	1	132.37	3	4	502.10	1.26
23	RC201-10	2	325.51	1	58.31	1	3	383.82	1.56
24	RC205-10	2	394.97	1	67.80	2	3	462.77	0.95
25	C103-15	3	338.81	2	175.47	5	5	514.28	2.69
26	C106-15	3	290.02	2	146.60	4	5	436.62	1.78
27	C202-15	3	355.91	1	100.81	3	4	456.72	2.60
28	C208-15	2	288.68	1	47.02	2	3	335.71	5.40
29	R102-15	4	338.19	1	45.07	2	5	383.26	2.65

30	R105-15	5	386.12	1	52.50	1	6	438.62	1.96
31	R202-15	2	494.60	3	209.75	7	5	704.35	6.18
32	R209-15	2	318.88	1	50.00	1	3	368.88	41.90
33	RC103-15	5	459.89	2	175.82	4	7	635.71	2.56
34	RC108-15	4	438.00	1	103.03	3	5	541.04	3.62
35	RC202-15	2	348.79	1	54.92	1	3	403.71	15.16
36	RC204-15	1	377.50	1	91.67	2	2	469.17	119.98
37	C101-25	6	693.96	2	212.74	8	8	906.70	7.70
38	C102-25	5	529.70	3	286.95	5	8	816.66	6.39
39	C105-25	8	761.48	1	105.86	4	9	867.35	6.05
40	C106-25	8	787.07	0	0.00	0	8	787.07	60.22
41	C205-25	3	431.54	1	51.37	2	4	482.91	76.72
42	C208-25	3	470.12	0	0.00	0	3	470.12	45.06
43	R101-25	11	761.03	3	242.44	3	14	1003.47	6.70
44	R105-25	8	662.13	1	99.59	4	9	761.72	7.51
45	RC101-25	9	865.39	1	125.64	3	10	991.03	6.96
46	RC102-25	9	815.68	1	119.10	3	10	934.78	6.96
47	RC201-25	2	948.21	0	0.00	0	2	948.21	68.61
48	RC206-25	3	584.29	1	114.02	1	4	698.31	107.04

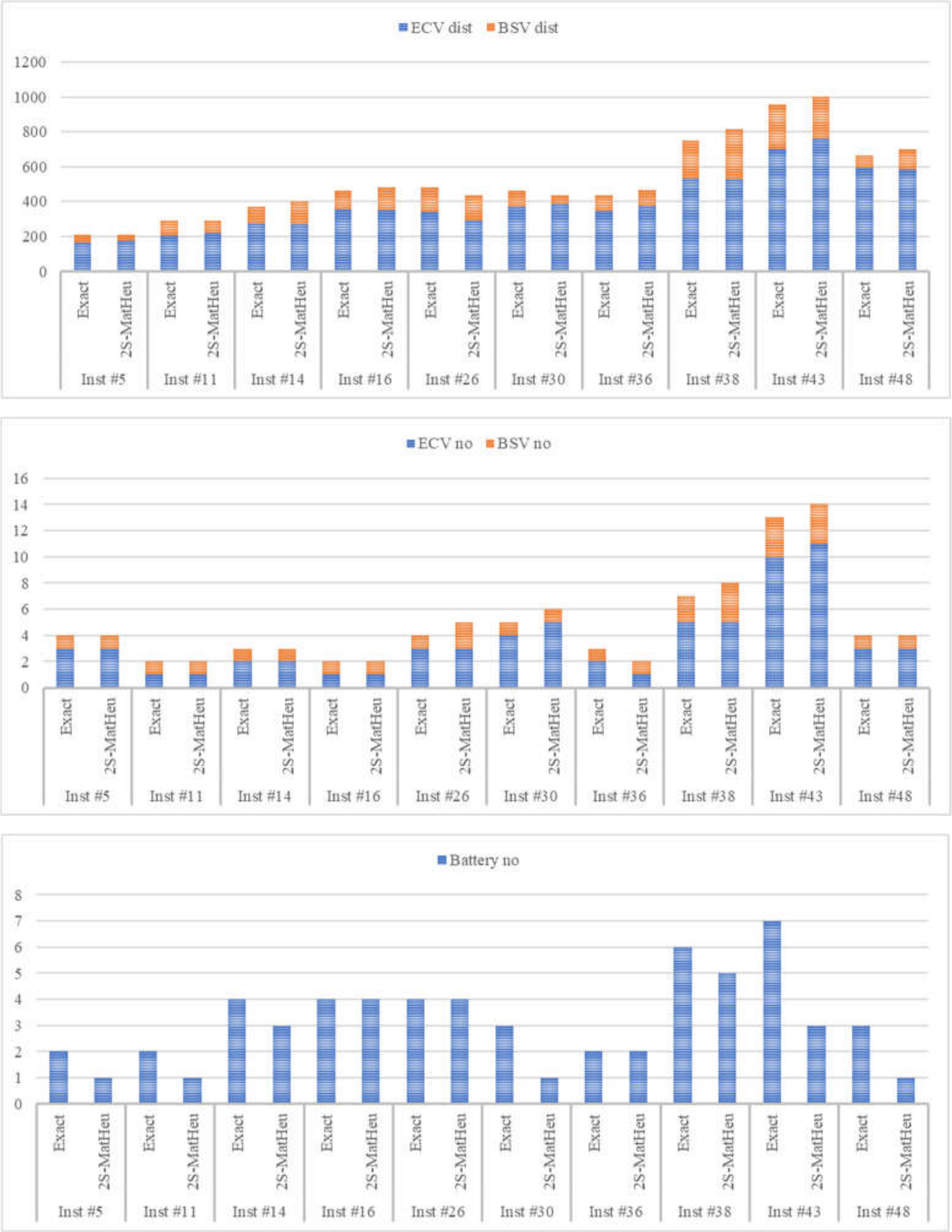


Figure 4-5 Comparison of the exact and the 2S-MatHeu algorithm in non-matching instances

Table 4-6 2S-MatHeu solutions to the EVRP-SABS instances with 100 customers and 21 CSs

No.	Inst.	Avg. paths	2S-MatHeu							
			$\mathrm{V_E}$	$\mathrm{D_E}$	$\mathrm{V_B}$	$\mathrm{D_B}$	S	$\mathrm{V_T}$	$\mathrm{D_T}$	t (m)
1	C101	1.00	13	1068.33	5	444.90	7	18	1513.23	25.22
2	C102	2.17	12	1002.88	3	357.99	9	15	1360.87	22.68
3	C103	3.46	11	948.63	5	459.85	10	16	1408.48	23.47
4	C104	4.81	10	841.51	4	434.44	10	14	1275.95	21.27
5	C105	1.18	12	964.10	4	429.98	8	16	1394.08	23.23
6	C106	1.31	12	953.91	4	416.86	8	16	1370.77	22.85
7	C107	1.39	12	939.26	5	453.81	8	17	1393.07	23.22
8	C108	1.62	12	1002.72	5	476.75	8	17	1479.47	24.66
9	C109	2.15	10	869.15	5	466.11	11	15	1335.26	22.25
10	C201	2.76	4	771.24	2	182.09	6	6	953.33	15.89
11	C202	4.22	5	967.40	2	161.92	2	7	1129.32	18.82
12	C203	5.31	4	760.88	2	206.73	4	6	967.61	16.13
13	C204	6.02	4	807.23	2	242.57	5	6	1049.80	17.50
14	C205	3.03	4	838.42	3	259.85	5	7	1098.27	18.30
15	C206	3.35	4	786.41	3	264.97	6	7	1051.38	17.52
16	C207	3.61	4	798.93	3	321.29	6	7	1120.22	18.67
17	C208	3.64	4	809.04	2	166.11	4	6	975.15	16.25
18	R101	0.83	23	3125.04	0	0.00	0	23	3125.04	52.08
19	R102	2.31	20	2761.56	0	0.00	0	20	2761.56	46.03
20	R103	3.85	17	2222.68	0	0.00	0	17	2222.68	37.04
21	R104	5.14	15	1224.49	5	386.34	10	20	1610.83	26.85
22	R105	1.15	23	1816.32	5	369.46	6	28	2185.78	36.43
23	R106	2.53	19	1623.45	5	396.92	10	24	2020.37	33.67
24	R107	3.91	17	1428.44	5	437.12	8	22	1865.56	31.09
25	R108	5.30	15	1227.00	5	402.32	7	20	1629.32	27.16
26	R109	1.81	18	1511.76	5	379.58	8	23	1891.34	31.52
27	R110	2.91	18	1459.61	5	354.74	6	23	1814.35	30.24
28	R111	3.24	18	1468.13	4	348.17	8	22	1816.30	30.27
29	R112	4.38	15	1240.66	4	357.42	7	19	1598.08	26.63
30	R201	3.89	6	1212.63	0	0.00	0	6	1212.63	20.21
31	R202	5.18	5	1114.26	0	0.00	0	5	1114.26	18.57
32	R203	6.30	4	908.60	0	0.00	0	4	908.60	15.14
33	R204	6.87	4	794.14	0	0.00	0	4	794.14	13.24
34	R205	4.94	4	976.25	0	0.00	0	4	976.25	16.27

35	R206	5.86	5	917.34	0	0.00	0	5	917.34	15.29
36	R207	6.44	4	862.32	0	0.00	0	4	862.32	14.37
37	R208	7.00	3	761.77	0	0.00	0	3	761.77	12.70
38	R209	5.91	5	878.22	0	0.00	0	5	878.22	14.64
39	R210	5.88	4	959.74	0	0.00	0	4	959.74	16.00
40	R211	6.83	4	776.47	0	0.00	0	4	776.47	12.94
41	RC101	0.83	21	2048.80	6	573.53	12	27	2622.33	43.71
42	RC102	2.14	19	1788.52	6	520.97	11	25	2309.49	38.49
43	RC103	3.52	16	1598.74	5	511.56	9	21	2110.30	35.17
44	RC104	4.95	13	1363.72	6	592.10	11	19	1955.82	32.60
45	RC105	1.39	20	1814.21	6	616.65	12	26	2430.86	40.51
46	RC106	1.40	17	1659.43	7	683.00	12	24	2342.43	39.04
47	RC107	2.24	15	1522.86	7	629.32	11	22	2152.18	35.87
48	RC108	3.26	15	1404.08	5	543.98	10	20	1948.06	32.47
49	RC201	3.98	6	1520.70	0	0.00	0	6	1520.70	25.35
50	RC202	5.43	5	1210.20	0	0.00	0	5	1210.20	20.17
51	RC203	6.48	5	960.39	0	0.00	0	5	960.39	16.01
52	RC204	7.23	4	818.20	0	0.00	0	4	818.20	13.64
53	RC205	4.97	6	1267.47	0	0.00	0	6	1267.47	21.12
54	RC206	5.09	5	1077.20	0	0.00	0	5	1077.20	17.95
55	RC207	6.08	5	1010.89	0	0.00	0	5	1010.89	16.85
56	RC208	7.32	4	850.16	0	0.00	0	4	850.16	14.17

4.7 Discussion and conclusion

To address the issue of range anxiety in goods distribution with ECVs, in this paper a new paradigm shift in EVRPTWs was proposed by exploiting new relevant technological developments that make mobile battery swapping or mobile rapid recharging possible. The new problem class of the EVRP-SABS in which an ECV can request a battery swapping service from a BSV on-the-fly was introduced and formulated. Using BSVs in the design of the ECV routes brings about savings in costs incurred by the total number of vehicles required and the total travelling distance. More importantly, it becomes an indispensable solution when ECV's driving range is so much restrictive to the extent that a feasible solution to the problem in hand is unobtainable. Hence, keeping recharging at available CSs in the network as the primary solution to EVRPTWs, we introduced ambulant battery swapping as the last resort when a feasible route cannot be retrieved by just visiting CSs.

To address the proposed problem, we developed new combinatorial results leading to an exact EPIP that allows the identification of all CS-paths that can contribute to an optimal solution, and eliminate all redundant paths from considerations in a pre-processing stage. Using the proposed EPIP, a strengthened EPIP-based MILP formulation was proposed for the problem and it was demonstrated that by just putting the formulation into a standard branch-and-bound solver one can solve larger size instances of the EVRP-SABS and its related class of the EVRPTW with CSs. Furthermore, exploiting the proposed EPIP, a two-stage matheuristic was proposed for the EVRP-SABS. Benefiting from an EPIP-based DP at its core, the first stage of the proposed solution algorithm tries to find feasible delivery routes for the employed ECVs without requesting any ambulant battery swapping services from a BSV; however, if such solution is not found, the first stage solver transfers spatiotemporal information of the requested swapping services to the second stage solver where a simple and small VRPTW is solved to optimality to route the required number of BSVs to provide the requested swapping services at the designated points and times.

The paper derived new benchmark instances for the EVRP-SABS from the available EVRPTW test instances and several numerical experiments where conducted to demonstrate the effectiveness of the proposed EPIP and the 2S-MatHeu. It was shown that the EPIP-based formulation can solve and improve several of the EVRPTW instances that were only solved previously using a dedicated branch-price-and-cut algorithm. The efficiency of the proposed 2S-MatHeu was also demonstrated against the available optimal (or near optimal) solutions and by its application on large sized test instances with 100 customers and 21 CSs. The numerical experiments of the proposed matheuristic provided also an insight on the complexity of the EVRP-SABS due to the inherent problem of interdependence that usually emerges in problems with spatiotemporal synchronisation requirements. In the case of the EVRP-SABS there is a significant trade-off between different attributes of the problem and a slight change in the delivery route and schedule of an ECV can have a considerable impact on the route of the BSV, and vice versa. One way to address these trade-offs is to integrate both stages of the problem and instead of postponing the evaluation of a solution until the termination of the first stage, one can conduct a full integrated ECV and BSV evaluation for every newly generated solution. However, this is evidently a very computationally expensive task. Alternatively, one can approach the first stage problem as a multi-objective optimisation problem and instead of returning one solution only, return a pool of trading off solutions to choose from for the second stage. In any case, it is a significant open research direction to develop new sophisticated algorithms for the EVRP-SABS.

Along with the development of tailored solution algorithms for the EVRP-SABS, there are several other important future research directions to extend the proposed problem in this paper. For example, in EVRPTWs the availability of a planned to visit CS on an ECV route is very crucial to the attainment of a feasible solution to complete the distribution task; if for any reason, the planned CS becomes unavailable upon the arrival of the ECV, the entire routing plan can be disrupted and yielded infeasible. The use of ambulant BSVs in this context can be quite helpful. Moreover, to reuse the expensive resources in the fleet, multi-trip planning of ECVs and BSVs can help cutting down on vehicle hiring costs and can be an important future line of research.

5. CONCLUSION

5.1 Summary

Urban freight distribution plays a vital role in the functioning of urban economies and is growing at a rapid pace due to the process of urbanisation. However, it is contributing significantly to problems such as traffic congestion and environmental pollution. Since its advent in the 1950s, the vehicle routing problem has contributed rather implicitly to reducing emissions from delivery routes by minimising the total distance travelled, and it has been recently enriched to incorporate environmental considerations more explicitly in the design of emissions-aware trips for distribution vehicles.

The main goal of this research is to contribute to greening urban freight distribution by developing new mathematical models and solution algorithms pertaining to the two major steams in VRPs with environmental considerations; i.e.: (i) VRPs with an explicit fuel consumption estimation component as a proxy for emissions, *aka* emissions minimising VRPs, and (ii) VRPs with vehicles in the fleet that run on a cleaner alternative fuel such as electricity, natural gas, hydrogen-gas, biofuel, etc., *aka* green VRPs. In the first stream, this thesis developed and solved a new realistic multi-objective variant of the pollution-routing problem that is studied directly on the original urban roadway network, and in the latter stream a paradigm shift in routing of ECVs was proposed by introducing the electric vehicle routing problem with ambulant battery swapping/recharging that exploits new technological developments corresponding to the possibility of mobile battery swapping (or recharging) of ECVs using a battery swapping van. The outcome of this research has been the study of 3 research topics that are briefly summarised in the sequel.

In chapter 2 of the thesis, the multi-objective Steiner PRP on congested urban road networks was introduced. This variant of the PRP is capable of incorporating the real operating conditions of urban freight distribution, and striking a balance between traditional business and environmental objectives. The proposed model integrates all factors that have a major impact on fuel consumption including the time-varying congestion speed, vehicle load, vehicle's physical and mechanical characteristics, and acceleration and deceleration rates. For the latter factor that has been proven to have a significant impact on the fuel consumption of the vehicle and consequently the emissions (Turkensteen, 2017), a new model was proposed to synthetize spatiotemporal driving cycles that can effectively represent the expected second-by-second speed profile of a vehicle travelling over a given road-link at a given time of the day. The chapter also showed that using multiple trips can bring in significant cost and emissions savings by using energy efficient resources in the fleet more than once. In addition to integrating all major factors that contribute to fuel consumption in an integrated modelling and solution scheme, the main added value of the study in this chapter is to develop techniques to overcome the difficulty of studying the proposed problem directly on the road network and to make the incorporation of important data contained in the original roadway network into the model possible. Last but not the least, the study of the problem as a multi-objective optimisation problem puts forth useful implications for decision making by providing a clear picture of the real trade-offs between business and environmental objectives.

While the problem proposed in the second chapter of the thesis is capable of providing a rather accurate exposition of the real conditions freight distributers face in urban areas, the resulting problem is significantly challenging to solve even in the case of small test instances. This problem unifies several hard variants of the VRP including the time-and-load dependent VRP, the multi-trip VRP, and the fleet size and mix VRP in a multi-objective optimisation setting and under the compound situation of solution development on the original roadway network where multiple paths between consecutive visits of a truck must be identified and considered. Therefore, in chapter 3, I focused on the development of tailored solution algorithms for the SPRP that can put forward a reasonable set of trading off solutions within a reasonable computational time. Three different multi-objective optimisation heuristics were hence developed for the SPRP and were compared against one another. Along with the introduction of innovative approaches to decompose and simplify the lower-level problems that arise in the context of the SPRP, the proposed solution algorithms have the added value of solving a unification of several hard and rich variants of the VRP. The outcome of these algorithms provides the decision maker with a pool of solutions representing clear trade-offs between the business and environmental objectives. The chapter also proposed an archive of benchmark test instances that resemble real world congested urban road networks. These test instances could be used for future algorithmic developments and examination with different logistics solutions and scenarios.

The third research topic presented in chapter 4 of the thesis turns attention towards the ultimate viable solution to combatting emissions from UFD in urban centres, i.e. to use electric commercial vehicle for last-mile delivery. To tackle the significantly impeding problem of range anxiety in the face of goods distribution using ECVs, the article presented in this chapter introduces the electric vehicle routing problem with synchronised battery swapping/recharging. The proposed problem is motivated by new technological developments that make mobile battery swapping or recharging of ECVs on-the-fly possible. In the EVRP-SABS, routing takes place in two levels for the ECVs that carry out delivery tasks, and for the BSVs that provide the running ECVs with fully charged batteries on their route. There is, therefore, a need to establish temporal and spatial synchronisations between the vehicles in the two levels and to do so a dedicated two-stage matheuristic was proposed that while decomposes the two levels into two independent stages, retains the communication between the two stages using the medium of “the total number of batteries required”. The paper demonstrated that through the exploitation of the new technologies pertinent to mobile battery swapping in the routing of ECVs, it is not only possible to achieve savings in costs, but also in realistic situations when completing the delivery tasks by merely visiting the available CSs in the network is impossible, the use of BSVs can be quite helpful. Moreover, along with its significant application in practice, the proposed problem puts forward several theoretical challenges that will motivate future research.

5.2 Perspectives for further research

While a large number of research articles has appeared in the area of VRPs with explicit environmental considerations in a rather short time over the past 10 years, there are still multiple promising research directions in the field. Given the rapid pace at which technologies relevant to transportation are advancing, a natural direction for future research would be to exploit further these developments in the design of delivery routes to improve their business and environmental performance. To name a few of these advancements and their connection with VRPs, one can refer to the widespread availability of a large number of real-time data on traffic congestion from across the roadway network that makes real-time routing and re-routing easier than ever. Technologies related to the alternative fuel vehicles are also making a fast progress and new ideas are right now being investigated to significantly improve their driving range and to facilitate their refuelling.

Furthermore, some of the new ideas and directions for future research that each chapter of this thesis puts forward are as follows:

- The incorporation of the effect of non-recurrent congestion in the routing decisions through the development of real-time or stochastic variants of the SPRP is a significant line for future research.

- The proposed benchmark test instances and solution algorithms in chapter 3 of the paper could serve as a platform for investigating the effect of different perturbations in various characteristics of a logistic system, like customers' demands, locations, and time-windows, depot location, and vehicle fleet characteristics, and for carrying out various scenarios and what-if analyses.
- The proposed algorithms in chapter 3 are SPRP dedicated and despite the fact that the SPRP per se includes different variants of the VRPTW, such as the TDVRPTW, the fleet size and mix VRPTW, the MT-VRPTW, and the MO-VRPTW, extra programming effort and parameter tuning is required to modify the algorithms to solve specific instances related to those variants. To address this limitation, in further research, the proposed algorithms can be extended to be used as unified general multi-objective solvers that can address these rich variants with minimum user interference.
- Time-dependent VRPs are often based on the assumption that historical traffic data represent a rather repeating pattern of congestion. However, a key missing part is to validate the data prior to feeding them into the model. Hence, an attractive and promising line of future research would be to couple forecasting techniques with time-dependent routing tools to make the best out of both.
- In EVRPTWs the availability of a planned to visit CS on an ECV route is very crucial to the attainment of a feasible solution to complete the distribution task; if for any reason, the planned to visit CS becomes unavailable upon the arrival of the ECV, the entire routing plan can be disrupted and yielded infeasible. The use of ambulant BSVs in this context can be quite helpful. In this vein, future research can investigate a variant of EVRPTW with disruption.

As a solution to the high acquisition cost of ECVs and BSVs, multi-trip planning of ECVs and BSVs can help cutting down on vehicle hiring costs and can be an important future line of research.

Bibliography

Ahn, K., & Rakha, H. (2008). The effects of route choice decisions on vehicle energy consumption and emissions. *Transportation Research Part D: Transport and Environment, 13(3)*, 151-167.

Alice/Ertrac (2015). *Urban freight research roadmap*. Brussels.

Allen, J., Browne, M., & Piecyk, M. (2017). *Assessing the European Commission's target of essentially CO2-free city logistics in urban centres by 2030.* CityLab.

Alves, M. J., & Clı´maco, J. (2007). A review of interactive methods for multiobjective integer and mixed-integer programming. *European Journal of Operational Research, 180 (1)*, 99–115.

Anderluh, A., Hemmelmayr, V. C., & Nolz, P. C. (2017). Synchronizing vans and cargo bikes in a city distribution network. *Central European Journal of Operations Research, 25*(2), 345-376.

Androutsopoulos, K. N., & Zografos, K. G. (2012). A bi-objective time-dependent vehicle routing and scheduling problem for hazardous materials distribution. *EURO Journal on Transportation and Logistics 1 (1-2)*, 157-183.

Androutsopoulos, K. N., & Zografos, K. G. (2017). An integrated modelling approach for the bicriterion vehicle routing and scheduling problem with environmental considerations. *Transportation Research Part C: Emerging Technologies, 82*, 180-209.

Archetti, C., & Speranza, M. G. (2014). A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization, 2*(4), 223-246.

Archetti, C., & Speranza, M. G. (2012). Vehicle routing problems with split deliveries. *International Transactions in Operational Research, 19*(1-2), 3-22.

Banos, R., Ortega, J., Gil, C., Marquez, A. L., & de Toro, F. (2013). A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows. *Computers & Industrial Engineering, 65*(2), 286-296.

Barth, M., Scora, G., & Younglove, T. (2004). Modal emissions model for heavy-duty diesel vehicles. *Energy and Environmental Concerns 2004*(1880), 10-20.

Basso, R., KulcsÃ¡r, B., Egardt, B., Lindroth, P., & Sanchez-Diaz, I. (2019). Energy consumption estimation integrated into the electric vehicle routing problem. *Transportation Research Part D: Transport and Environment 69,* , 141-167.

Becerra, R. L., & Coello, C. A. C. (2006). *Solving hard multiobjective optimization problems using ε-constraint with cultured differential evolution.* Paper presented at the Parallel Problem Solving from Nature-PPSN IX, Berlin, Heidelberg.

Bektaş, T., Ehmke, J. F., Psaraftis, H. N., & Puchinger, J. (2019). The Role of Operational Research in Green Freight Transportation *European Journal of Operational Research, 274*(3), 807-823.

Bektaş, T., & Laporte, G. (2011). The Pollution-Routing Problem. *Transportation Research Part B 45 (8)*, 1232–1250.

Belliss, G. (2004). *Detailed Speed and Travel Time Surveys Using Low-Cost GPS Equipment.* Paper presented at the IPENZ, Wellington, New Zealand.

Ben Ticha, H., Absi, N., Feillet, D., & Quilliot, A. (2018). Vehicle routing problems with road-network information: State of the art. *Networks 72(3)*, 393-406.

Bent, R., & Van Hentenryck, P. (2004). A two-stage hybrid local search for the vehicle routing problem with time windows. *Transportation Science, 38*(4), 515-530.

Berbeglia, G., Cordeau, J., Gribkovskaia, I., & Laporte, G. (2007). Static pickup and delivery problems: A classification scheme and survey. *Top, 15*(1), 1-31.

Bokare, P. S., & Maurya, A. K. (2017). Acceleration-deceleration behaviour of various vehicle types. *Transportation research procedia 25*, 4733-4749.

Boland, N., Charkhgard, H., & Savelsbergh, M. (2016). The L-shape search method for triobjective integer programming. *Mathematical Programming and Computation, 8*, 217–251.

Boland, N., Charkhgard, H., & Savelsbergh, M. (2017). The Quadrant Shrinking Method: A simple and efficient algorithm for solving tri-objective integer programs. *European Journal of Operational Research, 260(3)*, 873-885.

Boulter, P. G., Barlow, T., McCrae, I., Latham, S. E., & Burgwal, V. D. (2006). *Road traffic characteristics, driving patterns and emission factors for congested situations.* Delft, The Netherlands: Tech. Rep., TNO Automotive, Department Powertrains-Environmental Studies & Testing.

Braekers, K., Ramaekers, K., & Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering, 99*, 300-313.

Brandao, J., & Mercer, A. (1998). The multi-trip vehicle routing problem. *Journal of the Operational Research Society, 49*(8), 799-805.

Brandao, J., & Mercer, A. (1997). A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *European Journal of Operational Research, 100*(1), 180-191.

Bräysy, O., & Gendreau, M. (2005a). Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation Science, 39*(1), 104-118.

Bräysy, O., & Gendreau, M. (2005b). Vehicle routing problem with time windows, part II: Metaheuristics. *Transportation Science, 39*(1), 119-139.

Bruglieri, M., Pezzella, F., Pisacane, O., & Suraci, S. (2015). A variable neighborhood search branching for the electric vehicle routing problem with time windows. *Electronic Notes in Discrete Mathematics, 47*, 221-228.

Byon, Y., Shalaby, A., & Abdulhai, B. (2006). *Travel time collection and traffic monitoring via GPS technologies.* Paper presented at the Proceedings of the 9th International IEEE Conference on Intelligent Transportation Systems (ITSC 2006), Toronto, Canada.

Campbell, A. M., & Wilson, J. H. (2014). Forty years of periodic vehicle routing. *Networks, 63*(1), 2-15.

Cattaruzza, D., Absi, N., & Feillet, D. (2016). The multi-trip vehicle routing problem with time windows and release dates. *Transportation Science, 50*(2), 676-693.

Cattaruzza, D., Absi, N., Feillet, D., & Vidal, T. (2014). A memetic algorithm for the multi trip vehicle routing problem. *European Journal of Operational Research, 236*(3), 833-848.

Cattaruzza, D., Absi, N., Feillet, D., & Vigo, D. (2014). An iterated local search for the multi-commodity multi-trip vehicle routing problem with time windows. *Computers & Operations Research, 51*, 257-267.

Christofides, N., Mingozzi, A., & Toth, P. (1981a). Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming, 20*(1), 255-282.

Christofides, N., Mingozzi, A., & Toth, P. (1981b). State-space relaxation procedures for the computation of bounds to routing problems. *Networks, 11*(2), 145-164.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*(4), 568-581.

Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems* (Vol. 5): Springer.

Committee on Climate Change. (2010). *The fourth carbon budget - reducing emissions through the 2020s*. Retrieved from: https://www.theccc.org.uk/archive/aws2/4th%20Budget/CCC-4th-Budget-Book_interactive_singles.pdf

Conrad, R. G., & Figliozzi, M. A. (2011). The recharging vehicle routing problem. Paper presented at the *Proceedings of the 2011 Industrial Engineering Research Conference,* Reno, NV. 1-8.

Cornuéjols, G., Fonlupt, J., & Naddef, D. (1985). The traveling salesman problem on a graph and some related integer polyhedra. *Mathematical programming 33 (1)*, 1-27.

Crainic, T. G., Ricciardi, N., & Storchi, G. (2009). Models for evaluating and planning city logistics systems. *Transportation Science, 43*(4), 432-454.

Dabia, S., Demir, E., & Woensel, T. V. (2016). An exact approach for a variant of the pollution-routing problem. *Transportation Science, 51*(2), 607-628.

Dablanc, L. (2009). *Freight transport for development toolkit: Urban freight*. Retrieved from Washington DC:

Dächert, K., & Klamroth, K. (2015). A linear bound on the number of scalarizations needed to solve discrete tricriteria optimization problems. *Journal of Global Optimization, 61(4)*, 643-676.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science, 6*(1), 80-91.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Ieee Transactions on Evolutionary Computation, 6*(2), 182-197.

Demir, E., Bektaş, T., & Laporte, G. (2014a). The bi-objective Pollution-Routing Problem. *European Journal of Operational Research, 232*(3), 464-478.

Demir, E., Bektaş, T., & Laporte, G. (2014b). A review of recent research on green road freight transportation. *European Journal of Operational Research, 237*(3), 775-793.

Demir, E., Bektaş, T., & Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research 223 (2)*, 346-359.

Department for Transport. (2017). *Freight carbon review 2017: Moving britain ahead.* (). London: Department for Transport. Retrieved from https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/590922/freight-carbon-review-2017.pdf

Desaulniers, G., Errico, F., Irnich, S., & Schneider, M. (2016). Exact Algorithms for Electric Vehicle-Routing Problems with Time Windows. *Operations Research, 64*(6), 1388-1405.

Dhaenens, M., Lemesre, J., & Talbi, E. G. (2010). K-PPM: a new exact method to solve multi-objective combinatorial optimization problems. *European Journal of Operational. Research 200 (1)*, 45–53.

Doerner, K. F., & Schmid, V. (2010). Survey: Matheuristics for rich vehicle routing problems. Paper presented at the *International Workshop on Hybrid Metaheuristics,* 206-221.

Dokka, T., & Goerigk, M. (2017). *An Experimental Comparison of Uncertainty Sets for Robust Shortest Path Problems.* Paper presented at the

Proceedings of the 17th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS2017).

Dorigo, M., & Di Caro, G. (1999). Ant colony optimization: A new meta-heuristic. Paper presented at the *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. no. 99TH8406), 2* 1470-1477.

Drexl, M. (2012). Synchronization in vehicle routing-A survey of VRPs with multiple synchronization constraints. *Transportation Science, 46*(3), 297-316.

Durillo, J. J., & Nebro, A. J. (2011). jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software, 42*(10), 760-771.

Ehmke, J. F., Campbell, A. M., & Thomas, B. W. (2018). Optimizing for total costs in vehicle routing in urban areas. *Transportation Research Part E-Logistics and Transportation Review, 116*, 242-265.

Ehmke, J. F., Campbell, A. M., & Thomas, B. W. (2016). Vehicle routing to minimize time-dependent emissions in urban areas. *European Journal of Operational Research 251 (2)*, 478–494.

Ehrgott, M. (2005). *Multicriteria optimization* (Second ed.): Springer.

Eksioglu, B., Vural, A. V., & Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering, 57*(4), 1472-1483.

Erdoğan, S., & Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation Research Part E-Logistics and Transportation Review, 48*(1), 100-114.

European Commission. (2011). *Roadmap to a single european transport area: Towards a competitive and resource-efficient transport system.* Retreived from: (https://ec.europa.eu/transport/sites/transport/files/themes/strategies/doc/2011_white_paper/white-paper-illustrated-brochure_en.pdf)

Felipe, A., Ortuno, M. T., Righini, G., & Tirado, G. (2014). A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E-Logistics and Transportation Review, 71*, 111-128.

Figliozzi, M. A. (2010). Vehicle routing problem for emissions minimization. *Transportation Research Record,* (2197), 1-7.

Figliozzi, M. A. (2011). The impacts of congestion on time-definitive urban freight distribution networks CO2 emission levels: Results from a case study in portland, oregon. *Transportation Research Part C-Emerging Technologies, 19*(5), 766-778.

Fonseca, C. M., Knowles, J. D., Thiele, L., & Zitzler, E. (2005). *A tutorial on the performance assessment of stochastic multiobjective optimizers*. Paper presented at the Third international conference on evolutionary multi-criterion optimization (EMO).

Franceschetti, A., Demir, E., Honhon, D., Van Woensel, T., Laporte, G., & Stobbe, M. (2017). A metaheuristic for the time-dependent pollution-routing problem. *European Journal of Operational Research, 259*(3), 972-991.

Franceschetti, A., Honhon, D., Van Woensel, T., Bektaş, T., & Laporte, G. (2013). The time-dependent pollution-routing problem. *Transportation Research Part B-Methodological, 56*, 265-293.

Fukasawa, R., Longo, H., Lysgaard, J., de Aragão, M. P., Reis, M., Uchoa, E., & Werneck, R. F. (2006). Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming, 106*(3), 491-511.

Gao, X. J., Zhao, J. L., Shang, W. Z., Wang, T. B., & Wang, X. (2012). *A mobile electric vehicle battery swapping van for emergency* (Patent no: CN 201120341622.2)

Garaix, T., Artigues, C., Feillet, D., & Josselin, D. (2010). Vehicle routing problems with alternative paths: An application to on-demand transportation. *European Journal of Operational Research, 204*(1), 62-75.

Garcia-Najera, A., & Bullinaria, J. A. (2011). An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. *Computers & Operations Research, 38*(1), 287-300.

Gendreau, M., Ghiani, G., & Guerriero, E. (2015). Time-dependent routing problems: A review. *Computers & Operations Research, 64*, 189-197.

Gendreau, M., & Tarantilis, C. D. (2010). *Solving large-scale vehicle routing problems with time windows: The state-of-the-art.* Unpublished manuscript.

Ghoseiri, K., & Ghannadpour, S. F. (2010). Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. *Applied Soft Computing, 10*(4), 1096-1107.

Gillett, B. E., & Miller, L. R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations Research, 22*(2), 340-349.

Glover, F. (1989). Tabu search—part I. *ORSA Journal on Computing, 1*(3), 190-206.

Goeke, D., & Schneider, M. (2015). Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research, 245*(1), 81-99.

Golden, B. L., Raghavan, S., & Wasil, E. A. (2008). *The vehicle routing problem: Latest advances and new challenges* Springer Science & Business Media.

Grangier, P., Gendreau, M., Lehuédé, F., & Rousseau, L. (2017). A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking. *Computers & Operations Research, 84*, 116-126.

Grangier, P., Gendreau, M., Lehuede, F., & Rousseau, L. M. (2016). An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. *European Journal of Operational Research, 254*(1), 80-91.

Güner, A. R., Murat, A., & Chinnam, R. B. (2012). Dynamic routing under recurrent and non-recurrent congestion using real-time ITS information. *Computers & Operations Research, 39*(2), 358-373.

Hamacher, H. W., Ruzika, S., & Tjandra, S. A. (2006). Algorithms for time-dependent bicriteria shortest path problems. *Discrete optimization 3 (3)*, 238-254.

Hiermann, G., Puchinger, J., Ropke, S., & Hartl, R. F. (2016). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research, 252*(3), 995-1018.

Highways England. (2018). Highways agency network journey time and traffic flow data.

Hill, A. V., & Benton, W. (1992). Modelling intra-city time-dependent travel speeds for vehicle scheduling problems. *Journal of the Operational Research Society, 43*(4), 343-351.

Hof, J., Schneider, M., & Goeke, D. (2017). Solving the battery swap station location-routing problem with capacitated electric vehicles using an AVNS algorithm for vehicle-routing problems with intermediate stops. *Transportation Research Part B-Methodological, 97*, 102-112.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence* MIT press.

Horn, M. E. (2000). Efficient modeling of travel in networks with time-varying link speeds. *Networks 36 (2)*, 80-90.

Huang, Y. X., Zhao, L., Van Woensel, T., & Gross, J. P. (2017). Time-dependent vehicle routing problem with path flexibility. *Transportation Research Part B-Methodological, 95*, 169-195.

Ichoua, S., Gendreau, M., & Potvin, J.-Y. (2003). Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research 144 (2)*, 379-396.

Jozefowiez, N., Semet, F., & Talbi, E. G. (2008). Multi-objective vehicle routing problems. *European Journal of Operational Research, 189*(2), 293-309.

Kancharla, S. R., & Ramadurai, G. (2018). Incorporating driving cycle based fuel consumption estimation in green vehicle routing problems. *Sustainable Cities and Society (40)*, 214-221.

Kara, I., Kara, B. Y., & Yetis, M. K. (2007). *Energy minimizing vehicle routing problem.* Paper presented at the International Conference on Combinatorial Optimization and Applications, Berlin, Heidelberg.

Keskin, M., & Çatay, B. (2016). Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C-Emerging Technologies, 65*, 111-127.

Kirkpatrick, S., Gelatt, C. D.,Jr, & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science (New York, N.Y.), 220*(4598), 671-680.

Kirlik, G., & Sayın, S. (2014). A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. *European Journal of Operational Research 232(3)*, 479–488.

Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2014). The fleet size and mix pollution-routing problem. *Transportation Research Part B-Methodological, 70*, 239-254.

Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2015). A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows. *Computers & Operations Research, 64*, 11-27.

Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2016). Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research, 249*(1), 1-21.

Kramer, R., Subramanian, A., Vidal, T., & Lucídio dos Anjos, F Cabral. (2015a). A matheuristic approach for the pollution-routing problem. *European Journal of Operational Research, 243*(2), 523-539.

Kramer, R., Maculan, N., Subramanian, A., & Vidal, T. (2015b). A speed and departure time optimization algorithm for the pollution-routing problem. *European Journal of Operational Research, 247*(3), 782-787.

Lai, D. S. W., Demirag, O. C., & Leung, J. M. Y. (2016). A tabu search heuristic for the heterogeneous vehicle routing problem on a multi-graph. *Transportation Research Part E-Logistics and Transportation Review, 86*, 32-52.

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science, 43*(4), 408-416.

Laporte, G., Nobert, Y., & Desrochers, M. (1985). Optimal routing under capacity and distance restrictions. *Operations Research, 33*(5), 1050-1073.

Lee, R. J., Sener, I. N., & Mullins III, J. A. (2016). An evaluation of emerging data collection technologies for travel demand modeling: from research to practice. *Transportation Letters 8 (4)*, 181-193.

Letchford, A. N., Nasiri, S. D., & Theis, D. O. (2013). Compact formulations of the Steiner Traveling Salesman Problem and related problems. *European Journal of Operational Research, 228*(1), 83-92.

Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell System Technical Journal, 44*(10), 2245-2269.

Lin, C. H., Choy, K. L., Ho, G. T. S., Chung, S. H., & Lam, H. Y. (2014). Survey of Green Vehicle Routing Problem: Past and future trends. *Expert Systems with Applications, 41*(4), 1118-1138.

Liu, S. G., Huang, W. L., & Ma, H. M. (2009). An effective genetic algorithm for the fleet size and mix vehicle routing problems. *Transportation Research Part E-Logistics and Transportation Review, 45*(3), 434-445.

Lokman, B., & Köksalan, M. (2013). Finding all nondominated points of multi-objective integer programs. *Journal of Global Optimisation 57 (2)*, 347–365.

Lu, G. J., & Zhou, Y. P. (2013). *A electric vehicle with bottom lateral linkage battery swapping and its battery swapping devices* (Patent no: CN 201310164242.X)

Macedo, R., Alves, C., de Carvalho, J. M. V., Clautiaux, F., & Hanafi, S. (2011). Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model. *European Journal of Operational Research, 214*(3), 536-545.

Macrina, G., Pugliese, L. D. P., Guerriero, F., & Laporte, G. (2018). The green mixed fleet vehicle routing problem with partial battery recharging and time windows. *Computers & Operations Research, 101*, 183-199.

Malandraki, C., & Daskin, M. S. (1992). Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transportation Science, 26*(3), 185-200.

Malandraki, C., & Dial, R. B. (1996). A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem. *European Journal of Operational Research, 90*(1), 45-55.

Manthey, N. (2017). Nio launches EV with battery swap and mobile charging option. Retrieved from https://www.electrive.com/2017/12/19/nio-launches-ev-battery-swap-mobile-charging-option/

Martins, E. Q. V. (1984). On a Multicriteria Shortest-Path Problem. *European Journal of Operational Research, 16*(2), 236-245.

MDS Transmodal. (2012). *DG MOVE European Commission: Study on urban freight transport. Final report.*

Montoya, A., Gueret, C., Mendoza, J. E., & Villegas, J. G. (2017). The electric vehicle routing problem with nonlinear charging function. *Transportation Research Part B-Methodological, 103*, 87-110.

Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F., & Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering, 79*, 115-129.

Olivera, A., & Viera, O. (2007). Adaptive memory programming for the vehicle routing problem with multiple trips. *Computers & Operations Research, 34*(1), 28-47.

Ombuki, B., Ross, B. J., & Hanshar, F. (2006). Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence, 24*(1), 17-30.

Or, I. (1977). *Traveling salesman type combinatorial problems and their relation to the logistics of regional blood banking.* PhD dissertation.

Orda, A., & Rom, R. (1990). Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *Journal of the Association for Computing Machinery 37 (3)*, 607-625.

Özlen, M., Burton, B. A., & MacRae, C. A. G. (2014). Multi-objective integer programming: an improved recursive algorithm. *Journal of optimisation theory and applications, 160(2)*, 470–482.

Palmer, A. (2007). The Development of an integrated routing and carbon dioxide emissions model for goods vehicles. *Ph.D. thesis, Cranfield University, School of Management.*

Pelletier, S., Jabali, O., & Laporte, G. (2016). Goods distribution with electric vehicles: Review and research perspectives. *Transportation Science, 50*(1), 3-22.

Petch, R. J., & Salhi, S. (2003). A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. *Discrete Applied Mathematics, 133*(1-3), 69-92.

Pillac, V., Gendreau, M., Guéret, C., & Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research, 225*(1), 1-11.

Pisinger, D., & Ropke, S. (2010). Large neighborhood search. *Handbook of metaheuristics* (pp. 399-419) Springer.

Potvin, J., & Rousseau, J. (1993). A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research, 66*(3), 331-340.

Prins, C. (2009). Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence, 22*(6), 916-928.

Prins, C., Lacomme, P., & Prodhon, C. (2014). Order-first split-second methods for vehicle routing problems: A review. *Transportation Research Part C-Emerging Technologies, 40*, 179-200.

Qi, Y. T., Hou, Z. T., Li, H., Huang, J. B., & Li, X. D. (2015). A decomposition based memetic algorithm for multi-objective vehicle routing problem with time windows. *Computers & Operations Research, 62*, 61-77.

Qian, J., & Eglese, R. (2016). Fuel emissions optimization in vehicle routing problems with time-varying speeds. *European Journal of Operational Research 248 (3)*, 840–848.

Raeesi, R., & O'Sullivan, M. J. (2014). Eco-logistics: Environmental and economic implications of alternative fuel vehicle routing problem. *International Journal of Business Performance and Supply Chain Modelling, 6*(3-4), 276-297.

Raeesi, R., & Zografos, K. (2018). *Modeling and Solving the Multitrip Multiobjective Pollution Routing Problem on Urban Road Networks.* Paper presented at the Annual Meeting of the Transportation Research Board (TRB), Washington, DC United States.

Raeesi, R., & Zografos, K. G. (2019). The multi-objective Steiner pollution-routing problem on congested urban road networks. *Transportation Research Part B: Methodological, 122*, 457-485.

Ranjithan, S. R., Chetan, S. K., & Dakshina, H. K. (2001). *Constraint method-based evolutionary algorithm (CMEA) for multiobjective optimization.* Paper presented at the International Conference on Evolutionary Multi-Criterion Optimization, Berlin, Heidelberg.

Ritchie, Hanna, and Max Roser. 2018. Urbanization. Published online at OurWorldInData.org Retrieved from: https://ourworldindata.org/urbanization [Online Resource].

Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science, 40*(4), 455-472.

Salhi, S., & Petch, R. J. (2007). A GA based heuristic for the vehicle routing problem with multiple trips. *Journal of Mathematical Modelling and Algorithms 6(4)*, 591-613.

Salimifard, K., & Raeesi, R. (2014). A green routing problem: Optimising CO2 emissions and costs from a bi-fuel vehicle fleet. *International Journal of Advanced Operations Management, 6*(1), 27-57.

Schneider, M., Stenger, A., & Goeke, D. (2014). The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations. *Transportation Science, 48*(4), 500-520.

Shao, S. J., Guo, S. Y., & Qiu, X. S. (2017). A mobile battery swapping service for electric vehicles based on a battery swapping van. *Energies, 10*(10)

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. *Principles and Practice of Constraint Programming - Cp98, 1520*, 417-431.

Silvas, E., Hereijgers, K., Peng, H., Hofman, T., & Steinbuch, M. (2016). Synthesis of realistic driving cycles with high accuracy and computational speed, including slope information. *IEEE Transactions on Vehicular Technology 65 (6)*, 4118-4128.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research, 35*(2), 254-265.

Srigiriraju, C. K. (2000). *Noninferior surface tracing evolutionary algorithm (NSTEA) for multiobjective optimization.* (M.S.), North Carolina State University, Raleigh, N.C.

Sylva, J., & Crema, A. (2004). A method for finding the set of non-dominated vectors for multiple objective integer linear programs. *European Journal of Operational Research, 158*, 46–55.

Taillard, E. D., Laporte, G., & Gendreau, M. (1996). Vehicle routeing with multiple use of vehicles. *Journal of the Operational Research Society, 47*(8), 1065-1070.

Tan, K. C., Chew, Y. H., & Lee, L. H. (2006). A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems. *European Journal of Operational Research, 172*(3), 855-885.

Toth, P., & Vigo, D. (2014). *Vehicle Routing: Problems, Methods, and Applications, Second Edition.* Philadelpia: SIAM.

Transport for London. (2007). *London freight plan: Sustainable freight distribution: A plan for london.*

Turkensteen, M. (2017). The accuracy of carbon emission and fuel consumption computations in green vehicle routing. *European Journal of Operational Research 262 (2)*, 647-659.

Ubeda, S., Arcelus, F. J., & Faulin, J. (2011). Green logistics at Eroski: A case study. *International Journal of Production Economics 131 (1)*, 44 –51.

Van Veldhuizen, D. A., & Lamont, G. B. (1998). *Multiobjective evolutionary algorithm research: A history and analysis.* Wright-Patterson AFB, Ohio.

Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., & Velasco, N. (2013). A matheuristic for the truck and trailer routing problem. *European Journal of Operational Research, 230*(2), 231-244.

Wang, M., Daamen, W., Hoogendoorn, S., & Van Arem, B. (2011). Estimating acceleration, fuel consumption, and emissions from macroscopic traffic flow data. *Transportation Research Record: Journal of the Transportation Research Board 2260*, 123-132.

Xiao, Y., & Konak, A. (2016). The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion. *Transportation Research Part E 88*, 146-166.

Xiao, Y., Zhao, Q., Kaku, I., & Xu, Y. (2012). Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers & Operations Research 39 (7)*, 1419-1431.

Yang, J., & Sun, H. (2015). Battery swap station location-routing problem with capacitated electric vehicles. *Computers & Operations Research, 55*, 217-232.

Zegeye, S. K., De Schutter, B., Hellendoorn, J., Breunesse, E. A., & Hegyi, A. (2013). Integrated macroscopic traffic flow, emission, and fuel consumption model for control purposes. *Transportation Research Part C 31*, 158-171.

Ziliaskopoulos, A. K., & Mahmassani, H. S. (1993). Time-dependent, shortest-path algorithm for real-time intelligent vehicle highway system applications. *Transportation Research Record: Journal of the Transportation Research Board 1408*, 94-100.

Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation, 8*(2), 173-195.

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *Ieee Transactions on Evolutionary Computation, 3*(4), 257-271.