Optimization Letters (2021) 15:2813–2829
https://doi.org/10.1007/s11590-021-01714-3

ORIGINAL PAPER

A more efficient cutting planes approach for the green vehicle routing problem with capacitated alternative fuel stations

Maurizio Bruglieri[1] · Simona Mancini[2] · Ornella Pisacane[3]

Received: 26 December 2019 / Accepted: 11 February 2021 / Published online: 1 March 2021

Abstract
The Green Vehicle Routing Problem with Capacitated Alternative Fuel Stations assumes that, at each station, the number of vehicles simultaneously refueling cannot exceed the number of available pumps. The state-of-the-art solution method, based on the generation of all feasible non-dominated paths, performs well only with up to 2 pumps. In fact, it needs cloning the paths between every pair of pumps. To overcome this issue, in this paper, we propose new path-based MILP models without cloning paths, for both the scenario with private stations (i.e., owned by the fleet manager) and that with public stations. Then, a more efficient cutting plane approach is designed for addressing both the scenarios. Numerical results, obtained considering a set of benchmark instances ad hoc generated for this work, show both the efficiency and the effectiveness of this new cutting plane approach proposed. Finally, a sensitivity analysis, carried out by varying the number of customers to be served and their distribution, shows very good performances of the proposed approach.

Keywords Vehicle routing problem · Alternative fuel vehicles · Mixed integer linear programming · Fueling pump reservation

1 Introduction

The recent worries about the current environmental and climatic conditions of our planet are increasing the interest in the *Alternative Fuel Vehicles* (AFVs), i.e., vehicles that use alternative fuel like, methanol, electricity and natural gas.

✉ Maurizio Bruglieri
maurizio.bruglieri@polimi.it

[1] Dipartimento di Design, Politecnico di Milano, Milan, Italy

[2] Department of Operations, Energy, and Environmental Management, University of Klagenfurt, Klagenfurt, Austria

[3] Dipartimento di Ingegneria dell’Informazione, Universitá Politecnica delle Marche, Ancona, Italy

Moreover, the recent rules imposed by the European Commission aim at supporting the diffusion of the AFVs in place of the traditional Internal Combustion Engine Vehicles. A report by the International Energy Agency states that, only in 2018, the global electric car fleet overcame 5.1 million, about the double of 2017 [14]. However, one of the major drawback of using AFVs is that they may require to be refueled at the *Alternative Fuel Stations* (AFSs), also more than once, during a trip. This aspect, together with the fact that the AFSs are currently not widespread across the territory, leads to the need of properly routing the AFVs in order to avoid drivers remaining stuck during their trips.

The *Green Vehicle Routing Problem* (G-VRP) represents a very recent research field that is attracting attention of many researchers and aims at efficiently routing a fleet of AFVs, to serve a set of customers. In a route, each AFV starts from a common depot, where the fleet is based, serves a subset of customers, with possible fueling stops at AFSs and returns to the depot. The implicit hypothesis traditionally made is that each AFS s has an unlimited capacity, i.e., an unlimited number of AFVs may be refueled simultaneously at s.

The G-VRP has been efficiently solved through a path-based approach in the recent work [5]. Specifically, a *path* represents a leg of a route joining the depot or an AFS with the depot or another AFS and serving a subset of customers (possibly also empty), without intermediate stops at AFSs. In order to reduce the number of all feasible paths, dominance criteria among them are also defined.

To take into account that each AFS has actually a limited number of fueling pumps and then, only a limited number of AFVs can be simultaneously refueled at it, in [6], a new variant of the G-VRP has been proposed, i.e., the GVRP with Capacitated AFS (G-VRP-CAFS) where, for each AFS s, η_s fueling pumps are available. Two different scenarios are considered: one with public AFSs, for which the reservation of the fueling pumps is introduced and one with private AFSs, i.e., AFSs owned by the manager of the AFVs fleet. A variant of the path-based approach is proposed, where multiple fueling pumps at AFS are managed by properly cloning the paths, in both scenarios. Moreover, Cutting Planes (CPs) methods to efficiently solve the path-based model are also designed.

The current work addresses the same problem providing the following main contributions: (1) a new variant of the path-based approach without cloning the paths is proposed; (2) a more efficient CPs approach is designed; (3) a set of instances challenging for the case under consideration is generated; (4) an accurate sensitivity analysis is carried out in order to discuss the performances of the proposed approach by varying the number of customers and their distribution.

The rest of the paper is organized as follows. In Sect. 2, a short literature review is presented. In Sect. 3, the notation and the statement of the problem are introduced. In Sect. 4, the proposed path based formulation without cloning paths is presented for the scenario with private AFSs. In Sects. 5 and 6, the CP approaches for the scenarios with private AFSs and public AFSs, respectively, are outlined. In Sect. 7 the results are discussed. Finally, Sect. 8 draws some conclusions and remarks possible future research directions worthy of investigation.

2 Literature review

The G-VRP belongs to the class of the Vehicle Routing Problems (VRPs) where the fleet consists of only AFVs. It is introduced in [9] where it is firstly mathematically formulated via Mixed Integer Linear Programming (MILP) and on large-sized instances, it is solved through both a modified Clark & Wright Saving algorithm and a Density-Based Clustering approach.

The case in which the fleet consists of only Electric Vehicles (EVs) is instead addressed for the first time in [25] where together with the need of the EVs to be recharged during the routes, time windows at customers and EV loading capacity are both taken into account, introducing the so-called Electric Vehicle Routing Problem with Time Windows (E-VRPTW). The authors mathematically model it via MILP and solve large-sized instances combining a Variable Neighboord Search with a Tabu Search. In [26], this problem is addressed as a VRP with Intermediate Stops introduced in the literature by the work of [3].

An extended version of the G-VRP is described in [10] where the authors introduce also the possibility to use several alternative recharge technologies, each with a different cost. For such a new variant of the G-VRP, both constructive and improving heuristics are designed and embedded in a Simulating Annealing algorithm. Whereas the possibility of partially recharging the EVs is introduced in [4]. In particular, the authors firstly propose a MILP formulation of the E-VRPTW with partial recharges considering that the battery level reached by each EV after visiting an AFS is a decision variable, minimizing simultaneously the total travel time, the total waiting time, the total recharging time and the number of the employed EVs. A Variable Neighboord Search Branching matheuristic is also proposed for addressing large-sized instances.

The problem of both sizing the EV fleet and then, routing them considering possible stops at the stations is addressed in [12] where it is modelled via Mixed Integer Program (MIP). In addition, the authors also propose a set-partitioning model solved through a state-of-the-art branch-and-price algorithm designed for the VRP with time windows. Whereas in order to efficiently address large-sized instances, they design an Adaptive Large Neighbourhood Search meta-heuristic together with embedded local search and labeling procedures. Instead, the recent work of [13] addresses the problem of routing a mixed fleet, made up by conventional, plug-in hybrid and EVs, through a genetic algorithm integrated with a local large neighboord search.

In [17], the G-VRP is reformulated by using the Miller Tucker Zemlin capacity and subtour elimination constraints proposed for the Traveling Salesman Problem and solved through a branch-and-bound algorithm. A new G-VRP MILP formulation and a reduction procedure are then proposed in [18] where medium-sized instances are also solved. In [1], the G-VRP is modelled as a set-partitioning problem where the routes become simple circuits on a multi-graph. Each node of the multi-graph represents a customer and an arc between two customers i and j is a non-dominated path of AFSs visited by the vehicle travelling from i to j. The set-partitioning model is further strengthen by introducing valid inequalities. More efficient formulations and valid inequalities for the G-VRP are recently proposed in [7] where consecutive stops at AFSs are also allowed.

A two-phase heuristic approach is designed in [22] where feasible routes are firstly built through a randomized route-first cluster-second algorithm and then, an efficient AFS insertion procedure is used. The set of routes belonging to the final G-VRP solution are selected through a set-partitioning model. However, in [2], a multi-start local search heuristic is designed suitable to outperform all the existing meta-heuristics.

Several variants of both the G-VRP and the E-VRPTW have been introduced in the literature. For example, in [8], the authors address the case in which only one recharge per route is permitted and that in which this hypothesis is relaxed. Each of both scenarios is also studied either permitting the partial recharges or assuming only full recharges. An E-VRPTW with pick-ups and deliveries is instead addressed in [11]. Similarly, a G-VRP with pick-ups, deliveries and time windows at customers is proposed in [19]. Whereas in [20], the routing problem of a fleet of hybrid vehicles is addressed, i.e., allowing vehicles switching from the use of the traditional fuel to electricity during their trips. In addition, multi-objective G-VRPs are addressed in [15,24] and [23].

As already described in Sect. 1, recently in [5], the G-VRP is addressed through a path-based approach. First, all feasible non-dominated paths, serving customers between AFSs or between an AFS and the depot, are found. Then, a path-based mathematical model selects the paths to be in the optimal solution of the G-VRP. This approach is proven to outperform all the existing exact methods on small/medium-sized instances. Whereas on large-sized instances, feasible non-dominated paths are heuristically generated by applying the algorithm proposed in [22].

However, all the aforementioned works make the hypothesis that each station has unlimited capacity and then, that an unlimited number of vehicles can be refueled simultaneously at it. To the best of our knowledge, the first contribution, in which the realistic assumption of having capacitated stations is done, is that of [16]. Indeed, the authors address an E-VRPTW with capacitated recharging stations, by assuming that an EV may have to wait at the station if all chargers are busy. Then, a MILP formulation with time-dependent waiting times at the stations is proposed, minimizing delays at customers due to possible queues at the stations, driver costs, vehicle costs and also energy. Whereas large-sized instances are addressed through an Adaptive Large Neighborhood Search.

The G-VRP-CAFS is instead introduced in [6] where it is discussed how, depending on the location of customers, depot and AFSs, the station capacity may become a crucial issue to be considered during the AFV routing. Together with an Arc-based MILP formulation, the authors propose a variant of the path-based model described in [5] where the paths generation procedure is properly adapted to the context with capacitated AFSs. In particular, as many clones of each path starting from a station s_1 and arriving to a station s_2 as $\eta_{s_1} \cdot \eta_{s_2}$ are generated. In order to efficiently address the path-based formulation on large-sized instances, the authors also design two variants of the CPs method. Moreover, they also generate a benchmark set of challenging instances for the G-VRP-CAFS. However, when the number of pumps at AFSs grows, the proposed methods become impracticable and in fact, the experimental campaign is carried out on only cases with at most 2 fueling pumps per station. This paper overcomes this issue by properly extending the previous CPs approaches in order to address also cases with more than 2 fueling pumps per station. Indeed, the advantage of the approach proposed in this paper, is that it does not require to duplicate paths and stations to handle capacities greater than 1, unlike the method proposed in [6]. Therefore the newly proposed approach is the first method able to solve also instances with larger capacity, while [6] can only address capacity up to 2. The main novelty is that station capacity greater than 1 is directly handled imposing a maximum number of refueling operations overlapping, while in [6] each single pump was separately considered imposing that no simultaneous refueling can occur on it.

3 Problem statement

The G-VRP-CAFS is represented on a complete directed graph $G = (N, A)$ where $N = I \cup F \cup \{0\}$ contains the set of customers I, the set of AFSs F and the depot (indicated by 0), and A is the set of the arcs. For each pair of nodes $(i, j) \in A$, both the travel time t_{ij} and the travel distance d_{ij} are known. Each route that starts from the depot and ends at it cannot exceed the maximum duration T_{max}. Each AFV has a limited fuel capacity Q. If the depot is also an AFS, it is indicated by s_0. It is assumed that each AFV leaves the depot fully refueled and the time spent for the initial refuel is indicated by p^{start}. Moreover, for each station $s \in F$, the refueling time p_s is given and for each customer $i \in I$, the service time p_i is also given. A fictitious service time $p_0 = 0$ is assigned also to the depot, when it is not considered as an AFS. The fuel consumption rate is assumed to be proportional to the travel distance through the rate r and therefore, an AFV cannot travel longer than $D_{max} = \frac{Q}{r}$ without stopping at an AFS. In the G-VRP-CAFS, in addition to the aforementioned parameters, the number of fueling pumps for each AFS s, denoted by η_s, is also known.

4 Path based formulation for the scenario with private AFSs

For the G-VRP-CAFS with private AFSs, we propose a path-based MIP formulation. As in [6], this formulation is based on the idea that each feasible route can be seen as the combination of *paths*, each one handling a subset (also empty) of customers without intermediate stops at AFSs. Each path can link the depot or an AFS with the depot or another AFS. In particular, the path linking the depot with itself represents a route without intermediate stops. For each path k, the origin (starting node) s_k, the destination (arrival node) a_k, the travel distance d_k and the duration γ_k are known. In particular, γ_k is computed as the sum of the total travel time of path k and the service times at nodes visited by it. Let I_k the set of customers served by path k.

The set K of all feasible non-dominated paths can be generated according to Algorithm 1 of [6]. In particular, starting from all feasible paths in which only one customer is served, all feasible non-dominated paths of increased cardinality are generated by adding a new customer at a time. The path generation procedure is made more efficient thanks to the use of appropriate data structures (e.g., Hash Maps). Specifically, a path k is feasible (*feasibility rules*) if its total distance is less than or equal to D_{max} and its total duration, including the service times, is less than or equal to T_{max}. A feasible path k_1 dominates a feasible path k_2 (*dominance rules*) if they start/arrive from/to the same station, handle the same customers and the total distance of the former is less than or equal to that of the latter.

Moreover, a dummy path, k_0, which starts from and ends at the depot without visiting any customer, with duration $\gamma_{k_0} = 0$ and length $d_{k_0} = 0$ is generated. This path is used in the model to ensure routes continuity and to force the number of routes to be lower than the number m of available AFVs. We denote by $\tilde{K} = K \cup k_0$. Unlike [6], where each path $k \in K$ was duplicated $\eta_{s(k)} \cdot \eta_{a(k)}$ times to take into account multiple pumps in each station, in the new formulation here proposed, we do not create clones of them. A coverage parameter, c_{ik}, is also introduced, equal to 1 if $i \in I$ is handled in $k \in \tilde{K}$, 0 otherwise. From $\tilde{K}$, the set P of all the pairs of compatible paths is generated. A pair of paths $(k_1, k_2), k_1 \in \tilde{K}, k_2 \in \tilde{K}, k_1 \neq k_2$ is defined compatible if $\tilde{a}_{k_1} = \tilde{s}_{k_2}$, the sets of customers handled in the two paths are disjoint and $t_{0\tilde{s}_{k_1}} + \gamma_{k_1} + \gamma_{k_2} + t_{\tilde{a}_{k_2}0} + p_{\tilde{s}_{k_1}} + p_{\tilde{a}_{k_1}} + p_{\tilde{a}_{k_2}} + p^{start} \leq T_{max}$.

The following decision variables are introduced:

- Z_k, equal to 1 if $k \in \tilde{K}$ is selected, 0 otherwise;
- $X_{k_1k_2}$, equal to 1 if $k_2 \in \tilde{K}$ is covered just after $k_1 \in \tilde{K}$, 0 otherwise, $\forall (k_1, k_2) \in P$;
- T_k a non-negative variable representing the refueling start time of $k \in K$ at its final node a_k if $a_k \neq 0$, the arrival time of k to the depot if $a_k = 0$. Whereas for the dummy path k_0, $T_{k_0} = p^{start}$.
- $Y_{k_1k_2}$, equal to 1 if path k_1 reaches the AFS s before k_2 and 0 otherwise, $\forall k_1, k_2 \in K : a_{k_1} = a_{k_2} = s \neq 0$, $I_{k_1} \cap I_{k_2} = \emptyset$
- $\xi_{k_1k_2}$ equal to 1 if the starting refueling times of paths k_1 and k_2 at their final node s differ less or equal than p_s, 0 otherwise, $\forall k_1, k_2 \in K : a_{k_1} = a_{k_2} = s \neq 0$, $I_{k_1} \cap I_{k_2} = \emptyset$

Moreover, $\forall s \in F$, we indicate by $\mathcal{H}_s$ the family of all sets of $\eta_s + 1$ customer disjoint paths belonging to K with the same arrival AFS s.

The path-based MIP model for the scenario with private AFSs is the following:

$$\min \sum_{k \in K} d_k Z_k \tag{1}$$

$$\sum_{k \in K} c_{ik} Z_k = 1 \quad \forall i \in I \tag{2}$$

$$\sum_{k_2 \in \tilde{K}} X_{k_0k_2} \leq m \tag{3}$$

$$X_{k_1k} = \sum_{\substack{k_2 \in \tilde{K}: \\ (k,k_2) \in P}} X_{kk_2} \quad \forall k \in K \tag{4}$$

$$\sum_{\substack{k_1 \in \tilde{K}: \\ (k_1,k_2) \in P}} X_{k_1k_2} = Z_{k_2} \quad \forall k_2 \in K \tag{5}$$

$$T_{k_2} \geq T_{k_1} + p_{a_{k_1}} + \gamma_{k_2} - T_{max}(1 - X_{k_1k_2}) \quad \forall (k_1, k_2) \in P : k_2 \neq k_0 \tag{6}$$

$$T_{k_0} = p^{start} \tag{7}$$

$$T_k \leq T_{max} - p_{a_k} + T_{max}(1 - Z_k) \quad \forall k \in K \tag{8}$$

$$T_{k_2} \geq T_{k_1} - 2T_{max}(1 - Y_{k_1k_2}) + p_s - 2T_{max}(2 - (Z_{k_1} + Z_{k_2})) \\ -p_s\xi_{k_1k_2} \quad \forall s \in F, \forall H \in \mathcal{H}_s, \forall k_1, k_2 \in H \tag{9}$$

$$T_{k_1} \geq T_{k_2} - 2T_{max}Y_{k_1k_2} + p_s - 2T_{max}(2 - (Z_{k_1} + Z_{k_2})) \\ -p_s\xi_{k_1k_2} \quad \forall s \in F, \forall H \in \mathcal{H}_s, \forall k_1, k_2 \in H \tag{10}$$

$$\sum_{k_1,k_2 \in H: k_1 \neq k_2} \xi_{k_1k_2} \leq \binom{\eta_s + 1}{2} - 1 \ \ \forall s \in F, \forall H \in \mathcal{H}_s \tag{11}$$

$$Z_k \in \{0, 1\} \quad \forall k \in K \tag{12}$$

$$T_k \geq 0 \quad \forall k \in K \tag{13}$$

$$X_{k_1k_2} \in \{0, 1\} \quad \forall (k_1, k_2) \in P \tag{14}$$

$$Y_{k_1k_2} \in \{0, 1\} \quad \forall k_1, k_2 \in K : a_{k_1} = a_{k_2} \neq 0, I_{k_1} \cap I_{k_2} = \emptyset \tag{15}$$

$$\xi_{k_1k_2} \in \{0, 1\} \quad \forall k_1, k_2 \in K : a_{k_1} = a_{k_2} \neq 0, I_{k_1} \cap I_{k_2} = \emptyset \tag{16}$$

The objective function (1) to be minimized represents the total travel distance. Constraints (2) ensure that each customer is visited exactly once. Constraints (3) guarantee that the number of routes selected does not exceed the number of available AFVs. Constraints (4) force each route to be a sequence of paths where the first one and the last one is k_0. Indeed, they guarantee that each path $k \neq k_0$ selected is traveled just after a path k_1 and just before a path k_2, both selected in the same route. Since constraints (4) are imposed for all paths $k \in K$, the dummy path is the only one not requiring to have both a predecessor and a successor. Constraints (5) guarantee that a path can be inserted in a route only if it is selected. If $X_{k_1k_2} = 1$, the refueling of path k_2 cannot start before the refueling of path k_1 at a_{k_1} has been completed (6). Constraints (7) fix $T_{k_0} = p^{start}$ to take into account the time spent for the initial refuel at the depot. Constraints (8) ensure that the refueling of each selected path k must be completed within T_{max}.

Constraints (9) and (10) guarantee that if both paths k_1 and k_2 are chosen and variable $\xi_{k_1k_2} = 1$, where $a_{k_1} = a_{k_2} = s$, then their starting fueling time at s has to be distanced of p_s. Indeed, if $Y_{k_1k_2} = 1$ then constraint (9) is active imposing that the starting fueling time of k_2 at s is postponed at least of p_s from that of k_1. Vice versa, if $Y_{k_1k_2} = 0$ constraint (10) guarantees that the starting fueling time of k_1 at s is postponed at least of p_s from that of k_2. Constraints (11) ensure that the capacity of each AFS s is respected. Indeed, for each AFS s and for each set of $\eta_s + 1$ paths of K ending in s, they impose that at most $\binom{\eta_s+1}{2} - 1$ pairs of such paths (i.e., not all of them) can overlap at s (since they can satisfy the non-overlapping refueling constraints (9) and (10) with $\xi_{k_1k_2} = 1$, thus may violate in practice the non-overlapping condition). Finally, constraints (12)–(16) specify the variables nature.

We notice that formulation (2)–(16) is intractable to be solved by a commercial MILP solver such as Gurobi or CPLEX since constraints (9), (10) and (11) are in exponential number. Indeed, e.g., (11) must be imposed for each H in H_s and therefore it consists in an exponential number of constraints being H_s the family of all sets of $\eta_s + 1$ customer disjoint paths belonging to K with the same arrival AFS s.

5 Cutting planes approach for the scenario with private AFSs

CPs methods are exact algorithms to solve NP-hard optimization problems [21]. They are based on the following principle. Given a MIP formulation of a problem (denoted as Original Problem - OP), a Relaxation Problem (RP) of it is obtained by discarding a set of constraints responsible of its hardness (e.g., the integrality constraints) or containing a huge number of constraints. Once the RP is solved to optimality, the feasibility of its solution is checked with respect to OP. If the optimal solution of RP is feasible for OP, then it is optimal for OP too. Otherwise, we detect the violated constraints, i.e. the *cuts*, to add to RP and we solve it again. This procedure is iteratively repeated until an optimal solution for OP (if any) is found. If some of the initially dropped constraints are added to RP making it infeasible, OP is also infeasible. This method is exact since it converges to an optimal solution, if it exists, in a finite number of iterations. Indeed, in the worst case, all the relaxed constraints are added to RP for proving that either OP is infeasible or finding an optimal solution.

In the scenario with private AFSs, an RP is obtained starting from the formulation reported in the previous section, by dropping constraints (9)–(11). The obtained problem corresponds to the classical G-VRP in which no limits on simultaneous refueling operations are imposed. The RP is then solved to optimality by a MIP commercial solver. The feasibility of the obtained solution is then checked with regard to the G-VRP-CAFS formulation (1)–(16), i.e., for every AFS s it is checked that the number of vehicles simultaneously refueling at s is not greater than η_s. If the solution is feasible, then it is optimal for G-VRP-CAFS too. Otherwise, all the violated constraints are added to RP, which is solved again. This procedure is iteratively repeated until either the solution of RP becomes feasible for G-VRP-CAFS or a Time Limit (TL) is reached. In the latter case, the RP optimal solution obtained at the last iteration represents the best lower bound. In fact, the RP_i, at a generic iteration i, is obtained adding further cuts to the RP at the previous iteration, RP_{i-1}, and, therefore, the optimal solution of RP_i, $z^*(RP_i)$, must be not lower than $z^*(RP_{i-1})$.

6 Cutting planes approach for the scenario with public AFSs

In the scenario with public AFSs, as introduced in [6], we suppose companies can reserve in advance the use of the fueling pumps of each AFS, in order to avoid unpredictable waiting times at the AFSs because all pumps are busy. In order to address this scenario, multiple time windows associated with the fueling pumps, to take into account reservations already made by other users, are considered. For each fueling pump h, let μ_h be the number of time windows associated with it and let $[e_h^q, l_h^q]$ its q-th time window, i.e., the range within which a refuel can start.

The CP approach presented in the previous section can be modified to address this scenario considering that, due to the time windows associated with each pump, the capacity of each AFS can vary over time. To this purpose, for each AFS s, from the time windows associated with its fueling pumps, we can build the piecewise constant function $\gamma_s(t)$ representing the maximum number of vehicles that can simultaneously fuel at s at time t. At each iteration of the CP, $\forall s \in F$, let $\varphi_s(t)$ a piecewise constant function representing the number of vehicles that fuel in AFS s at time t in the current relaxation of the problem. Consider now the piecewise constant function $\delta_s(t) = \varphi_s(t) - \gamma_s(t)$ and let n_s^δ the number of time intervals, each one denoted by $\Gamma_{si} = [e_i, l_i]$, within which $\delta_s(t)$ does not vary. Moreover, let γ_{si} the constant value assumed by $\gamma_s(t)$ in the interval Γ_{si}.

We add to the formulation also the decision variables $\tilde{Z}_{ik} = 1$ if path k is selected and the fuel happens in time interval Γ_{si}, 0 otherwise, $\forall s \in F, \forall i = 1, \ldots, n_s^\delta, \forall k \in K : a(k) = s$ and the auxiliary binary variables Z'_{ik} and Z''_{ik} defined on the same index set.

Moreover, the following cuts are added to the current RP instead of constraints (9)–(11), where $\tilde{T}$ denotes the time horizon width and ε denotes a suitable small constant (e.g., we fixed $\varepsilon = 10^{-5}$, in the experiments):

$$\begin{aligned} T_{k_2} \geq T_{k_1} - 2T_{max}(1 - Y_{k_1k_2}) + p_s - 2T_{max}(2 - (\tilde{Z}_{ik_1} + \tilde{Z}_{ik_2})) \\ -p_s\xi_{k_1k_2} \forall s \in F, \forall i = 1, \ldots, n_s^\delta, \forall k_1, k_2 \in K : \\ a(k_1) = a(k_2) = s \text{ and } \delta_s(t) > 0 \; \forall t \in \Gamma_{si} \end{aligned} \tag{17}$$

$$\begin{aligned} T_{k_1} \geq T_{k_2} - 2T_{max}Y_{k_1k_2} + p_s - 2T_{max}(2 - (\tilde{Z}_{ik_1} + \tilde{Z}_{ik_2})) \\ -p_s\xi_{k_1k_2} \quad \forall s \in F, \forall i = 1, \ldots, n_s^\delta, \quad \forall k_1, k_2 \in K : \\ a(k_1) = a(k_2) = s \text{ and } \delta_s(t) > 0 \quad \forall t \in \Gamma_{si} \end{aligned} \tag{18}$$

$$\begin{aligned} \sum_{\substack{k_1,k_2 \in K: a(k_1)=a(k_2)=s, \\ \delta_s(t)>0 \forall t \in \Gamma_{si}, k_1 \neq k_2}} \xi_{k_1k_2} \leq \binom{\gamma_{si}+1}{2} - 1 \\ \forall s \in F, \forall i = 1, \ldots, n_s^\delta, \end{aligned} \tag{19}$$

$$\begin{aligned} \frac{e_i + \varepsilon - T_k}{\tilde{T}} \leq Z'_{ik} \quad \forall s \in F, \forall i = 1, \ldots, n_s^\delta, \\ \forall k \in K : a(k) = s \end{aligned} \tag{20}$$

$$\begin{aligned} \frac{T_k + p_s + \varepsilon - l_i}{\tilde{T}} \leq Z''_{ik} \quad \forall s \in F, \forall i = 1, \ldots, n_s^\delta, \\ \forall k \in K : a(k) = s \end{aligned} \tag{21}$$

$$\tilde{Z}_{ik} \geq Z'_{ik} + Z''_{ik} - 1 \quad \forall s \in F, \forall i = 1, \ldots, n_s^\delta, \quad \forall k \in K : a(k) = s \tag{22}$$

Indeed, constraints (17) and (18) guarantee that if both paths k_1 and k_2 are chosen and variable $\xi_{k_1k_2} = 1$, where $a_{k_1} = a_{k_2} = s$, then their starting fueling time at s has to be distanced of p_s. Constraints (19) ensure that the capacity of each AFS s is respected. Indeed, for each AFS s and for each time interval Γ_{si}, they impose that at most $\binom{\gamma_{si}+1}{2} - 1$ pairs of paths of K ending in s and that in Γ_{si} violate the capacity γ_{si} (i.e., not all of them) can overlap at s. Constraints (20) ensure that if the refuel of path k starts before or at e_i then Z'_{ik} must be equal to 1. In similar way, constraints (21) guarantee that if the refuel of path k ends after or at l_i then Z''_{ik} must be equal to

1. Finally, constraints (22) enforce variables $\tilde{Z}_{ik}$ to be equal to 1 if both variables Z'_{ik} and Z''_{ik} are so.

7 Computational results

In this section, we present the numerical results obtained on a set of challenging instances, named Central, introduced in [6], with 2 fueling pumps per station. This set is composed of 10 instances with 15 customers. The depot is located very far from the customers area and only one AFS is considered, located at the center of the customers area. The number of available vehicles is equal to 8. The parameters have been set as in the following: $T_{max} = 7$, $p^{start} = 0$, $p_s = 0.5\ \forall s \in F$, $p_i = 0.5\ \forall i \in I$, $Q = 32$ and $r = 0.2$. The travel time from the depot to the AFS is 2 hours. The travel times between customers or customer and the AFS or customer and depot are computed considering the Euclidean distance and a constant speed equal to 40. This way, the available time for refueling is small and the AFS capacity becomes a crucial issue.

Each one of these instances has been properly modified in order to have 3 and 4 fueling pumps per station. The time windows related to the availability of each pump at every AFS have been settled in such a way that the optimal solution of the uncapacitated problem is not feasible for the capacitated one, to make the instance challenging.

All the instances are available at https://github.com/OrnellaPisacane/G-VRP-with-Capacitated-Alternative-Fuel-Stations.

The NEW-CP has been run under Xpress 7.9 on a machine equipped with a processor Intel i7-5500U at 2.4 GHz with 16 GB of RAM, with a CPU time limit of 3,600 seconds. The obtained results have been compared with those of the two CPs approaches presented in [6], named CP and CP-proactive, respectively.

In Table 1, the first column refers to the instance name, the next ones report the total Travel Distance (*TD*) of the optimal solution, the CPU time (*CPU*) required to certify it, the number of cuts (*Cuts*) generated and finally, the number of iterations (*Iter*) needed, respectively, for the scenario with 2, 3 and 4 fueling pumps.

Computational results show the efficiency and the effectiveness of the proposed method which is able to solve to optimality all the instances, with 2,3 and 4 pumps per station, in a very small number of iterations and in an average computational time of 16.35 seconds. Both CP and CP-proactive proposed in [6] are able to close to optimality, within the CPU time limit of 3, 600 seconds, only 8 over 10 instances with 2 pumps, in an average computational time of 1476.04 and 1450.83 seconds, respectively. Figure 1 depicts, instance by instance, a CPU time comparison among the three CP approaches on the CENTRAL instances when 2 fueling pumps per station are considered.

Since both the CP and the CP-proactive require to generate as many clones of each path as the number of pumps in the starting station multiplied by the number of pumps in the arrival station, they cannot address instances with 3 and 4 pumps. On the contrary, NEW-CP allows to directly address instances with more than 2 pumps without cloning the paths. This way, NEW-CP is able to solve to optimality also

Table 1 Numerical comparisons varying the number of pumps per AFS in CENTRAL instances

	2 Pumps				3 Pumps				4 Pumps			
Instance	TD	CPU	Cuts	Iter	TD	CPU	Cuts	Iter	TD	CPU	Cuts	Iter
Central 1	953.94	1.50	0	1	953.94	7.89	6	4	953.94	9.52	16	5
Central 2	959.29	10.33	2	3	959.29	12.61	12	4	959.29	9.24	8	3
Central 3	951.20	25.45	9	7	950.73	34.47	36	9	950.73	5.48	4	2
Central 4	967.96	43.74	7	6	1092.68	7.19	8	3	1092.68	15.09	4	2
Central 5	716.09	19.39	4	3	716.09	24.17	6	4	714.55	13.43	2	2
Central 6	844.43	15.70	1	2	844.43	15.68	0	1	844.43	15.55	0	1
Central 7	862.68	20.01	2	2	862.68	50.57	2	2	862.68	50.43	2	2
Central 8	712.83	4.20	0	1	712.83	8.77	2	2	716.9	18.71	10	4
Central 9	855.43	7.80	0	1	855.43	13.36	2	2	855.43	13.24	2	2
Central 10	905.59	15.35	4	3	914.81	15.39	6	3	905.59	9.13	4	2
AVG	872.94	16.35	2.9	2.9	886.29	19.01	8.0	3.4	885.62	15.98	5.2	2.5

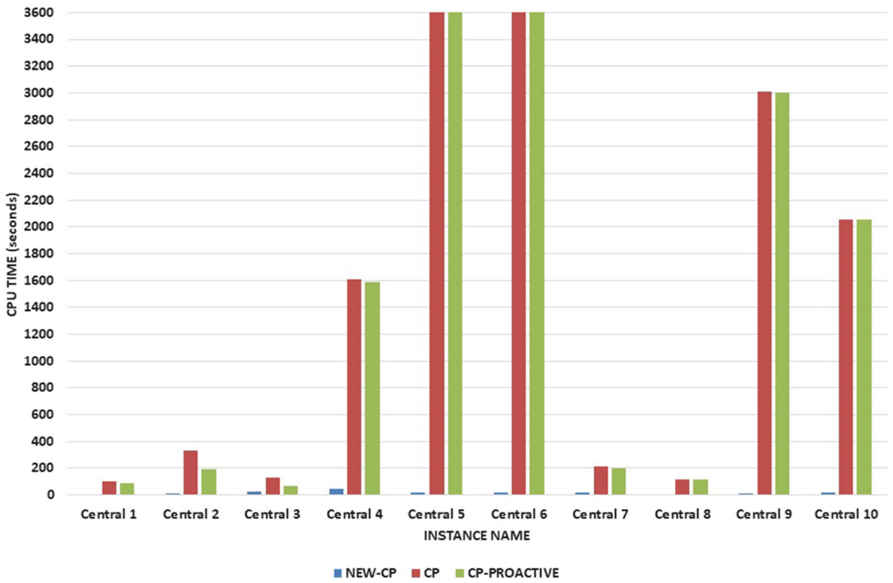

Fig. 1 CPU times comparisons among CP, CP-proactive and NEW-CP on the CENTRAL instances with 2 fueling pumps per AFSs

instances with 3 and 4 pumps per station, in very reasonable computational times (19.01 and 15.68 seconds, on average, respectively).

The main strength of this method is that the number of pumps does not impact on the computational time. This is a big advantage respect to the algorithms from the literature which, instead, strongly suffer this issue.

7.1 Sensitivity analysis on number of customers and their distribution

Table 2 shows the performances of NEW-CP by varying the number of customers (i.e., 15, 20 and 30) in the Central instances, being the number of pumps fixed to 5. The number of vehicles available is equal to 8 in the instances with both 15 and 20 customers whereas it has been fixed equal 12 in the instances with 30 customers to ensure feasibility.

First, it is worth remarking that the results with 15 customers are equal to those obtained considering 4 pumps, as shown in Table 1. This is due to the fact that above a certain threshold, the number of pumps does not affect the final results. Regarding the CPU time, it is more than twice and more than five times passing from 15 to 20 customers and from 20 to 30 customers, respectively. Concerning the number of cuts added, about 4 extra cuts and only 1 extra cut are added respectively passing from 15 to 20 customers and from 20 to 30 customers. We can notice that, with the increasing of the number of customers, the computational times increase but not in an exponential way (see also Fig. 2). Therefore, instances in which the number of customers is up to 30 are tractable within very reasonable computational times (around 3 minutes).

Table 2 Numerical comparisons varying the number of customers in CENTRAL instances with 5 pumps

	15 Customers				20 Customers				30 Customers			
Instance	TD	CPU	Cuts	Iter	TD	CPU	Cuts	Iter	TD	CPU	Cuts	Iter
Central 1	953.94	9.52	16	5	1531.34	33.47	4	2	2145.76	12.72	8	3
Central 2	959.29	9.24	8	3	1531.92	63.94	2	2	1880.48	47.81	8	3
Central 3	950.73	5.48	4	2	1366.44	6.03	12	4	2083.62	737.98	32	8
Central 4	1092.68	15.09	4	2	1306.65	5.057	12	3	1914.37	59.748	20	5
Central 5	714.55	13.43	2	2	1311.35	21.22	4	2	1882.43	33.69	4	2
Central 6	844.43	15.55	0	1	1286.09	45.90	20	3	1704.04	126.23	8	2
Central 7	862.68	50.43	2	2	1321.29	11.23	8	2	1706.19	295.8	14	2
Central 8	716.9	18.71	10	4	1134.37	6.63	4	2	1703.35	41.39	4	2
Central 9	855.43	13.24	2	2	1332.63	7.465	20	4	1678.90	245.80	4	2
Central 10	905.59	9.13	4	2	1313.57	132.64	10	3	1878.19	165.45	4	2
AVG	885.62	15.98	5.20	2.50	1343.57	33.36	9.60	2.70	1857.73	176.66	10.60	3.10

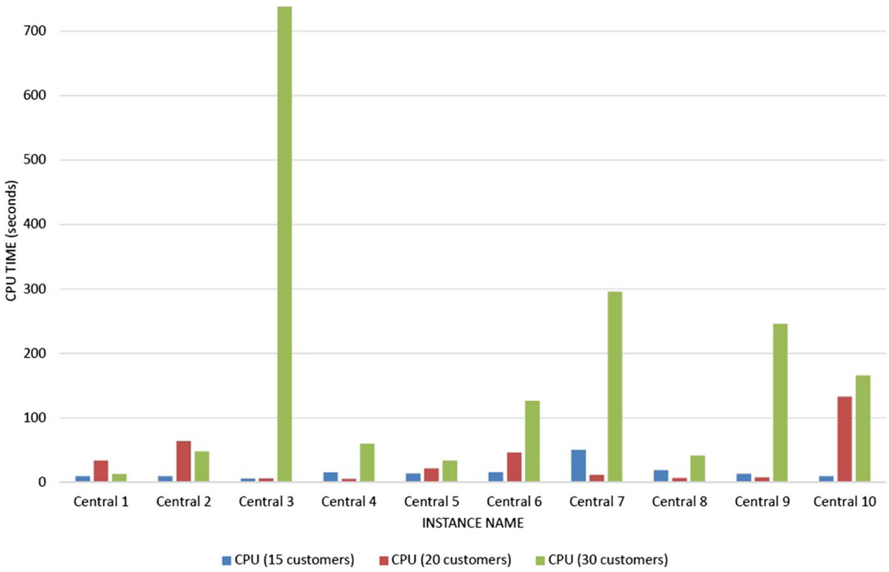

Fig. 2 CPU times comparison varying the number of customers in the Central instances

Table 3 Numerical comparisons varying customers distribution in CENTRAL instances with 4 pumps

	15 Customers Random				15 Customers Clustered			
Instance	TD	CPU	Cuts	Iter	TD	CPU	Cuts	Iter
Central 1	953.94	9.52	16	5	1375.80	2.21	20	6
Central 2	959.29	9.24	8	3	1084.54	0.78	8	3
Central 3	950.73	5.48	4	2	1177.56	0.44	8	2
Central 4	1092.68	15.09	4	2	1366.45	4.09	24	7
Central 5	714.55	13.43	2	2	1384.94	0.22	8	2
Central 6	844.43	15.55	0	1	1092.51	0.57	12	3
Central 7	862.68	50.43	2	2	887.48	0.86	10	3
Central 8	716.9	18.71	10	4	1083.19	1.19	14	5
Central 9	855.43	13.24	2	2	1088.74	0.61	16	3
Central 10	905.59	9.13	4	2	1100.58	0.32	8	2
AVG	885.62	15.98	5.20	2.50	1164.18	1.13	12.80	3.60

Table 3 compares performances of NEW-CP on instances with random and clustered distribution. In these instances, the number of customers and of pumps have been fixed to 15 and 4, respectively. It is worth noting that the average computation times decrease of about 93% when the customers are clustered (see also Fig. 3). Indeed, in this case, the number of feasible non-dominated paths is remarkably lower since there are very few paths serving customers belonging to different clusters being these far from each other. However, the average number of cuts added increases of about 7.6 when the customers are clustered, being more challenging the problem of scheduling refuels

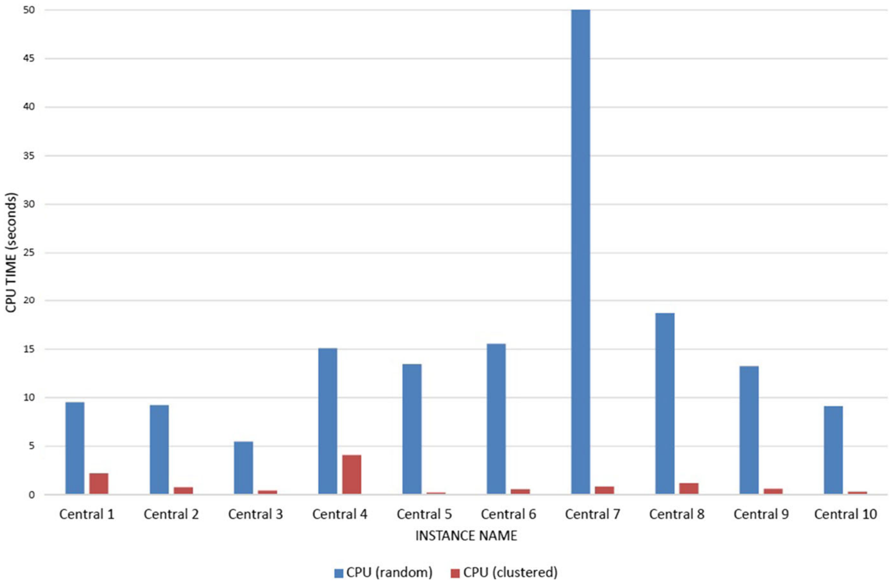

Fig. 3 CPU times comparison varying the customers distribution in the Central instances

at the station. In fact, since customers are clustered, the refuels at the station are performed either before or after serving them otherwise the detour cost may strongly increase. This causes more overlaps at the station and then, the need of adding more cuts during the approach execution.

8 Conclusions and future works

The Green Vehicle Routing Problem with Capacitated Alternative Fuel Stations (G-VRP-CAFS) is a more realistic variant of the Green Vehicle Routing Problem (G-VRP) where the AFSs capacity is taken into account. It was introduced in [6] where the authors, together with an Arc-based MILP formulation, proposed also a path-based MILP model. However, for efficiently solving this latter model, also on instances in which the number of feasible non-dominated paths was high, two slightly different Cutting Planes (CPs) approaches were proposed, named CP and CP-proactive, respectively. All the proposed approaches were firstly designed for considering the scenario with private AFSs and then, extended for that with public AFSs where their reservation was allowed. In the latter scenario, multiple time windows associated with the fueling pumps of the AFSs were introduced to consider the reservations already made by other users.

However, the main drawback of the proposed approaches was that in order to take into account the AFSs' capacity, as many clones of each path as the number of fueling pumps of the starting station multiplied by that of the arrival station must be introduced. Cloning paths may make the path-based approaches impracticable on instances in which the number of fueling pumps per station is higher than 2.

In order to overcome this issue, in this paper, we firstly proposed new path-based MILP models without cloning paths, for both the scenario with private AFSs and that with public AFSs. Then, we designed a more efficient CP approach (NEW-CP) suitable to address both the scenarios. Numerical results showed that NEW-CP was suitable to close to optimality all the instances with 2 fueling pumps per station in an average CPU time that is 98.89% and 98.87% less than that of CP and CP-proactive, respectively. It is worth remarking that both CP and CP-proactive were not able to close to optimality 2 of the 10 instances.

Both the efficiency and the effectiveness of NEW-CP were shown on a new set of challenging instances, generated in this work, where 3 and 4 fueling pumps per station were considered. The NEW-CP approach was suitable to solve to optimality all these instances, in an average CPU time of 19.01 and 15.98 seconds, respectively. On the contrary, CP and CP-proactive were not suitable to address those instances because, due to the very large number of paths involved, the solver reaches the *out of memory* status.

Our solution approach could be extended to all scheduling problems, in which a maximum number of tasks can be simultaneously executed by the same machine. This approach could be also useful to describe air flight scheduling, in which, for operational reasons, a maximum number of flights can simultaneously overfly a given zone. In the vehicle routing field, a possible application could address VRPs with cross-docking operations at satellites, where a maximum number of vehicles can simultaneously carried out loading/unloading operations, due to the limited available space.

Acknowledgements This work was partially founded by the Dipartimento di Ingegneria dell'Informazione of the Universitá Politecnica delle Marche, under the project entitled "A network-based approach to uniformly extract knowledge and support decision making in heterogeneous application contexts" (RSA-B 2018).

Funding Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement.

References

1. Andelmin, J., Bartolini, E.: An exact algorithm for the green vehicle routing problem. Transp. Sci. **51**(4), 1288–1303 (2017)
2. Andelmin, J., Bartolini, E.: A multi-start local search heuristic for the green vehicle routing problem based on a multigraph reformulation. Comput. Oper. Res. **109**, 43–63 (2019)
3. Bard, J.F., Huang, L., Dror, M., Jaillet, P.: A branch and cut algorithm for the VRP with satellite facilities. IIE Trans. **30**(9), 821–834 (1998)
4. Bruglieri, M., Pezzella, F., Pisacane, O., Suraci, S.: A variable neighborhood search branching for the electric vehicle routing problem with time windows. Electron. Notes Discrete Math. **47**, 221–228 (2015)

5. Bruglieri, M., Mancini, S., Pezzella, F., Pisacane, O.: A path-based solution approach for the green vehicle routing problem. Comput. Oper. Res. **103**, 109–122 (2019a)
6. Bruglieri, M., Mancini, S., Pisacane, O.: The green vehicle routing problem with capacitated alternative fuel stations. Comput. Oper. Res. **112**, 104759 (2019b)
7. Bruglieri, M., Mancini, S., Pisacane, O.: More efficient formulations and valid inequalities for the green vehicle routing problem. Transp. Res. Part C Emerg. Technol. **105**, 283–296 (2019c)
8. Desaulniers, G., Errico, F., Irnich, S., Schneider, M.: Exact algorithms for electric vehicle-routing problems with time windows. Oper. Res. **64**(6), 1388–1405 (2016)
9. Erdoğan, S., Miller-Hooks, E.: A green vehicle routing problem. Transp. Res. Part E Logist. Transp. Rev. **48**(1), 100–114 (2012)
10. Felipe, Á., Ortuño, M.T., Righini, G., Tirado, G.: A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. Transp. Res. Part E Logist. Transp. Rev. **71**, 111–128 (2014)
11. Grandinetti, L., Guerriero, F., Pezzella, F., Pisacane, O.: A pick-up and delivery problem with time windows by electric vehicles. Int. J. Product. Qual. Manag. **18**(2–3), 403–423 (2016)
12. Hiermann, G., Puchinger, J., Ropke, S., Hartl, R.F.: The electric fleet size and mix vehicle routing problem with time windows and recharging stations. Eur. J. Oper. Res. **252**(3), 995–1018 (2016)
13. Hiermann, G., Hartl, R.F., Puchinger, J., Vidal, T.: Routing a mix of conventional, plug-in hybrid, and electric vehicles. Eur. J. Oper. Res. **272**(1), 235–248 (2018)
14. IEA: Global ev outlook 2019. IEA, Paris (2019)
15. Jemai, J., Zekri, M., Mellouli, K.: An NSGA-II algorithm for the green vehicle routing problem. In: European Conference on Evolutionary Computation in Combinatorial Optimization, pp 37–48. Springer (2012)
16. Keskin, M., Laporte, G., Çatay, B.: Electric vehicle routing problem with time-dependent waiting times at recharging stations. Comput. Oper. Res. **107**, 77–94 (2019)
17. Koç, Ç., Karaoglan, I.: The green vehicle routing problem: a heuristic based exact solution approach. Appl. Soft Comput. **39**, 154–164 (2016)
18. Leggieri, V., Haouari, M.: A practical solution approach for the green vehicle routing problem. Transp. Res. Part E Logist. Transp. Rev. **104**, 97–112 (2017)
19. Majidi, S., Hosseini-Motlagh, S.M., Yaghoubi, S., Jokar, A.: Fuzzy green vehicle routing problem with simultaneous pickup-delivery and time windows. RAIRO Oper. Res. **51**(4), 1151–1176 (2017)
20. Mancini, S.: The hybrid vehicle routing problem. Transp. Res. Part C Emerg. Technol. **78**, 1–12 (2017)
21. Marchand, H., Martin, A., Weismantel, R., Wolsey, L.: Cutting planes in integer and mixed integer programming. Discrete Appl. Math. **123**, 397–446 (2002)
22. Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G.: A multi-space sampling heuristic for the green vehicle routing problem. Transp. Res. Part C Emerg. Technol. **70**, 113–128 (2016)
23. Sawik, B., Faulin, J., Pérez-Bernabeu, E.: Multi-criteria optimization for fleet size with environmental aspects. Transp. Res. Proc. **27**, 61–68 (2017a)
24. Sawik, B., Faulin, J., Pérez-Bernabeu, E.: A multicriteria analysis for the green VRP: a case discussion for the distribution problem of a Spanish retailer. Transp. Res. Proc. **22**, 305–313 (2017b)
25. Schneider, M., Stenger, A., Goeke, D.: The electric vehicle-routing problem with time windows and recharging stations. Transp. Sci. **48**(4), 500–520 (2014)
26. Schneider, M., Stenger, A., Hof, J.: An adaptive VNS algorithm for vehicle routing problems with intermediate stops. OR Spectr. **37**(2), 353–387 (2015)