Computers and Operations Research

A GRASP with penalty objective function for the Green Vehicle Routing Problem with Private Capacitated Stations

--Manuscript Draft--

Manuscript Number:	
Article Type:	Research Article
Keywords:	Routing; Metaheuristics; Greedy Randomized Adaptive Search Procedure; Alternative Fuel Stations; Route Incompatibility
Corresponding Author:	Maurizio Bruglieri, Prof. Politecnico di Milano Milano, ITALY
First Author:	Maurizio Bruglieri, Prof.
Order of Authors:	Maurizio Bruglieri, Prof.
	Daniele Ferone, PhD
	Paola Festa, Prof.
	Ornella Pisacane, Ph.D.
Abstract:	Due to the recent worries about the environment, the transport companies are incentivazing to use Alternative Fuel Vehicles (AFVs) instead of the conventional ones. However, due to the limited AFV driving range and since the Alternative Fuel Stations (AFSs) are usually not widespread on the territory, the routes of the AFVs have to be properly planned in order to prevent them from remaining without the sufficient fuel to reach the depot or the closest station. The Green Vehicle Routing Problem (G-VRP) aims at determining the AFVs routes, each one serving customers within a maximum duration, minimizing the total travel distance and if necessary including stops at AFSs. On the contrary of the G-VRP, the G-VRP with Capacitated AFSs (G-VRP-CAFS) more realistically assumes that each AFS has alimited number of fueling pumps and then prevents overlapping in refueling operations. In this paper, a Greedy Randomized Adaptive Search Procedure (GRASP), properly guided by some theoretical results, is designed to efficiently solve also large-sized instances of the G-VRP-CAFS. Computational results, carried out on both benchmark instances and large-sized instances, show the effectiveness and the efficiency of the proposed GRASP.

Cover Letter

Dear Editor,

please find enclosed the paper "*A GRASP with penalty objective function for the GreenVehicle Routing Problem with Private CapacitatedStations*", by M. Bruglieri, D. Ferone, P. Festa and O. Pisacane.

In this paper, we address the GVRP-CAFS, a recent variant of the Green Vehicle Routing Problem (G-VRP) under the more realistic assumption that the Alternative Fuel Stations (AFSs) have a limited fuel capacity.

The problem was already addressed in the literature through exact methods (a Path-based model solved by cutting plane approaches) only on instances with up to 20 customers.

In order to address also large-sized instances, in this work we develop a Greedy Randomized Adaptive Search Procedure (GRASP), properly guided by some theoretical results. Computational experiments, carried out on both literature benchmark instances and new large-sized instances with up 100 customers, show the effectiveness and the efficiency of the proposed solution approach.

Given the practical importance of the problem addressed and the good results obtained, we think the paper can be considered for publication in *Computers and Operations Research*.

Finally, we believe that our paper is also consistent with the aims and scopes of the journal since some works on similar topic have been already published by COR, as proven by our literature review, e.g., Andelmin & Bartolini (2019), Keskin et al. (2018, 2019, 2020).

Thank you.

Best regards,

Maurizio Bruglieri

(on behalf of the co-authors)

A GRASP with penalty objective function for the Green Vehicle Routing Problem with Private Capacitated Stations

M. Bruglieri[a,*], D. Ferone[b], P. Festa[b], O. Pisacane[c]

[a] *Dipartimento di Design, Politecnico di Milano, Italy*
[b] *Dipartimento di Matematica e Applicazioni "R. Caccioppoli", Universitá degli Studi di Napoli Federico II, Italy*
[c] *Dipartimento di Ingegneria dell'Informazione, Universitá Politecnica delle Marche, Italy*

Abstract

Due to the recent worries about the environment, the transport companies are incentivazing to use *Alternative Fuel Vehicles* (AFVs) instead of the conventional ones. However, due to the limited AFV driving range and since the *Alternative Fuel Stations* (AFSs) are usually not widespread on the territory, the routes of the AFVs have to be properly planned in order to prevent them from remaining without the sufficient fuel to reach the depot or the closest station. The *Green Vehicle Routing Problem* (G-VRP) aims at determining the AFVs routes, each one serving customers within a maximum duration, minimizing the total travel distance and if necessary including stops at AFSs. On the contrary of the G-VRP, the *G-VRP with Capacitated AFSs* (G-VRP-CAFS) more realistically assumes that each AFS has a limited number of fueling pumps and then prevents overlapping in refueling operations. In this paper, a *Greedy Randomized Adaptive Search Procedure* (GRASP), properly guided by some theoretical results, is designed to efficiently solve also large-sized instances of the G-VRP-CAFS. Computational results, carried out on both benchmark instances and large-sized instances, show the effectiveness and the efficiency of the proposed GRASP.

Keywords: Routing, Metaheuristics, Greedy Randomized Adaptive Search Procedure, Alternative Fuel Stations, Route Incompatibility

1. Introduction

According to a recent report published by the European Environment Agency, about 25% of the total greenhouses gas emissions in Europe is due to the transport sector (Agency (2020)). Therefore, cutting emissions due to this sector plays a key role in order to reach one of the long-term goals

*Corresponding author: maurizio.bruglieri@polimi.it

of the European Union, i.e., having net-zero greenhouses gas emissions by 2050. Indeed, the European Union is currently incentivizing the Alternative Fuel Vehicles (AFVs), instead of the traditional Internal Combustion Engine Vehicles (ICEVs), that use alternative fuel like, for instance, biodisel, electricity, hydrogen. As a result, many companies operating in the Logistics sector, are currently considering either replacing or integrating their fleet with AFVs.

Compared to the ICEVs, the AFVs guarantee several advantages among which less global harmful emissions (*environmental sustainability*) and lower kilometer costs (*economic sustainability*). In addition, they can also reach the so-called Limited Traffic Zones, typically forbidden to the ICEVs. As a result, they guarantee both a widespread customers coverage especially in last-mile Logistics and a more efficient door-to-door transport (*social sustainability*). However, despite these benefits, the AFVs have usually a limited driving range and therefore, they may require to refuel more than once during a trip. In addition, the Alternative Fuel Stations (AFSs) are currently not widespread on the territory and this implicitly requires to properly plan the route of such vehicles in order to avoid that they may remain without the sufficient fuel level to reach the closest station.

In fact, it is not surprising if a growing number of operations research specialists are currently studying the problem of efficiently routing a fleet of AFVs, i.e., the *Green Vehicle Routing Problem* (G-VRP). The G-VRP was introduced by the seminal work of Erdoğan and Miller-Hooks (Erdoğan & Miller-Hooks (2012)). Compared to the traditional *Vehicle Routing Problems* (VRPs), this problem takes into account also the need of the fleet to refuel during the trips and then, it includes stops at the AFSs. The route of each AFV of the fleet starts from a common depot and returns to it within a maximum time duration. During its route, an AFV serves some customers, each one with a specific service time, possibly stopping at stations to refuel. The fuel consumption rate is assumed proportional to the travelled distance and therefore, the driving range of each AFV can be easily deduced from its tank capacity. Moreover, the refueling time at each station is assumed to be constant. Therefore, the G-VRP aims at routing a set of AFVs to serve customers geographically distributed with possible stops at AFSs, minimizing the total travel distance.

Starting from Erdoğan & Miller-Hooks (2012), an increasing number of scientific contributions have been presented in the literature, in which either the original G-VRP or its variants have been addressed. In almost all these contributions, each AFS is assumed to have an unlimited number of fueling pumps and therefore, each AFV always fuels at a station without waiting. However, this assumption does not hold in real-life applications where each AFS has instead a limited number of fueling pumps. In fact, Bruglieri, Mancini and Pisacane in 2019 introduced the G-VRP with Capacitated AFSs (G-VRP-CAFS), i.e., the G-VRP, in which the stations have a limited number of fueling pumps (Bruglieri et al. (2019b)). The same authors also showed as the AFS capacity becomes a bottleneck in some particular situations in which several AFVs may overlap at the same AFS during the refueling operations. Compared to Bruglieri et al. (2019b), in this paper, we focus attention on the private scenario in which the AFSs are only those owned by the company. The main contributions of this work are in the following:

- designing a metaheuristic solution approach suitable to efficiently address the G-VRP-CAFS allowing route duration infeasibility in the starting solution and eliminating it along the search through a penalty objective function;
- deriving some theoretical results on the compatibility among routes for properly and efficiently driving the proposed solution approach;
- generating a more realistic set of benchmark instances for the G-VRP-CAFS, i.e., instances with a significant number of customers to serve.

The rest of the paper is organized as follows. Section 2 presents an overview of the main contributions in the literature. Section 3 describes the problem addressed and the notation used. Section 4 provides some theoretical results. Section 5 describes the solution approach. Section 6.1 details the instances generation procedure and Sections 6.2 and 6.3 discuss the experimental results. Finally, Section 7 draws some conclusions and outlines some future research directions worthy of investigation.

2. Related Work

The G-VRP, belonging to the class of the VRPs Toth et al. (2015), is currently attracting the interest of many operations research specialists as shown in several recent surveys, e.g., Bektaş et al. (2016), Bektaş et al. (2019), Asghari et al. (2020), Moghdani et al. (2020). Introduced in Erdoğan & Miller-Hooks (2012), it is mathematically modelled through Mixed Integer Linear Programming (MILP) by cloning the stations in order to allow multiple visit to the same station. The need of cloning the stations makes this formulation not suitable to be used on real-life alike instances. For this reason, a Clarke & Wright Savings algorithm, properly modified for the G-VRP, is also proposed.

A first attempt to get rid of AFS clones is described in Koç & Karaoglan (2016) where a three-index MILP model is proposed, through variables indicating if an AFV stops at an AFS traveling from a node to another. The authors propose both a Branch&Cut exact algorithm, combining several valid inequalities and a Simulated Annealing (SA) based approach to obtain upper bounds. Also in the MILP cloneless formulation proposed in Bruglieri et al. (2016), the stops at the AFSs are only implicitly considered. But, compared to Koç & Karaoglan (2016), the number of variables is significantly reduced by pre-computing a set of efficient AFSs for each pair of customers. The problem of avoiding the AFS clones is addressed also in Bruglieri et al. (2019c) where two MILP models are proposed. In the first one, between two customers or between a customer and the depot, only one stop at an AFS is allowed. On the contrary, in the second model, two consecutive stops at the AFSs are also allowed. Both valid inequalities and dominance criteria to a priori identify the AFSs that are more efficient to use in each route are proposed. Finally, in Leggieri & Haouari (2017), a MILP formulation is proposed together with a reduction procedure, shown to be competitive on medium-sized instances.

An exact solution approach, based on the definition of a multigraph, is proposed in Andelmin & Bartolini (2017). In particular, each node is a customer whereas an arc between two nodes is a nondominated path that may include also stops at the AFSs. The set of routes found on the multigraph is the input of a set partitioning model strengthen through valid inequalities. The authors optimally solve instances with up to 100 customers. A path-based exact solution approach is proposed in Bruglieri et al. (2019a) where the authors exploit both the limited driving range of the AFVs and the maximum route duration. In particular, a path represents an ordered sequence of customers, visited from a starting node to an ending node without intermediate stops at any AFS. The number of all exhaustively generated paths is properly reduced by both feasibility and dominance conditions. The set of all feasible non-dominated paths is finally given in input to a path-based model for building the optimal solution. The proposed approach outperforms all exact solution methods already existing on both small and medium-sized instances. On large-sized instances, the set of feasible non-dominated paths is instead heuristically generated.

Regarding meta-heuristics designed for the G-VRP, the one proposed in Montoya et al. (2016) consists in two stages. In the first stage, a pool of routes is generated through some randomized route-first cluster-second heuristics and a properly designed AFSs insertion procedure. In the second stage, this pool becomes input of a set partitioning formulation. A Genetic Algorithm (GA) is instead proposed in da Costa et al. (2018), tested on real-world instances including road speed and gradient data. Whereas, in Affi et al. (2018), a Variable Neighborhood Search (VNS) is designed and shown to be competitive on the large-sized benchmark instances. The multi-start local search algorithm described in Andelmin & Bartolini (2019) uses the multigraph reformulation proposed in Andelmin & Bartolini (2017). The solutions iteratively built are improved by a local search and all these routes are input of a set partitioning model. The final solution is further improved through a local search.

Variants of the problem include, for instance, the G-VRP with Pickups and Deliveries (G-VRPPD). For example, in Madankumar & Rajendran (2018), the problem is formulated for minimizing costs, considering pickup and delivery operations, product and vehicle compatibility, vehicle capacity, request-priorities and request-types, and start/completion time constraints. A second model is also formulated considering different fuel prices at different AFSs, minimizing both routing and refueling costs. A G-VRPPD is also addressed in Soysal et al. (2018). In particular, compared to the models traditionally proposed for the one-to-one pickup and delivery problem, new aspects, e.g., fuel consumption, variable vehicle speed and road categorization (i.e., urban, non-urban) are introduced. This problem is modelled through MILP and solved on a case study from the Netherlands. A bi-objective Fuel G-VRP with varying speed constraint is discussed in Poonthalir & Nadarajan (2018). The problem, modelled through goal programming for minimizing both route cost and fuel consumption, is solved by Particle Swarm Optimization with greedy mutation operator and time varying acceleration coefficient. In Zhang et al. (2018), the G-VRP with vehicle capacity constraint is addressed and both a two-phase heuristic and an Ant Colony based meta-heuristic are designed. In Xu et al. (2019), together with the vehicle capacity constraint, the G-VRP with time-varying vehicle speed and soft time windows is modelled through multi-objective Mixed Integer NonLinear Programming, solved through a non-dominated sorting GA with adaptive and greedy strategies. In Hooshmand & MirHassani (2019), it is assumed that the travel time and the fuel consumption on each arc depend on the distance traveled and the time of the day at which that arc is traversed. The problem is then modelled through MILP, solved by a hybrid heuristic on large-sized instances. In Yu et al. (2019), a Branch&Price algorithm is designed to solve the G-VRP with heterogeneous fleet and time windows. For such a variant, a labeling based multi-vehicle approximate dynamic programming algorithm is developed. A G-VRP with Time Windows is studied in Yu et al. (2020) for which an Adaptive Large Neighborhood Search (ALNS) is proposed.

Some papers assume that the green fleet specifically consists of Electric Vehicles (EVs). The Electric VRP with Time Windows (E-VRPTW), introduced in Schneider et al. (2014), is more difficult than the G-VRP since the recharging time depends on the actual battery level. The E-VRPTW is formulated by MILP and a hybrid heuristic combining a VNS with a Tabu Search is also proposed for solving large-sized instances. After that, an increasing number of works address the E-VRPTW and its variants. For example, in Felipe et al. (2014), the amount of energy recharged and the technology used are both taken into account together with the route plan. Constructive and local search heuristics are then designed and embedded into a non deterministic SA. The E-VRPTW with partial recharges is introduced in Bruglieri et al. (2015), assuming that an EV is not always fully recharged at a station. This way, the battery level recharged each time becomes a decision variable. For this variant, a MILP formulation is proposed, minimizing the total travel time, the total waiting time and the total recharging time plus the number of the EVs used. Then, a VNS Branching (VNSB) matheuristic is designed in order to solve also large-sized instances in reasonable computational times. Whereas, considering partial recharges, an ALNS is designed and described in Keskin & Çatay (2016). Four variants of the E-VRPTW are instead addressed in Desaulniers et al. (2016), i.e., with either at most a single or multiple recharges per route, with either full or partial recharges. For these variants, Branch&Price&Cut algorithms are proposed. Recently, in Ceselli et al. (2021), an EVRP with multiple recharge technologies is addressed and solved through a Branch&Cut&Price algorithm. In particular, in the master problem, each column is a sequence of customers visited between two stations. The proposed algorithm is suitable to solve instances with up to thirty customers, nine recharge stations, five vehicles and three technologies. In Bruglieri et al. (2017), a VNSB is used in a three-phase framework for addressing the E-VRPTW with partial recharges. In particular, the number of EVs used is firstly optimized and then, the total time spent by the EVs, i.e., travel, charging and waiting times, is minimized. In the first two phases, MILP-based Programs are used to generate feasible solutions, input of the VNSB. In Keskin & Çatay (2018), the E-VRPTW with partial recharges is addressed considering a normal, a fast and a super-fast recharging configuration. The total recharging cost while operating minimum number of vehicles is minimized. A MILP model is then proposed and solved on small-sized instances. Whereas, the large-sized instances are solved through an ALNS. EVs are instead used for both pickup and delivery operations in Goeke (2019). The author designs a metaheuristic to solve the problem and derives a policy determining the energy to be recharged in the case in which time windows are present. In Lu et al. (2020), the time-dependent EVRP is addressed in which together with the routes plan, decisions on vehicle's speed and departure time at each arc of the routes have to be also taken for minimizing a cost function. This problem is modelled through ILP and solved by an iterated VNS on more realistic instances. Finally, in Montoya et al. (2017), nonlinear charging functions are assumed in contrast with traditionally made assumption according to which the battery-charge level is a linear function of the charging time. For this variant, a hybrid metaheuristic is designed.

In Poonthalir & Nadarajan (2019), the G-VRP is addressed assuming that, in some cases, the AFVs have to wait before being refuelled. For this reason, each AFS is modelled as an M/M/1 queue. This new variant is then solved through Chemical Reaction Optimization. In Keskin et al. (2019), the E-VRPTW is addressed considering waiting times at stations and a time horizon split into intervals with varying queueing times. An ALNS combined with an exact method based on the solution of a MILP model is proposed. The same authors in Keskin et al. (2020) formulate the problem as a two-stage model. In the first stage, expectations of queueing times are used to detect routes by the ALNS. Whereas, in the second stage, these routes are corrected via simulation.

To the best of our knowledge, only in Bruglieri et al. (2019b), the G-VRP-CAFS is addressed without queuing at AFSs. In particular, two different scenarios are considered. In the private scenario, the AFSs are owned by the transport company and therefore, stops at them are properly managed without overlapping among the AFVs. In the public scenario, instead, the AFSs are not owned by the transport company and a fueling pumps reservation mechanism is considered in order to prevent AFVs' queuing at stations. As a consequence, time windows associated with AFSs availability are introduced. For both the scenarios, arc-based MILP formulations are proposed. Moreover, the path-based approach of Bruglieri et al. (2019a) is also extended for this new variant. In particular, for efficiently solving the path-based models, two cutting-plane exact approaches are also designed. The exact solution approaches are nevertheless suitable to solve only middle-sized instances.

In the present work, starting from the aforementioned assumptions holding in the private scenario, an efficient and effective metaheuristic method is described in order to address also large-sized instances of the G-VRP-CAFS.

3. Problem Statement and Notation

In the following, we recall the problem statement, the notation and the assumptions already introduced in Bruglieri et al. (2019b), as also summarized in Table 1.

The G-VRP-CAFS is defined on a direct complete graph $G = \langle N, A \rangle$ where $N = I \cup F \cup \{0\}$ denotes the set of nodes containing the set of customers I to serve, the set of private AFSs F and the common depot 0. Whereas, the set A contains all the arcs (i, j) such that $i, j \in N$ and $i \neq j$.

The number of AFVs available is an input data, indicated by m. Moreover, the maximum route duration as well as the maximum distance an AFV can travel after a full refueling are denoted by T_{max} and D_{max}, respectively. In addition, v, Q and r indicate the average vehicle speed, the vehicle fuel capacity and the fuel consumption rate, respectively. This way, the parameter D_{max} can be easily deduced as $D_{max} = \frac{Q}{r}$.

For each customer $i \in I$, a service time p_i is given. Similarly, for each AFS $s \in F$, a constant refueling time p_s is known. Since all the AFVs start from the depot fully refueled, a starting refueling time p^{start} is also introduced to take into account the time spent for the initial refueling. For each pair of nodes $i, j \in N$, a travel distance d_{ij} as well as a travel time t_{ij} are known. Finally, for each AFS s, the number of fueling pumps η_s is also given as input.

Table 1: Nomenclature of the G-VRP-CAFS

Notation	Meaning
F	Set of AFSs
I	Set of customers
N	Set of nodes
A	Set of arcs
m	Number of available AFVs
0	Depot
v	Average speed
Q	Maximum fuel capacity
r	Fuel consumption rate
T_{max}	Maximum route duration
D_{max}	Maximum driving range
p^{start}	Starting refueling time
p_s	Refueling time at AFS s
p_i	Service time at customer i
d_{ij}	Travel distance from node i to node j
t_{ij}	Travel time from node i to node j
η_s	Number of fueling pumps at AFS s

The G-VRP-CAFS consists in finding at most m routes, starting and ending in the depot, such that all the customers are served and the maximum route duration, the driving range after a refueling and the AFS capacity are never exceeded. The objective is the minimization of the total traveled distance.

4. Some Theoretical Results

A question that frequently must be addressed when the G-VRP-CAFS is solved with a metaheuristic approach is whether two routes, refueling at the same AFS, are compatible, i.e., they can be scheduled in such a way that do not overlap when refueling.

To this purpose, for each route we can compute the earliest visit time of each node and the maximum shifting that does not violate T_{max} (i.e., the difference between T_{max} and the minimum route duration). The question is solved by the following proposition, when the station capacity is 1.

Proposition 4.1. *Two routes ω_1 and ω_2, refueling at the same AFS s with capacity $\eta_s = 1$, are incompatible if and only if the following two conditions simultaneously hold:*

$$\tau_1 + p_s - \tau_2 > \sigma_2 \tag{1}$$

$$\tau_2 + p_s - \tau_1 > \sigma_1 \tag{2}$$

where τ_1 and τ_2, are the earliest visit time of station s in the routes ω_1 and ω_2, respectively, with $\tau_1 \leq \tau_2$, and σ_1, σ_2 their maximum shifting.

Proof. ($\Leftarrow$) Suppose that the two inequalities are satisfied. From $\tau_1 + p_s - \tau_2 > \sigma_2$ we deduce $\tau_1 + p_s > \tau_2$ and therefore the earliest refueling starting time of ω_2 overlaps with the earliest refueling ending time of ω_1. Moreover, it is not possible to postpone the refueling starting time of ω_2 when the refueling of ω_1 is finished since in this case ω_2 should be postponed by $\tau_1+p_s-\tau_2$ that exceeds its maximum allowed shifting σ_2 because of inequality (1). On the other hand, if we postpone the refueling of ω_1 starting it when the refueling of ω_2 is ended, then the shifting of ω_1 would be $\tau_2 + p_s - \tau_1$ violating its maximum allowed shifting σ_1 due to inequality (2).

($\Rightarrow$) Suppose that ω_1 and ω_2 are incompatible. If inequality (1) was not satisfied then it would be possible to postpone the refueling starting time of ω_2 immediately after the refueling ending time of ω_1 (given by $\tau_1 + p_s$) without exceeding its maximum allowed shifting since $\tau_1 + p_s - \tau_2 \leq \sigma_2$. In similar way, if inequality (2) was not satisfied then it would be possible to postpone the refueling starting time of ω_1 immediately after the refueling ending time of ω_2 (given by τ_2+p_s) without exceeding its maximum allowed shifting since $\tau_2 + p_s - \tau_1 \leq \sigma_1$. Hence, both the conditions (1) and (2) are necessary for the incompatibility. □

According to Proposition 4.1, the compatibility of two routes sharing an AFS with capacity 1 can be checked in polynomial time just verifying that at least one between inequality (1) and (2) is violated. More in general, if the routes are m, in order to establish if they are compatible or not, we have to proceed as follows. First of all, we order the routes for increasing earliest visit time of the shared station. If a route and the next one satisfy the inequalities (1) and (2) then the two routes are incompatible and we stop. Else if only one of the two inequalities is violated and the two routes overlap while refueling then only one of the two routes, $\overline{\omega}$, can be postponed in feasible way to avoid to overlap the other route while refueling. Then, in this case, its refueling starting time $\tau_{\overline{\omega}}$ and its shifting $\sigma_{\overline{\omega}}$ is updated, as well as its position in the ordered list of the routes for increasing visit time of the shared station. Else if both the inequalities (1) and (2) are violated and the two routes overlap while refueling then the overlap can be avoided in two feasible ways: either postponing the refueling starting time of the first route or that one of the second route. If making one of the two choices we arrive to an infeasibility (i.e. both (1) and (2) are satisfied), we have to backtrack to check if, with the other choice, the infeasibility can be avoided. If all the routes have been checked without detecting infeasibility, we stop stating that the m routes are compatible.

We observe that Proposition 4.1 is also useful when a station has a capacity greater than 1, e.g. 2, and the routes passing through this shared station are $m > 2$. Indeed, as soon as a pair of incompatible routes is found (i.e., they satisfy (1) and (2)), then the two routes must necessarily refuel on 2 different pumps and then we must verify that none of the $m-2$ remaining routes are incompatible simultaneously with both the route refueling at the first pump and the one refueling at the second pump.

Hereafter, we call the procedure that allows checking if a given set of m routes can be scheduled in such a way that the capacity of the shared station is not violated $Reschedule(r_1, \ldots, r_m)$. Therefore, this procedure returns $false$ if the routes are incompatible, otherwise it adjusts the visit time of the shared AFS in such a way that the AFVs do not overlap when refueling in the shared AFS.

5. A Greedy Randomized Adaptive Search Procedure for the G-VRP-CAFS

GRASP, introduced in Feo & Resende (1989), is a multi-start metaheuristic, and it has been widely applied to a large set of problems, including facility location Ferone et al. (2017), scheduling González-Neira et al. (2017), constrained shortest path Ferone et al. (2016), and vehicle routing problems Nguyen et al. (2012); Haddadene et al. (2016). For an extensive survey, the reader is referred to Festa & Resende (2002, 2009a,b).

Each GRASP iteration is characterized by two main phases: construction and local search. Starting with an empty solution, the construction phase iteratively adds an element, one at a time, until a complete solution is obtained. The elements that can be added to the solution make up the *Candidate List* (CL). The elements are sorted according to some greedy function, that measures the benefit of selecting each of them. Therefore, the construction phase creates the *Restricted Candidate List* (RCL) selecting the best elements of CL with respect to the greedy function. At the end, an element is randomly selected from the RCL. The aim of the local search is to improve the solution generated by the construction phase, returning a locally optimal solution with respect to some neighborhood structure. The pseudo-code of a generic GRASP for a minimization problem is reported in Algorithm 1.

5.1. Construction phase

In our proposal, the construction phase iteratively adds a route at a time until all customers are served or it is not possible to serve the remaining ones. The pesudo-code of the construction phase is reported in Algorithm 2, where $\oplus$ denotes the concatenation operator.

At each iteration, a new route is initialized with the depot (line 4). Therefore, the set of reachable locations is computed (line 6), and a location is randomly selected and added to the route (lines 8–9). When the set of reachable locations is empty, the route ends (lines 7 and 12). It is duly

Algorithm 1: Pseudo-code of a generic GRASP.

Output: Best found solution sol^*.

```
sol* ← NIL;
cost(sol*) ← +∞;
while a stop criterion is not met do
    Build a greedy randomized solution sol ;
    sol ← LocalSearch(sol) ;
    if cost(sol) < cost(sol*) then
        sol* ← sol;
```

Algorithm 2: Pseudo-code of the construction phase.

Input: The set of not served customers UC;
Input: Non-symmetric distribution function D.
Output: Solution sol.

```
sol ← ∅;
do
    added ← false;
    r ← [0] ;
    while true do
        CL ← getReachableLocations(UC, r) ;
        if CL = ∅ then break ;
        Randomly select i ∈ {1, ..., |CL|} according to distribution
          D ;
        r ← r ⊕ [CL_i] ;
        UC ← UC \ {CL_i};
        added ← true;
    r ← r ⊕ [0] ;
    if added then sol ← sol ∪ {r};
while added;
Reschedule(sol)
```

to note that we used the biased-randomized selection proposed in Ferone et al. (2019). Generally, the best elements of join the RCL and an element is randomly selected among them with a uniform distribution. In this case, we randomly select among all the elements according to a non-symmetric distribution that is biased towards the most promising solution elements.

A crucial aspect of the construction phase is represented by the function that returns the set of reachable locations, i.e., `getReachableLocations`. Let d_r and τ_r be the travel distance since the last refueling and the total travel time of the partial route r, respectively. Moreover, let i the last node of the partial route r, we define the following four sets.

- The set L_1 of all the unserved customers that can be served returning to the depot without violating the constraints on both the maximum distance travelled without refueling D_{max} and the maximum route duration T_{max}. Formally,

$$L_1 = \left\{ c \in UC \;\middle|\; \begin{array}{l} d_r + d_{ic} + d_{c0} \leq D_{max}, \\ \tau_r + t_{ic} + p_c + t_{c0} \leq T_{max}. \end{array} \right\}$$

- The set L_2 of all the unserved customers that can be served coming back to the depot after visiting a fueling station:

$$L_2 = \left\{ c \in UC \;\middle|\; \exists s \in F\colon \begin{array}{l} d_r + d_{ic} + d_{cs} \leq D_{max}, \\ \tau_r + t_{ic} + p_c + t_{cs} + p_s + t_{s0} \leq T_{max}. \end{array} \right\}$$

- The set L_3 of fueling stations that can be used to serve a customer respecting the constraints on both the maximum driving range and the maximum route duration:

$$L_3 = \left\{ s \in F \;\middle|\; \exists c \in UC\colon \begin{array}{l} d_r + d_{is} \leq D_{max}, \\ d_{sc} + d_{c0} \leq D_{max}, \\ \tau_r + t_{is} + p_s + t_{sc} + p_c + t_{c0} \leq T_{max}. \end{array} \right\}$$

- The set L_4 of fueling stations that can be used to come back to the depot:

$$L_4 = \left\{ s \in F \;\middle|\; \begin{array}{l} d_r + d_{is} \leq D_{max}, \\ \tau_r + t_{is} + p_s + t_{s0} \leq T_{max}. \end{array} \right\}$$

  If the route r contains an AFS, the set of reachable locations L is equal to L_1. On the contrary, L is defined as $L = \bigcup_{i=1}^{4} L_i$. All the locations $l \in L$ are sorted in non-decreasing order with respect to the distance d_{il}, and the greedy criterion prefers the shortest ones.

Finally (line 14), the function *Reschedule*, introduced in Section 4, tries to schedule the routes of the solution *sol* in such a way that they do no overlap when refuel in the same AFS. If this is not possible, the overlap is the same avoided violating the maximum duration time T_{max} of one of the incompatible routes, choosing, for each pair of incompatible routes, the route with the least violation of T_{max}. The infeasibility on the duration time is managed along the search procedure as explained in detail in Section 5.3.

5.2. Local search

Each GRASP iteration entails a local search as intensification phase. Generally, given a neighborhood function N, this procedures looks for a local optimum respect to N. Since during the years, several neighborhood functions have been proposed for routing problems, we adapted some of them to our problem.

In particular, as local search we used a Variable Neighborhood Descent (VND) approach whose pseudo-code is reported in Algorithm 3. For each $h = 1, \ldots, h_{\max}$, the h-th neighborhood of solution *sol* is explored obtaining a solution sol'. If the cost of sol' is lower than the current one, it replaces *sol* and the procedure restarts from the first neighborhood structure, otherwise it uses the next neighborhood structure.

The local search stops when all the neighborhoods have been explored without improving the solution. Consequently, the final solution is a local optimum with respect to all neighborhood functions.

Algorithm 3: Pseudo-code of the VND.

```
Input: Solution sol
Output: Improved solution sol
improved ← 1 ;
while improved = 1 do
    h ← 1 ;
    improved ← 0 ;
    while h ≤ h_max do
        sol' ← N_h(sol);
        if cost(sol') < cost(sol) then
            sol ← sol';
            improved ← 1;
            h ← 1 ;
        else
            h ← h + 1;
return sol ;
```

In the next sections, we analyze the neighborhoods used in the local search. The implemented VND explores the neighborhoods in the order here presented.

5.2.1. Change station

The Figure 1 depicts the first move used in the VND. It consists in changing the used fuel stations. All the routes in the solution are examined, and for each visited AFS s, the procedure checks if there exists another station s' that can be used to improve the solution quality. In this case, the route is modified inserting s' at the place of s.

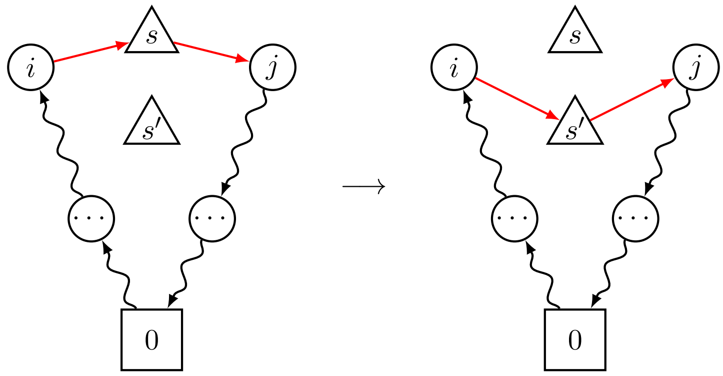

Figure 1: Change station move.

5.2.2. Destroy route

The second neighborhood structure consists in destroying a route and inserting its customers in the remaining routes. For each customer, the best insertion position in terms of solution quality is found. During the local search, the move is applied for each route, and the best found solution is selected.

5.2.3. 2-Opt

2-Opt is a classical neighborhood structure for routing problems, and the main idea behind it is to take a route that crosses over itself and reorder it so that it does not. An example is given in Figure 2. For each route, all possible pairs of arcs are examined, and if there exists at least one improving swapping, the best one is performed.

5.2.4. Remove AFS

The move consists in removing an AFS from a route, if it improves the solution. Therefore, we iterate over all routes and check if it is possible to obtain a better solution removing the stop at the station. The procedure removes all the stops that can be skipped.

5.2.5. Relocate

The neighborhood relocates a customer. The final position can belong either to the same starting route or to another one. The procedure scans all

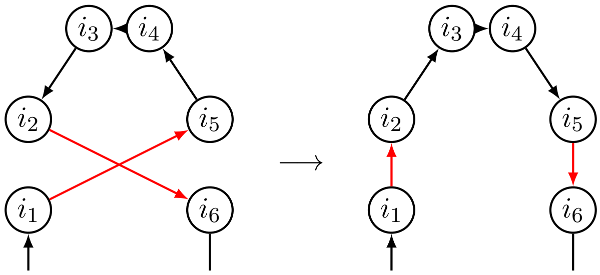

Figure 2: 2-Opt neighborhood.

the customers and checks if there exists at least another position to serve the customer that improves the solution. All the positions are inspected, and the best one is selected.

5.3. Infeasibility management

The construction phase (Section 5.1) can lead to an infeasible solution. Indeed, Algorithm 2 does not check that the capacity constraints at AFSs are satisfied, and the scheduling of the refuels can yield routes exceeding T_{max}.

Therefore, the local search aims to either improve the cost of a feasible solution or restore its feasibility. This second purpose is accomplished through a penalty function $\phi(sol)$ that maps each solution in $[0,1]$, where a feasible solution sol has $\phi(sol) = 0$. When comparing two solutions, all the local searches perform a lexicographic comparison on the pair penalty, routing cost. As a consequence, each feasible solution is better than all infeasible solutions.

Let $\hat{m}$ the number of used AFVs (i.e., the number of routes of the solution), $\hat{T}$ the maximum duration of the routes, and $\hat{d}$ the maximum travelled distance by a vehicle without refueling or before and after a possible refueling. We define $\bar{m} = \max\{\hat{m}, m\}$, $\bar{T} = \max\{T_{max}, \hat{T}\}$, and $\bar{d} = \max\{\hat{d}, D_{max}\}$, the penalty function ϕ is computed as

$$\phi(sol) = \frac{1}{3} \cdot \frac{\bar{m} - m}{\lceil 1.25m \rceil - m} + \frac{1}{3} \cdot \frac{\bar{T} - T_{max}}{1.5T_{max} - T_{max}} + \frac{1}{3} \cdot \frac{\bar{d} - D_{max}}{1.5D_{max} - D_{max}} \quad (3)$$

It is duly to note that all the solutions where at least one among $\hat{m}$, $\hat{T}$ and $\hat{d}$ exceeds $\lceil 1.25m \rceil$, $1.5T_{max}$ and $1.5D_{max}$ respectively, are discarded.

6. Computational Results

6.1. Problem Instances and parameters setting

Four sets of instances have been experimented, as detailed in the following. The first three sets have been introduced in Bruglieri et al. (2019b), whereas the last set has been introduced for the first time in this paper to have a larger number of customers. In all the sets of instances, the average speed v has been set equal to 40 miles per hour and for each AFS s, its capacity η_s is assumed to be equal to 1.

- EMH Set: 10 instances from the 40 proposed by Erdoğan & Miller-Hooks (2012), considered challenging from the capacity point of view. They have been in fact selected if in the optimal solution of the G-VRP there are at least two AFVs visiting the same AFS. The original values of the parameters have been kept, i.e., $T_{max} = 11$ hours, $p^{start} = 0.25$ hours, $p_s = 0.25$ hours $\forall s \in F$, $p_i = 0.5$ hours $\forall i \in I$, $Q = 60$ gallons and $r = 0.2$ gallons/miles (leading to $D_{max} = 300$ miles). The number of customers varies from 6 to 20, the number of AFSs from 3 to 10 and the number of AFVs from 3 to 8. As observed in Bruglieri et al. (2019b), since both the depot and the AFSs are embedded in the customers area, refuels may happen at very different times along the routes or do not happen at all. For this reason, they can be considered slightly challenging for the G-VRP-CAFS.

- TRIANGLE Set: it is composed of 10 instances with 15 customers, 3 AFSs and 10 vehicles. Also in this case, the original parameters setting is kept, i.e., $T_{max} = 11$, $p^{start} = 0$, $p_s = 0.5$ $\forall s \in F$, $p_i = 0.75$ $\forall i \in I$, $Q = 50$ and $r = 0.2$ (leading to $D_{max} = 250$). As observed in Bruglieri et al. (2019b), these instances are characterized by the fact that the stations lay in the middle between the depot and the customers. For this reason, each AFV needs to refuel, either at the beginning or at the end of its route leading to a more challenging scheduling problem. Therefore, these instances can be considered medium challenging.

- CENTRAL set: it is composed of 10 instances with 15 customers, only one AFS, at the center of the customers area, and 15 vehicles. Again, we maintain the original parameters setting and therefore, $T_{max} = 7$, $p^{start} = 0$, $p_s = 0.5$ $\forall s \in F$, $p_i = 0.5$ $\forall i \in I$, $Q = 32$ and $r = 0.2$ (leading to $D_{max} = 160$). These instances have the depot far from the customers and the travel time from the depot to the station is 2 hours, leading to a very small time available for refuels. Therefore, the station capacity may be a bottleneck and then, these instances are extremely challenging.

- Large-sized CENTRAL set: it is composed of 10 instances with 25, 50 and 100 customers. All the parameters are set as in the CENTRAL set. Also the layout is similar to that used in the CENTRAL set, i.e., the depot is far from the customers area and the travel time from the depot to the AFS is 2 hours. Moreover, the customers are clustered in clusters with at most 4 customers. These instances are generated with the following procedure. First of all, we compute the minimum number m of vehicles needed to serve the customers considering the ceiling of n divided by 4. We generate m random initial customers in a rectangle with height and base equal to 20 and 60, respectively, centered at the station. If n modulo 4 (i.e. the remainder of the division of n by 4) is 0, for each of the m initial customers we randomly generate 3 new customers so that they are no more far than a certain threshold σ from the initial customer. Otherwise, for each of the first $m - 1$ initial customers we randomly generate 3 new customers so that they are not more far than σ from the initial customer; for the $m - th$ initial customer we randomly generate (n modulo 4) -1 new customers so that they are not more far than σ from the $m - th$ customer. We chose $\sigma = 5$ to guarantee that the total distance travelled to serve the 4 customers associated with each route is lower than 20 miles, i.e. can be covered within half an hour (since the speed is of 40 miles/h). Indeed, this is the time that is left over considering $T_{max} = 7$ hours, 4 hours are necessary to reach the customer area from the depot and to go back to the depot, 2 hours are necessary to serve all the customers (being $p_i = 0.5 \; \forall i \in I$) and half an hour is necessary to refuel at the AFS.

In what follows, we describe the tuning phase for the parameter related to the construction of the initial solutions for the GRASP framework, that is characterized by the parameter β of the Geometric Distribution that guides Biased Randomization and belongs to the range $[0, 1]$.

To determine its value, we construct a training set, by randomly selecting the 30% of the instances EMH, TRIANGLE and CENTRAL. Once the set has been constructed, we performed an *iterated racing procedure*, implemented in **irace** package López-Ibáñez et al. (2016), to automatically find the best configuration of β. On the basis of the experiments carried out by using the irace package, the best performance are obtained by setting $\beta = 0.23$. Thus, this setting is used in the testing phase.

6.2. Results on benchmark instances

In this section, we compare the results found, on the first three sets of benchmark instances, by the proposed GRASP with those of the CP-Proactive, presented and showed to reach the best performances in Bruglieri et al. (2019b).

The results are reported in Tables 2–4. Each table is organized as described in the following. The first column reports the instance name whereas, the second and the third column indicate, respectively, the *Total Distance* (TD) and the total *CPU Time* (in seconds) of the CP-Proactive. The next two columns report the *Min Total Distance* (Min TD), i.e., the total distance obtained by the best solution found by the GRASP and the *Avg Total Distance* (Avg TD), i.e., the average total distance on all the five runs performed by the GRASP. Finally, the last two columns indicate the *Avg Time-to-best*, i.e., the average time within the GRASP method was suitable to find the best solution and the percentage *Gap* between the average total distance found by the GRASP method and the total distance of the CP-Proactive, computed as in the following:

$$GAP(\%) = \frac{(AvgTD - TD)}{TD} \cdot 100 \tag{4}$$

It is worth remarking that the total computational time given to the GRASP on these instances was set equal to 1 minute.

Concerning the results comparison shown in Table 2, the *Avg TD* always equals *TD* that means that all the solutions found by the GRASP on each instances are always optimal. Indeed, only on one instance (i.e., $S2\_6i6s$), the *Avg TD* is slightly greater than TD (of about 4.82%). However, the best solutions of the GRASP are found in an average CPU time that is by far less than that of the CP-Proactive (of about 61.37%).

Moreover, the proposed approach closes to optimality all the Triangle instances in an average CPU time that is again by far less than that required by the CP-Proactive (of about 91.82%).

Finally, on the Central instances, our GRASP finds the same solution of the CP-Proactive approach in the 60% of the cases, in an average CPU time that is by far less than that required by the CP-Proactive (of about 98.63%). Indeed, the only instance for which the solution of the GRASP has a significant percentage gap is $Central4$ that is also the case where the CP-Proactive reaches the CPU time limit of 1 hour. This may be due to two factors. Firstly, the instances of this set are extremely challenging concerning the AFS capacity, as explained in their description. Secondly, $Central4$ is the instance that has both the maximum average distance customer–depot, and the maximum average sum between distances customer–depot and customer–AFS.

6.3. Results on large-sized instances

Regarding the large-sized instances, generated in this work, it was prohibitive to solve them with the path-based approaches proposed in Bruglieri et al. (2019b). Indeed, on the instances with 25 customers, almost 300 thousands of *pairs* are generated on average, on the instances with 50 customers more than 2 millions and on the instances with 100 customers even the phase of generating all the feasible non-dominated paths failed since exceeding the CPU time limit of 1 hour. Therefore, in the following, only the results found by the proposed GRASP are presented. In this case, the total computational time given to the GRASP is equal to 2 minutes.

Table 2: Results on EMH instances

Instance	CP-Proactive		GRASP			
	TD	CPU Time	Min TD	Avg TD	Avg Time-to-best	GAP(%)
20c3sC5	2156.01	1.45	2156.01	2156.01	0.23	0.00
20c3sC6	2758.17	0.3	2758.17	2758.17	0.22	0.00
20c3sC7	1393.99	0.49	1393.99	1393.99	0.02	0.00
20c3sC8	3139.72	0.65	3139.72	3139.72	0.88	0.00
20c3sC10	2583.42	6.82	2583.42	2583.42	0.10	0.00
20c3sU1	1797.49	13.42	1797.49	1797.49	0.75	0.00
S2_2i6s	1633.1	65.58	1633.10	1633.10	3.46	0.00
S2_6i6s	2431.33	2.31	2431.33	2548.50	29.61	4.82
S2_8i6s	2158.35	0.81	2158.35	2158.35	0.18	0.00
S2_10i6s	1585.46	1.95	1585.46	1585.46	0.53	0.00
Average	**2227.95**	**10.20**	**2227.95**	**2240.97**	**3.94**	**0.58**

Table 3: Results on Triangle instances

Instance	CP-Proactive		GRASP			
	TD	CPU Time	Min TD	Avg TD	Avg Time-to-best	GAP(%)
Triangle1	1871.61	1.04	1871.61	1871.61	1.63	0.00
Triangle2	2191.73	4.09	2191.73	2191.73	0.52	0.00
Triangle3	1872.12	4.56	1872.12	1872.12	0.59	0.00
Triangle4	1869.07	4.52	1869.07	1869.07	0.05	0.00
Triangle5	1852.73	11.02	1852.73	1852.73	0.19	0.00
Triangle6	1865.49	5.25	1865.49	1865.49	0.33	0.00
Triangle7	1898.00	3.10	1898.00	1898.00	1.11	0.00
Triangle8	2197.49	13.76	2197.49	2197.49	0.20	0.00
Triangle9	1862.50	6.30	1862.50	1862.50	0.07	0.00
Triangle10	1864.73	5.04	1864.73	1864.73	0.07	0.00
Average	**1934.55**	**5.87**	**1934.55**	**1934.55**	**0.48**	**0.00**

Table 4: Results on Central instances

Instance	CP-Proactive		GRASP			
	TD	CPU Time	Min TD	Avg TD	Avg Time-to-best	GAP(%)
Central1	953.94	130.61	953.94	953.94	44.72	0.00
Central2	948.69	441.91	959.88	959.88	10.19	1.18
Central3	943.12	3600.00	958.94	959.37	15.78	1.72
Central4	967.96	3600.00	1099.24	1110.27	23.59	14.70
Central5	714.55	2.98	714.55	714.55	0.05	0.00
Central6	844.43	1348.23	844.43	844.43	0.17	0.00
Central7	862.68	148.07	862.68	862.68	22.11	0.00
Central8	712.83	4.60	712.83	712.83	0.33	0.00
Central9	855.43	22.10	855.43	855.43	0.11	0.00
Central10	901.19	453.02	905.59	906.29	16.23	0.57
Average	**870.48**	**975.15**	**886.75**	**887.97**	**13.33**	**2.01**

In particular, each of the next three tables (Tables 5–7) reports, for each instance, on all the solutions found, the Minimum Travel Distance ($MinTD$), the Average Total Distance ($AvgTD$), the Average Time-to-best (in seconds) and finally, the average number of AFVs used and of refuelings required.

On the instances with 25 customers, the solutions found by the GRASP differ from the best ones of about 0.19%, on average. The average number of EVs used is equal to 6.30, each of them refueling once except in one case (instance Central25_9 where there is one EV not refueling).

On the instances with 50 customers, the solutions found are far from the best ones on average of about 0.87% using 12.94 EVs, each of them refueling at most once.

Finally, on the instances with 100 customers,the solutions found are far from the best ones on average of about 0.53% using about 25 EVs, each of them refueling almost always once.

Table 5: Results on Central instances with 25 customers

Instance	GRASP				
	Min TD	Avg TD	Avg Time-to-best	AFVs used	Refuelings
Central25_1	1129.85	1135.88	73.56	6.00	6.00
Central25_2	1113.97	1115.14	70.92	6.00	6.00
Central25_3	1321.82	1323.10	62.62	7.00	7.00
Central25_4	1122.73	1123.73	80.97	6.00	6.00
Central25_5	1110.36	1116.28	87.55	6.00	6.00
Central25_6	1090.04	1090.39	44.18	6.00	6.00
Central25_7	1105.04	1106.89	37.15	6.00	6.00
Central25_8	1141.23	1144.17	47.77	6.00	6.00
Central25_9	1281.57	1281.98	42.65	7.00	6.80
Central25_10	1311.67	1313.26	82.07	7.00	7.00
Average	**1172.83**	**1175.08**	**62.94**	**6.30**	**6.28**

On all the tested instances we verified the effectiveness of the different moves considering the average percentage improvement provided by each kind of move. The most effective move is the *Destroy route* with an average improvement of 3.74%. The second more effective move is the *Relocate* move with an average improvement of 0.30%, followed by the $2 - Opt$ and $RemoveAFS$ moves with an average improvement of 0.25% and 0.11%, respectively.

7. Conclusions and Future Work

In this paper, we proposed a Greedy Randomized Adaptive Search Procedure (GRASP) for the Green Vehicle Routing Problem with Capacitated

Table 6: Results on Central instances with 50 customers

Instance	GRASP				
	Min TD	Avg TD	Avg Time-to-best	AFVs used	Refuelings
Central50_1	2520.33	2533.43	78.06	13.00	13.00
Central50_2	2435.78	2448.75	71.69	13.00	12.80
Central50_3	2425.64	2447.54	58.90	13.00	13.00
Central50_4	2271.14	2330.86	71.00	12.40	12.00
Central50_5	2467.29	2482.22	61.25	13.00	13.00
Central50_6	2422.87	2457.04	62.84	13.00	12.60
Central50_7	2405.19	2435.18	73.27	13.00	12.00
Central50_8	2487.32	2487.32	44.37	13.00	13.00
Central50_9	2450.04	2458.80	49.76	13.00	13.00
Central50_10	2473.63	2489.88	73.41	13.00	13.00
Average	**2435.92**	**2457.10**	**64.46**	**12.94**	**12.74**

Table 7: Results on Central instances with 100 customers

Instance	GRASP				
	Min TD	Avg TD	Avg Time-to-best	AFVs used	Refuelings
Central100_1	4966.03	5000.79	82.23	26.00	26.00
Central100_2	4730.11	4746.06	90.05	25.00	24.67
Central100_3	4757.50	4782.34	87.17	25.00	25.00
Central100_4	4697.35	4726.88	80.93	25.00	25.00
Central100_5	4732.49	4753.31	120.93	25.00	25.00
Central100_6	4524.44	4603.63	91.32	24.40	24.40
Central100_7	4786.57	4790.82	57.06	25.00	25.00
Central100_8	4784.18	4808.73	40.25	25.00	25.00
Central100_9	4795.70	4795.70	66.51	25.00	25.00
Central100_10	4686.57	4705.79	63.06	25.00	25.00
Average	**4746.09**	**4771.40**	**77.95**	**25.04**	**25.01**

Alternative Fuel Stations (G-VRP-CAFS). The G-VRP-CAFS aims at routing a fleet of Alternative Fuel Vehicles (AFVs), each one starting and ending in a common depot, serving a set of customers at the minimum total distance, within a maximum duration. Along the routes, each AFV is allowed to refuel at the Alternative Fuel Stations (AFSs) supposed to have a limited number of fueling pumps. This implies that only a certain number of AFVs can simultaneously refuel at the same station. The G-VRP-CAFS was introduced in Bruglieri et al. (2019b) and solved through an exact approach on a set of benchmark instances with up to 20 customers.

The aim of this work was to design a metaheuristic based on the GRASP framework in order to efficiently address also large-sized instances of the G-VRP-CAFS with even 100 customers. Some theoretical results on the compatibility among routes were also detected in order to properly guide the GRASP. Moreover, along the local search, possible violations on the route duration were allowed and properly faced through a penalty objective function in order to better explore the neighborhoods.

Computational results were carried out on both the sets of benchmark instances taken from Bruglieri et al. (2019b) and a set of large-sized instances specifically introduced in this paper. In particular, the results obtained by the GRASP on the benchmark instances were compared with those found by the CP-Proactive proposed and shown to reach the best performances in Bruglieri et al. (2019b). The proposed GRASP was suitable to close to optimality about 83% of the instances with an average time-to-best (the time required by the GRASP to obtain the best solution among those generated) that is by far less (about 98%) than the total computational time required by the CP-Proactive. Finally, the GRASP was shown to efficiently address also the large-sized instances in an average time-to-best equal to about 1 minute.

Future works may concern the extension of the proposed procedure to similar problems, such as, the VRP with Intermediate Facilities and the Electric VRP where, unlike the G-VRP, the refueling time is not constant but depends on the vehicle state of charge.

References

Affi, M., Derbel, H., & Jarboui, B. (2018). Variable neighborhood search algorithm for the green vehicle routing problem. *International Journal of Industrial Engineering Computations*, *9*, 195–204.

Agency, E. E. (2020). Greenhouse gas emission intensity of fuels and biofuels for road transport in europe, .

Andelmin, J., & Bartolini, E. (2017). An exact algorithm for the green vehicle routing problem. *Transportation Science*, *51*, 1288–1303.

Andelmin, J., & Bartolini, E. (2019). A multi-start local search heuristic for the green vehicle routing problem based on a multigraph reformulation. *Computers & Operations Research*, *109*, 43–63.

Asghari, M., Al-e, S. M. J. M. et al. (2020). Green vehicle routing problem: A state-of-the-art review. *International Journal of Production Economics*, (p. 107899).

Bektaş, T., Demir, E., & Laporte, G. (2016). Green vehicle routing. In H. N. Psaraftis (Ed.), *Green Transportation Logistics: The Quest for Win-Win Solutions* (pp. 243–265). Cham: Springer International Publishing. URL: https://doi.org/10.1007/978-3-319-17175-3_7. doi:10.1007/978-3-319-17175-3_7.

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*, 807–823.

Bruglieri, M., Mancini, S., Pezzella, F., & Pisacane, O. (2016). A new mathematical programming model for the green vehicle routing problem. *Electronic Notes in Discrete Mathematics*, *55*, 89–92.

Bruglieri, M., Mancini, S., Pezzella, F., & Pisacane, O. (2019a). A path-based solution approach for the green vehicle routing problem. *Computers & Operations Research*, *103*, 109–122.

Bruglieri, M., Mancini, S., Pezzella, F., Pisacane, O., & Suraci, S. (2017). A three-phase matheuristic for the time-effective electric vehicle routing problem with partial recharges. *Electronic Notes in Discrete Mathematics*, *58*, 95–102.

Bruglieri, M., Mancini, S., & Pisacane, O. (2019b). The green vehicle routing problem with capacitated alternative fuel stations. *Computers & Operations Research*, *112*, 104759.

Bruglieri, M., Mancini, S., & Pisacane, O. (2019c). More efficient formulations and valid inequalities for the green vehicle routing problem. *Transportation Research Part C: Emerging Technologies*, *105*, 283–296.

Bruglieri, M., Pezzella, F., Pisacane, O., Suraci, S. et al. (2015). A variable neighborhood search branching for the electric vehicle routing problem with time windows. *Electron. Notes Discret. Math.*, *47*, 221–228.

Ceselli, A., Felipe, Á., Ortuño, M. T., Righini, G., & Tirado, G. (2021). A branch-and-cut-and-price algorithm for the electric vehicle routing problem with multiple technologies. In *SN Operations Research Forum* (pp. 1–33). Springer volume 2.

da Costa, P. R. d. O., Mauceri, S., Carroll, P., & Pallonetto, F. (2018). A genetic algorithm for a green vehicle routing problem. *Electronic notes in discrete mathematics*, *64*, 65–74.

Desaulniers, G., Errico, F., Irnich, S., & Schneider, M. (2016). Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research*, *64*, 1388–1405.

Erdoğan, S., & Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation research part E: logistics and transportation review*, *48*, 100–114.

Felipe, Á., Ortuño, 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.

Feo, T. A., & Resende, M. G. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, *8*, 67–71. doi:10.1016/0167-6377(89)90002-3.

Ferone, D., Festa, P., Guerriero, F., & Laganà, D. (2016). The constrained shortest path tour problem. *Computers and Operations Research*, *74*, 64–77. doi:10.1016/j.cor.2016.04.002.

Ferone, D., Festa, P., Napoletano, A., & Resende, M. G. C. (2017). A new local search for the p-center problem based on the critical vertex concept. In R. Battiti, D. E. Kvasov, & Y. D. Sergeyev (Eds.), *Learning and Intelligent Optimization. LION 2017* (pp. 79–92). Cham: Springer volume 10556 of *Lecture Notes in Computer Science*. doi:10.1007/978-3-319-69404-7_6.

Ferone, D., Gruler, A., Festa, P., & Juan, A. A. (2019). Enhancing and extending the classical grasp framework with biased randomization and simulation. *Journal of the Operational Research Society*, *70*, 1362–1375. doi:10.1080/01605682.2018.1494527.

Festa, P., & Resende, M. G. C. (2002). GRASP: An annotated bibliography. In *Essays and surveys in metaheuristics* (pp. 325–367). Springer.

Festa, P., & Resende, M. G. C. (2009a). An annotated bibliography of GRASP - Part I: Algorithms. *International Transactions in Operational Research*, *16*, 1–24.

Festa, P., & Resende, M. G. C. (2009b). An annotated bibliography of GRASP - Part II: Applications. *International Transactions in Operational Research*, *16*, 131–172.

Goeke, D. (2019). Granular tabu search for the pickup and delivery problem with time windows and electric vehicles. *European Journal of Operational Research*, *278*, 821–836.

González-Neira, E. M., Ferone, D., Hatami, S., & Juan, A. A. (2017). A biased-randomized simheuristic for the distributed assembly permutation flowshop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, *79*, 23 – 36. doi:10.1016/j.simpat.2017.09.001.

Haddadene, S. R. A., Labadie, N., & Prodhon, C. (2016). A GRASP × ILS for the vehicle routing problem with time windows, synchronization and precedence constraints. *Expert Systems with Applications*, *66*, 274–294. doi:10.1016/j.eswa.2016.09.002.

Hooshmand, F., & MirHassani, S. (2019). Time dependent green vrp with alternative fuel powered vehicles. *Energy Systems*, *10*, 721–756.

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.

Keskin, M., & Çatay, B. (2018). A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. *Computers & Operations Research*, *100*, 172–188.

Keskin, M., Çatay, B., & Laporte, G. (2020). A simulation-based heuristic for the electric vehicle routing problem with time windows and stochastic waiting times at recharging stations. *Computers & Operations Research*, *125*, 105060.

Keskin, M., Laporte, G., & Çatay, B. (2019). Electric vehicle routing problem with time-dependent waiting times at recharging stations. *Computers & Operations Research*, *107*, 77–94.

Koç, Ç., & Karaoglan, I. (2016). The green vehicle routing problem: A heuristic based exact solution approach. *Applied Soft Computing*, *39*, 154–164.

Leggieri, V., & Haouari, M. (2017). A practical solution approach for the green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, *104*, 97–112.

López-Ibáñez, Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., & Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, *3*, 43–58. doi:10.1016/j.orp.2016.09.002.

Lu, J., Chen, Y., Hao, J.-K., & He, R. (2020). The time-dependent electric vehicle routing problem: Model and solution. *Expert Systems with Applications*, (p. 113593).

Madankumar, S., & Rajendran, C. (2018). Mathematical models for green vehicle routing problems with pickup and delivery: A case of semiconductor supply chain. *Computers & Operations Research*, *89*, 183–192.

Moghdani, R., Salimifard, K., Demir, E., & Benyettou, A. (2020). The green vehicle routing problem: a systematic literature review. *Journal of Cleaner Production*, (p. 123691).

Montoya, A., Guéret, C., Mendoza, J. E., & Villegas, J. G. (2016). A multi-space sampling heuristic for the green vehicle routing problem. *Transportation Research Part C: Emerging Technologies*, *70*, 113–128.

Montoya, A., Guéret, 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.

Nguyen, V.-P., Prins, C., & Prodhon, C. (2012). Solving the two-echelon location routing problem by a grasp reinforced by a learning process and path relinking. *European Journal of Operational Research*, *216*, 113–126. doi:10.1016/j.ejor.2011.07.030.

Poonthalir, G., & Nadarajan, R. (2018). A fuel efficient green vehicle routing problem with varying speed constraint (f-gvrp). *Expert Systems with Applications*, *100*, 131–144.

Poonthalir, G., & Nadarajan, R. (2019). Green vehicle routing problem with queues. *Expert Systems with Applications*, *138*, 112823.

Schneider, M., Stenger, A., & Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, *48*, 500–520.

Soysal, M., Cimen, M., & Demir, E. (2018). On the mathematical modeling of green one-to-one pickup and delivery problem with road segmentation. *Journal of cleaner production*, *174*, 1664–1678.

Toth, P., Vigo, D., for Industrial, S., & Mathematics, A. (2015). *Vehicle Routing: Problems, Methods, and Applications*. MOS-SIAM series on optimization. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104). URL: https://books.google.it/books?id=YL_CswEACAAJ.

Xu, Z., Elomri, A., Pokharel, S., & Mutlu, F. (2019). A model for capacitated green vehicle routing problem with the time-varying vehicle speed and soft time windows. *Computers & Industrial Engineering*, *137*, 106011.

Yu, Y., Wang, S., Wang, J., & Huang, M. (2019). A branch-and-price algorithm for the heterogeneous fleet green vehicle routing problem with time windows. *Transportation Research Part B: Methodological*, *122*, 511–527.

Yu, Z., Zhang, P., Yu, Y., Sun, W., & Huang, M. (2020). An adaptive large neighborhood search for the larger-scale instances of green vehicle routing problem with time windows. *Complexity*, *2020*.

Zhang, S., Gajpal, Y., & Appadoo, S. (2018). A meta-heuristic for capacitated green vehicle routing problem. *Annals of Operations Research*, *269*, 753–771.