

A Path-based exact solution approach for the Green Vehicle Routing Problem

Maurizio Bruglieri

Dipartimento di Design, Politecnico di Milano

Simona Mancini*

Dipartimento di Matematica e Informatica, Università di Cagliari

Ferdinando Pezzella

Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche

Ornella Pisacane

Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche

Abstract

The Green Vehicle Routing Problem aims routing Alternative Fuel Vehicles, based at a depot, minimizing the total travel distance. Each vehicle handles a subset of customers, leaving from and returning to the depot, respecting a maximum duration and a distance traveled without refuels. We propose a two phases exact approach, composing each route through two/more paths. Each path serves a subset of customers without intermediate refuels. Dominance rules limit the number of feasible paths. Our approach, tested on benchmark instances, strongly outperforms the existing exact methods and it can be generalized to solve other VRP with Intermediate Stops.

Keywords:

Vehicle Routing Problem, Alternative Fuel Vehicles, Dominance rules, Mixed Integer Linear Programming formulation

*Corresponding author

Email address: `simona.mancini@unica.it`, `simona.mancini@polito.it` (Simona Mancini)

1. Introduction

Since the industrial revolution, we are assisting to an increment of the global atmospheric CO_2 emissions. In the last 7 years (from 2010 to 2017), the increment has been of about 5% of the CO_2 emissions, passing from the 388.72 part per million (ppm) of 2010 to the 407.05 ppm of 2017 (data updated on July¹).

In particular, the transportation sector is responsible of about the 23% of the global CO_2 emissions, percentage that is forecast to double in 2050. For this reason, European Commission is going to restrict the use of the internal combustion engine vehicles in order to incentivize the use of the so called Green Vehicles. Such vehicles use alternative fuel, e.g., methanol, electricity, hydrogen and so on and therefore, they are also called *Alternative Fuel Vehicles* (AFVs).

A significant reduction of the global harmful emissions is absolutely one of the main advantages guaranteed by the AFVs. They are also characterized by a kilometer cost lower than that of the traditional internal combustion engine vehicles. Moreover, they are allowed to reach also the Limited Traffic Zones, i.e., urban areas where the traffic is restricted to specific vehicles. This is a considerable advantage, for instance, in the Last Mile Logistics and also in the Dial-a-Ride systems where a door-to-door transportation service is required.

However, despite these advantages, they have a limited drive range and therefore, they may require to be refueled more than once during a daily trip ([18]). In addition, the *Alternative Fuel Stations* (AFSs) are currently not widespread on the territory. This means that it becomes very significant to a priori define the routes of these vehicles in order to avoid drivers to remain stuck along their route, i.e., without the sufficient fuel level either to go back to depot or to reach the closest AFS.

A relatively new research topic, called the *Green Vehicle Routing Problem* (GVRP), aims to properly route the AFVs, planning their stops at the AFSs. The GVRP belongs to the most general class of Vehicle Routing Problems (VRPs), recently reviewed in [23] and it was introduced in the literature by the seminal work of [6]. The GVRP aims to serve a set I of customers, geographically distributed, by using a homogenous fleet of m AFVs, based on a common depot, with the objective of minimizing their total traveled

¹www.co2.earth

distance. It is defined on a complete and directed graph $G = (N, A)$ where N denotes the set of nodes, containing I , the depot (node 0) and the set F of AFSs, while A is the set of arcs. For each AFS s , the service time p_s is known, such as, the time to fully refuel an AFV. Similarly, for each customer $i \in I$, the service time p_i is given. For each $(i, j) \in A$, the travel distance d_{ij} as well as the time t_{ij} necessary to go from node i to node j are given. For each AFV, both the maximum fuel capacity Q and the average speed v are known. Moreover, the fuel consumption is assumed to be linearly proportional to the travel distance through the input constant value r (fuel consumption rate). Each AFV can perform a route starting from the depot and ending to it, without exceeding the maximum duration T_{max} (duration constraint). Due to the limited fuel autonomy, each AFV may require to stop at the AFSs during its trip, since after a full refuel it can travel a maximum distance given by $D_{max} = Q/r$ (fuel level constraint). It is assumed that the AFVs leave the depot fully refueled and perform a full refuel each time they stop at an AFS. A feasible solution for the GVRP consists in a set of at most m routes (one for each available AFV).

In Figure 1, a solution example is shown for the instance 20c3sU2 of [6], where 20 customers (nodes with prefix C) have to be served by at most $m = 6$ AFVs. For example, the route handling customers C_4, C_{12}, C_{10}, C_7 does not require any stop at AFSs since its total distance is lower than D_{max} . Instead, the route handling customers $C_{18}, C_9, C_{13}, C_6, C_8$, requires to stop at an AFS since its total distance is greater than D_{max} .

Each route can be seen as the composition of paths serving a subset of customers without intermediate stops at AFSs. Each path can be of one of the following types: between the depot and an AFS; between an AFS and the depot; between two AFSs and finally, between the depot and itself. In the latter case, the path represents the whole route. For example, the route $\{D, C_{18}, C_9, C_{13}, AFS, C_6, C_8, D\}$ can be seen as the composition of the path $\{D, C_{18}, C_9, C_{13}, AFS\}$ and the path $\{AFS, C_6, C_8, D\}$.

A similar approach has been used in the literature for a problem of metro network design, [13] and has been proved to effective and efficient.

It is worth noting that, differently from the general VRPs, where the number of all the paths increases exponentially with the number of nodes in the graph, in the GVRP, due to both the maximum duration and the fuel level constraints, such a number is often limited. This means that some paths can be a-priori excluded from the set of all the feasible ones.

Our paper aims to propose a new approach for solving the GVRP to the

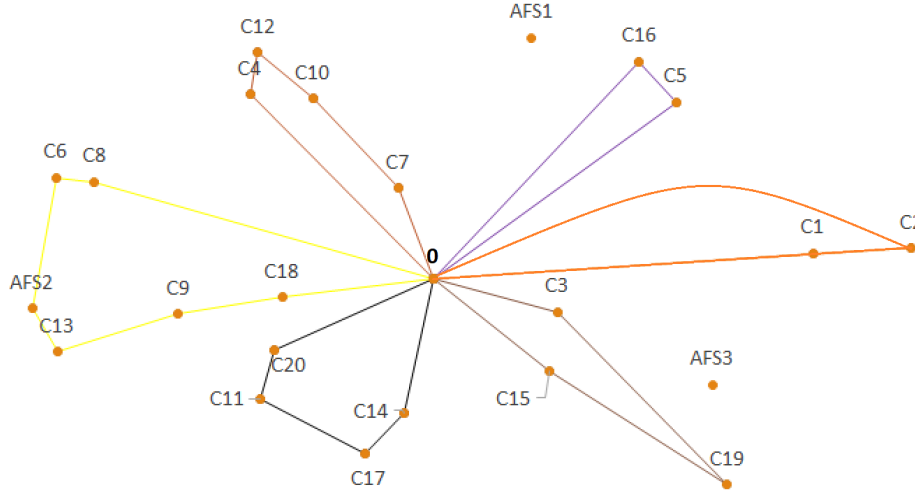


Figure 1: Optimal GVRP solution on the benchmark instance 20c3sU2 introduced in [6] with $D_{max} = 300$ miles, $T_{max} = 11$ hours, $v = 40$ miles/hours. The service times of the customers are all equals to 0.5 hours while the ones at the AFSs are all equal to 0.25 hours.

optimality in reasonable amount of time that, to the best of our knowledge, outperforms all the existing solution methods on the medium size benchmark instances. It is based on two phases. In the first phase, the set of all the feasible paths is generated, removing from it, all those dominated. Moreover, the set of all compatible pairs of paths is also determined. This is done in order to speed-up the second phase in which the paths are combined for creating the routes of the final solution. This second phase is solved through a set partitioning based formulation, that combines only pairs of feasible and not dominated paths, minimizing the total traveled distance without exceeding T_{max} .

Therefore, compared to the state-of-art, our work contributes to:

- solve the GVRP to the optimality in reasonable amount of time through a path-based Mixed Integer Linear Programming (MILP) formulation, easily extendable to similar problems like VRP with Intermediate Stops and GVRP with multiple recharging technologies;
- introduce feasibility rules for generating paths and pairs of paths;
- define dominance criteria among the paths.

The rest of this paper is organized as in the following. Section 2 reviews the state-of-the art on both exact and heuristic/meta-heuristic approaches proposed for solving the GVRP. Section 3 details the exact solution approach proposed in this paper for solving the GVRP. Section 4 shows the numerical results found by our solution approach on some benchmark instances taken from literature. This section is also devoted to numerical comparisons with the already existing solution approaches on both solutions quality and computational times. Finally, Section 5 draws some conclusions and highlights some useful future research directions to be investigated.

2. Literature Review

The GVRP and its several variants are attracting over the years the interest of the scientific community, specially the one of the operations researchers.

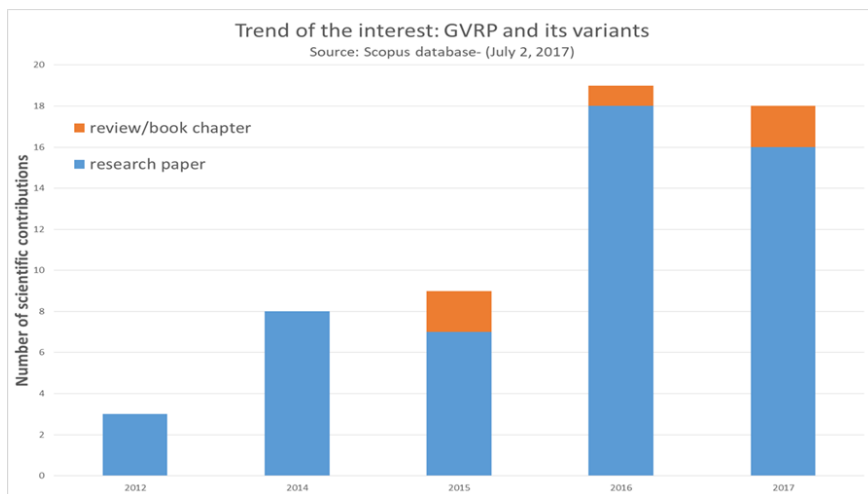


Figure 2: Scientific contributions on the GVRP and its variants

Figure 2 shows the increasing interest, justified by an increasing trend of the production of scientific contributions (scientific papers, reviews and book chapters).

As shown in the figure, the GVRP was introduced in 2012 by the paper [6]. The authors represent it on a directed completed graph and model it through a MILP formulation (hereafter denoted as EMH) in which, beyond the traditional arc routing variables, the ones on both the fuel level and

the arrival time at each node are introduced. The objective function to be minimized denotes the total travel distance and beyond the traditional flow conservation constraints, it is imposed that: each customer is handled exactly once; the number of AFVs leaving the depot and returning to it is limited to a fixed value m . With reference to this last aspect, they properly clone the AFSs in order to allow that each route is an elementary cycle, i.e., the routes do not share AFSs. In this way, an AFS may be used more than one time in the same route and eventually also in more routes. However, a critical issue of the formulation proposed in [6] is determining the proper number of AFS clones to introduce: if it is too low, beneficial multiple stops at AFSs may be restricted and thus, the formulation may be not exact. On the contrary, if it is too large, the network size increases a lot, making very time consuming the solution of the corresponding MILP. For the experimental campaign, the authors introduce the GVRP benchmark instances, distinguished in medium and large scale case studies. In order to solve the large scale instances in reasonable amount of time, they propose both a Modified Clark and Wright Saving heuristic and a Density-Based Clustering algorithm together with a post-optimization procedure in order to improve the solution quality.

In 2014, the work of [20] studied the specific case in which the fleet is composed of only Electric Vehicles, (EVs). Such a problem aims to handle a set of customers, minimizing the total travel distance and taking into consideration, among the traditional arc routing constraints, the following additional ones: the demand of each customer has to be served in exactly one route and within a specified time window; the cargo capacity of each EV cannot be exceeded as well as its battery capacity can never go under zero; each EV is fully recharged at a station and the battery consumption is assumed to be linearly proportional to the travel distance. For such Electric Vehicle Routing Problem with Time Windows and Recharge Stations, (EVRP-TW) they propose both a MILP formulation and a Variable Neighborhood Search combined with a Tabu Search (VNS-TS). In the experimental campaign, they also use the GVRP benchmark instances, showing that the proposed VNS-TS significantly outperforms the results found by [6]. In the same year, in [7], a variant of the GVRP was introduced. The authors, in fact, take into account the possibility to use multiple technologies and partial recharges at the stations. In such a case, the objective function to minimize represents the total cost given by three components: the first related to the cost for recharging an EV at the depot during the night, the second associated with the cost for recharging an EV at the stations and finally, the third cost component

due to the battery life cycle. They propose a Simulated Annealing and a Local Search, called *48A* and based on 48 combinations of 6 different neighborhoods. On the GVRP benchmark instances, the *48A* outperforms the VNS-TS. Recently, in [24], the problem of scheduling a fleet of electric buses or vehicles has been addressed assuming both fully and partial recharges at the stations. With the aim of minimizing both the number of EVs and the total traveling distance, the authors formulate a MILP model and design an adaptive large neighborhood search heuristic.

In 2015, in [21], the VRP with Intermediate Stops (VRPIS) was introduced in order to generalize the GRVP. In fact, the VRPIS represents all the cases in which vehicles have to stop at certain intermediate facilities to replenish/unload their cargo and/or to be refueled. The authors propose an Adaptive Variable Neighborhood Search (AVNS) that applied on the GVRP benchmark instances significantly outperforms the existing approaches especially for what concerns the solution quality.

In 2016, a first attempt to get rid of the AFS clones was described in [12] in which the authors propose a three-index MILP (hereafter indicated as KK), where the arc routing and binary variables indicate if an AFV stops at an AFS going from a customer to another one. They solve the proposed model through a Branch-and-Cut approach. In the same year, in [19], a two phase solution approach (hereafter called MSH) for the GVRP was proposed. In particular, a Randomized route-first cluster second heuristic is firstly used for generating a set of feasible routes. Then, such a feasible set is given in input to a set partitioning formulation for determining a solution of the GVRP. To the best of our knowledge, it outperforms all the other heuristics and meta-heuristics designed for the GVRP.

In [17], the Hybrid Vehicle Routing Problem was addressed in which the decision process is related to decide both when to recharge and when to switch from an internal combustion engine vehicle to an EV. The author proposes a Large Neighborhood Search based Matheuristic (hereafter called MH) and she also uses the GVRP benchmark instances for the experimental campaign. In [14], a new exact solution approach for solving the GVRP was proposed. Starting from a non-linear formulation, they propose a MILP formulation (hereafter indicated as LH1) together with a reduction procedure. In addition, both valid inequalities, i.e., a 2-customer sub-tour elimination constraints and a pre-processing procedure, i.e., the fixing of some binary variables, are also described. Therefore, the authors propose and experiment other four different versions of LH1: LH2, i.e., LH1 with the valid inequal-

ities; LH3, i.e., LH1 with the reduction procedure and the fixing; LH4, i.e., LH2 with the reduction procedure and the fixing and finally, LH5, i.e., LH4 with the relaxation on the maximum number of routes. Experimental results on the GVRP benchmark instances show that their formulations outperform both EMH and KK. Very recently, in [1], a set partitioning formulation was proposed for modeling the GVRP. Feasible routes are firstly generated as simple circuits in a multigraph in which each node is associated with a customer while an arc represents a not dominated path between two customers that eventually uses AFSs. The set partitioning formulation is strengthened through valid inequalities and it is then solved with a column generation approach. Numerical results show that the proposed exact solution approach is suitable to solve to the optimality some large benchmark instances in an average computational time of 3 hours. However, numerical results on the medium scale benchmark instances are not provided. As reported in [15], columns generation methods are very effective but their convergence is very slow. Therefore they are generally suitable to solve large instances in long computational times, but they are not suited to solve smaller problems in very fast computational times.

Over the years, several and different other versions of the GVRP were proposed. Just to cite a few, a GVRP with cross-docking was addressed in [25] while the Green Inventory Routing Problems with Heterogeneous Fleets was studied in [5]. In [4], the Electric Vehicle Routing Problem with Time Windows and Partial Recharges was introduced and addressed through a Variable Neighborhood Search Branching while the same problem was addressed by an Adaptive Large Neighborhood Search algorithm in [11]. In [10], the authors addressed a Pick-up and Delivery Problem with soft Time Windows, through a fleet of EVs. Such a problem is formulated as a multi-objective MILP model and the objective function to minimize denotes the total travel distance, the total cost for the EVs used and the total penalty cost for the unsatisfied time windows. In [3], a three-phase matheuristic is designed for solving the Electric Vehicle Routing Problem with Time Windows and Recharge Stations. It mainly combines an exact method with a VNS local Branching (VNSB). Recently two new extensions of the EVRP have been proposed: in [9] the authors dealt with an EVRP with non linear recharging functions, while in [8] they considered capacitated charging stations. The problem of dealing with capacitated alternative fuel stations have been introduced also for the GVRP by [2]. In [16], two MILPs have been proposed for routing a set of AFVs considering pick-up and delivery operations in a

Semiconductor Supply Chain. While, in [22], the Green Time Dependent Capacitated Vehicle Routing Problem has been addressed through dynamic programming.

In this paper, we address the GVRP by proposing a new exact solution approach based on generating feasible not dominated paths and solving a set partitioning based MILP formulation to compose the routes of the final solution. The numerical results, found on the medium scale GVRP benchmark instances, show that our approach is suitable to dominate all the already existing others. It is worth remarking that we cannot compare our approach with the one proposed in [1] since they do not report the results on those instances.

3. A path-based exact solution approach

Starting from the assumptions specified in Section 1, we propose a path-based exact solution approach for the GVRP.

The method works as follows. Let define Σ as the set containing all the AFSs and the depot 0. The method is made up by two phases: in the first one, all the feasible and not dominated paths connecting two nodes in Σ are generated. Then, in the second phase, some of the paths obtained are selected and combined determining their sequence through a MILP formulation with the objective of minimizing the total traveled distance and respecting the maximum route duration constraint and serving all the customers exactly once.

In fact, each route of the solution for the GVRP can be seen as the combination of paths. A path, identified by number k , with starting point s_k and arrival point a_k handles a subset of customers and s_k as well as a_k can be either an AFS or the depot. It means that we are looking for paths that are either between two stations or a station and the depot or the depot and a station or the depot and itself. Figure 3 shows an example of a route seen as the combination of two paths k_1 and k_2 : k_1 goes from the AFS s_{k_1} to the AFS a_{k_1} while k_2 goes from the AFS a_{k_1} (that is s_{k_2}) to the AFS a_{k_2} . In order to allow that the route starts from the depot and ends to it, we properly add the arc from the depot to s_{k_1} and the arc from a_{k_2} to the depot. Therefore, the total duration of the route is the sum of: $t_{0s_{k_1}}$, the duration t_{k_1} of path k_1 , the duration t_{k_2} of path k_2 , $t_{a_{k_2}0}$ and the service time at each AFS. The duration of each path is computed as the sum of the travel time on each of its arcs plus the service time at each of its customers.

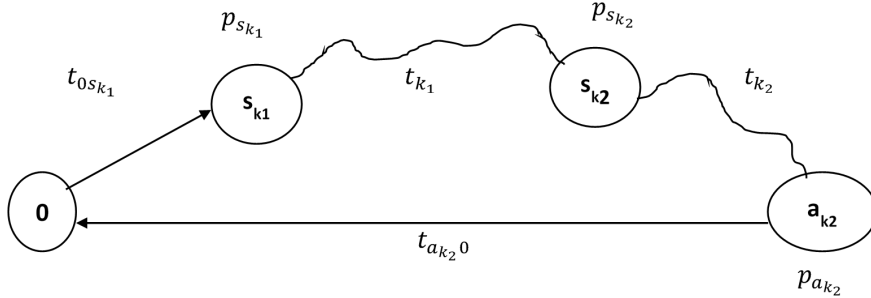


Figure 3: A route example as a combination of two paths

Among all the possible paths, we are looking for all the feasible ones, i.e., the paths that respect both the duration and the fuel level constraints.

Feasibility Rules. Given a path k , with length d_k , starting point s_k , arrival point a_k and its set $I_k \subseteq I$ of customers, its total duration t_k can be computed as in the following:

$t_k = \frac{d_k}{v} + \sum_{i \in I_k} p_i$. Therefore, k is a feasible path if and only if the following two conditions simultaneously hold:

1. $d_k \leq D_{max}$;
2. $t_{0s_k} + t_k + p_{s_k} + p_{a_k} + t_{a_k 0} \leq T_{max}$.

Condition (1) guarantees that an AFV goes from an AFS to another with the sufficient fuel level while condition (2) imposes that the total duration of a path does not exceed the maximum time limit.

Once the set of all the feasible paths has been generated, we remove from it all the paths that are dominated by the others, determining the set K of all the not dominated feasible paths, according to the following rules:

Dominance Rules. Given two feasible paths k_1 and k_2 , k_1 is dominated by k_2 if and only if the following conditions simultaneously hold:

1. $a_{k_1} = a_{k_2}$;
2. $s_{k_1} = s_{k_2}$;
3. $I_{k_1} \subseteq I_{k_2}$;
4. $d_{k_1} > d_{k_2}$.

The first two conditions ensure that the two paths have the same origin and destination; the third condition guarantees that k_2 handles all the customers served in k_1 and finally, the length of k_1 is greater than the length of

k_2 . However, due to both the fuel level and duration constraints, not all the not dominated feasible paths can be combined to each other for composing a route. For this reason, we also define rules for generating a priori the set P of pairs of not dominated feasible paths that are compatible.

Compatibility Rules. Given two paths $k_1, k_2 \in K$, they are compatible to be combined into a pair (k_1, k_2) if and only if the following four conditions simultaneously hold:

1. $s_{k_2} \neq 0 \wedge a_{k_1} \neq 0$;
2. $a_{k_1} = s_{k_2}$;
3. $t_{0s_{k_1}} + p_{s_{k_1}} + t_{k_1} + p_{a_{k_1}} + t_{k_2} + p_{a_{k_2}} + t_{a_{k_2}0} \leq T_{max}$;
4. $I_{k_1} \cap I_{k_2} = \emptyset$.

Figure 3 represents an example of a pair of compatible not dominated feasible paths. Therefore, our solution approach can be summarized as in the following steps:

1. Determine the set of all the feasible paths by invoking, for example, a depth first search algorithm and applying the feasibility rules;
2. Remove from the set of feasible paths, the ones that are dominated by others, by applying the dominance rules: set K ;
3. Find pairs of not dominated feasible paths by applying the compatibility rules: set P ;
4. Solve a set partitioning based MILP model that receives in input both the set K and P .

In the following subsection, the set partitioning based MILP is detailed.

3.1. A set partitioning based MILP

Let define K as the set containing all the feasible paths k , i.e. all the paths that can be covered without refueling and without exceeding the maximum route duration T_{max} . We create a dummy path which starts and ends at the depot, with $t_0 = 0$ and $d_0 = 0$. For each path k , the travel time t_k , the travel distance d_k , the starting site s_k and the arrival site a_k are known. Finally, let define the parameter c_{ik} which is equal to 1 if customer i is served by path k and 0 otherwise. The refueling time p_k is fixed for all paths. A feasible route may be composed by one, two or more paths. We also introduce *elementary* paths which directly connect two distinct nodes in Σ . Such paths may appear more than once in a feasible solution therefore a sufficient number of clones

of them is generated, each one of which can be inserted only once in the solution. The following types of variables are involved in the formulation:

- Z_k : binary variable taking value equal to 1 if path k is selected and 0 otherwise;
- X_{kl} : binary variable taking value equal to 1 if path l is covered just after path k and 0 otherwise;
- T_k : positive variable representing the starting time of path k .

It is worth noting that, in order to reduce the number of variables involved, the X_{kl} is defined only if $a_k = s_l$.

The mathematical model can be written as follows:

$$\min \sum_{k \in K} d_k Z_k \quad (1)$$

s.t.

$$\sum_{k \in K} c_{ik} Z_k = 1 \quad \forall i \in C \quad (2)$$

$$\sum_{k \in K} X_{0k} \leq m \quad (3)$$

$$\sum_{\substack{k_1 \in K: \\ k_1 \neq k_2, (k_1, k_2) \in P}} X_{k_1 k_2} = \sum_{\substack{k_1 \in K: \\ k_1 \neq k_2, (k_2, k_1) \in P}} X_{k_2 k_1} \quad \forall k_2 \in K \quad (4)$$

$$\sum_{\substack{k_1 \in K: \\ k_1 \neq k_2, (k_1, k_2) \in P}} X_{k_1 k_2} = Z_{k_2} \quad \forall k_2 \in K | k_2 \neq 0 \quad (5)$$

$$T_{k_2} \geq T_{k_1} + t_{k_1} X_{k_1 k_2} + p - T_{max} (1 - X_{k_1 k_2}) \quad \forall (k_1, k_2) \in P | k_1 \neq k_2 \quad (6)$$

$$T_k + t_k \leq T_{max} \quad \forall k \in K \quad (7)$$

$$T_k \geq 0 \quad \forall k \in K \quad (8)$$

$$Z_k \in \{0, 1\} \quad \forall k \in K \quad (9)$$

$$X_{k_1 k_2} \in \{0, 1\} \quad \forall (k_1, k_2) \in P | k_1 \neq k_2 \quad (10)$$

The objective function (1) concerns the minimization of the total traveled distance. Constraints (2) imply that each customer is visited exactly once. The number of vehicles used cannot exceed the number of available vehicles, constraint (3). Route continuity is ensured by constraint (4). A path can be inserted in a route, only if it is selected (constraint 5). Starting time for each route is computed by means of constraint (6). Travel time for each route cannot exceed the value T_{max} (constraint 7).

4. Instances description and numerical results

The above proposed approach has been tested on the benchmark instances introduced in [6]. Computational tests have been carried out on a machine with a processor Intel i7-5500U at 2.4 Ghz with 16 Gb of Ram. The path generation phase has been coded in Java while the model has been solved with Xpress 7.9. Instances are grouped in 4 sets with 20 customers each and with different features, as specified in the following:

- **S1**, characterized by a uniform customer distribution. It is made up of 10 randomly generated instances of 20 uniformly distributed customers with 3 AFSs;
- **S2**, characterized by a clustered customer distribution. It is made up of 10 randomly generated instances of 20 clustered customers with 3 AFSs;
- **S3**, aimed to analyze the impact of the spatial AFSs configuration. It has 10 instances, half selected from S1 and half from S2. Each instance has 6 AFSs, randomly generated;
- **S4**, aimed to study the impact of the AFSs density. It has 10 instances, half of which has been created from one instance of S1 and half from one instance of S2, gradually increasing the number AFSs from 2 to 10 with a step of 2.

As suggested in [6], customers which are not reachable from the depot with at most one intermediate refuel stop are discarded.

We compare results obtained by the path based approach (PBA) with those obtained by the exact methods available from the literature that have been tested on the same benchmark instances: the mathematical formulation proposed in [6], (EMH), the branch&cut by [12], (KK) and the five formulations proposed in [14], (LH1),(LH2),(LH3),(LH4) and (LH5).

In Tables, 1,2,3 and 4, we report results related to sets S1, S2, S3 and S4, respectively. All the methods have been run with a timelimit of 3600 seconds. The symbol “-” means that no feasible solution has been found within the time limit. For each approach, we report the best upper bound reached and the computational time elapsed. The EMH model has been reimplemented and run on the same machine of the PBA, because in its

original version it has been run on a machine slower than ours, while for the other methods we have considered results published in the respective papers, which have been obtained on faster machines respect to ours. The second last row reports average values, while the last one reports the number of optimal solutions certified. A summary of both the number of optimal solutions certified and the average computational times, grouped by set, are reported in Table 5. As it can be deduced, PBA is the only method suitable to certify the optimal solutions for all the 40 instances, while both LH1 and LH4 certify 38 optimal solutions. LH2, LH3, LH5 and KK certify 37, 36, 25 and 21, respectively, while EMH always reaches the execution time limit without finding any optimal solution. Moreover, PBA is faster than the other known exact methods, with an average computational time of 358.01 seconds against 429.25 seconds of the second fastest method, i.e., LH4. The dominance of PBA, with respect to the other methods, in terms of both solution quality and computational effort required, is clearly shown by the plot reported in Figure 4.

Table 1: Comparison of methods performances on set S1

INSTANCE	I	F	Total Traveled Distance										Total CPU Time														
			MH	KK	LH1	LH2	LH3	LH4	LH5	PBA	MH	KK	LH1	LH2	LH3	LH4	LH5	PBA									
20c3sU1	20	3	1797.49	1797.49	1797.49	1797.49	1797.49	1797.49	1797.49	1797.49	1797.49	1797.49	1797.49	1797.49	1797.49	1797.49	1797.49	1797.49	3600	172.3	432.57	233.46	252.83	315.1	322.16	31.4	
20c3sU2	20	3	1574.77	1574.77	1574.77	1574.77	1574.77	1574.77	1574.77	1574.77	1574.77	1574.77	1574.77	1574.77	1574.77	1574.77	1574.77	1574.77	3600	3600	277.69	78.5	188.97	121.47	138.69	68.7	
20c3sU3	20	3	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	3600	1789	63.03	33.17	88.27	48.97	44.56	71.2	
20c3sU4	20	3	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	1704.48	3600	3600	520.64	170.37	540	399.82	331.53	21.5	
20c3sU5	20	3	1689.37	1689.37	1689.37	1689.37	1689.37	1689.37	1689.37	1689.37	1689.37	1689.37	1689.37	1689.37	1689.37	1689.37	1689.37	1689.37	3600	2165.5	92.13	124.08	195.1	142.16	308.47	31.6	
20c3sU6	20	3	1618.65	1618.65	1618.65	1618.65	1618.65	1618.65	1618.65	1618.65	1618.65	1618.65	1618.65	1618.65	1618.65	1618.65	1618.65	1618.65	3600	3600	167.4	148.42	215.34	324.81	313	100.4	
20c3sU7	20	3	1713.8	1713.66	1713.66	1713.66	1713.66	1713.66	1713.66	1713.66	1713.66	1713.66	1713.66	1713.66	1713.66	1713.66	1713.66	1713.66	3600	3600	2349.85	3600	2419.88	3600	3600	3600	99.2
20c3sU8	20	3	1709.43	1706.5	1706.5	1706.5	1706.5	1706.5	1706.5	1706.5	1706.5	1706.5	1706.5	1706.5	1706.5	1706.5	1706.5	1706.5	3600	1601.3	39.69	118.86	121.66	60.55	99.6	20.1	
20c3sU9	20	3	1708.81	1708.81	1708.81	1708.81	1708.81	1708.81	1708.81	1708.81	1708.81	1708.81	1708.81	1708.81	1708.81	1708.81	1708.81	1708.81	3600	3600	1380.2	3600	3600	2417.56	3600	91.1	
20c3sU10	20	3	1181.31	1181.31	1181.31	1181.31	1181.31	1181.31	1181.31	1181.31	1181.31	1181.31	1181.31	1181.31	1181.31	1181.31	1181.31	1181.31	3600	2.3	4.88	4.03	6.3	2.26	4.1	134.2	
Average			1617.70	1617.70	1617.70	1617.70	1617.70	1617.70	1617.70	1617.70	1617.70	1617.70	1617.70	1617.70	1617.70	1617.70	1617.70	1617.70	3600	2373.0	532.8	811.09	762.83	743.95	876.21	66.9	
Optima			4	9	7	8	8	8	8	8	8	8	8	8	8	8	8	8	10								

Table 2: Comparison of methods performances on set S2

INSTANCE	I	F	Total Traveled Distance										Total CPU Time													
			MH	KK	LH1	LH2	LH3	LH4	LH5	PBA	MH	KK	LH1	LH2	LH3	LH4	LH5	PBA								
20c3sC1	20	3	1179.17	1173.57	1173.57	1173.57	1173.57	1173.57	1173.57	1173.57	1173.57	1173.57	1173.57	1173.57	1173.57	1173.57	1173.57	1173.57	3600	3600	45.73	23.84	57.82	12.36	43.18	250.33
20c3sC2	19	3	-	1539.97	1539.97	1539.97	1539.97	1539.97	1539.97	1539.97	1539.97	1539.97	1539.97	1539.97	1539.97	1539.97	1539.97	1539.97	3600	1164.5	29.53	38.87	36.54	23.4	91.64	224.03
20c3sC3	12	3	880.2	880.2	880.2	880.2	880.2	880.2	880.2	880.2	880.2	880.2	880.2	880.2	880.2	880.2	880.2	880.2	3600	25.4	2.06	1.75	2.7	1.01	2.43	21.19
20c3sC4	18	3	1109.73	1059.35	1059.35	1059.35	1059.35	1059.35	1059.35	1059.35	1059.35	1059.35	1059.35	1059.35	1059.35	1059.35	1059.35	1059.35	3600	3600	536.01	238.28	676.27	253.87	341.63	319.79
20c3sC5	19	3	2156.01	2156.01	2156.01	2156.01	2156.01	2156.01	2156.01	2156.01	2156.01	2156.01	2156.01	2156.01	2156.01	2156.01	2156.01	2156.01	3600	2246.4	3600	2012.25	3600	1626.2	3600	2035.77
20c3sC6	17	3	2758.17	2758.17	2758.17	2758.17	2758.17	2758.17	2758.17	2758.17	2758.17	2758.17	2758.17	2758.17	2758.17	2758.17	2758.17	2758.17	3600	61.6	19.66	9.56	12.65	15.54	16.23	1.23
20c3sC7	6	3	1393.99	1393.99	1393.99	1393.99	1393.99	1393.99	1393.99	1393.99	1393.99	1393.99	1393.99	1393.99	1393.99	1393.99	1393.99	1393.99	3600	0.1	0.48	0.13	0.25	0.3	0.55	0.06
20c3sC8	18	3	-	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	3600	53.7	12.52	3.7	6.27	6.94	6.24	0.51
20c3sC9	19	3	1813.93	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	1799.94	3600	113.9	47.45	16.36	49.39	31.92	60.22	356.33
20c3sC10	15	3	2583.42	2583.42	2583.42	2583.42	2583.42	2583.42	2583.42	2583.42	2583.42	2583.42	2583.42	2583.42	2583.42	2583.42	2583.42	2583.42	3600	2067.5	26.01	9.33	16.86	16.88	12.81	5.22
Average			1848.43	1848.43	1848.43	1848.43	1848.43	1848.43	1848.43	1848.43	1848.43	1848.43	1848.43	1848.43	1848.43	1848.43	1848.43	1848.43	3600	1293.31	431.94	235.41	445.88	198.84	417.49	321.4
Optima			8	9	9	9	9	9	10	10	9	9	9	10	10	9	10	10								

Table 5: Summary of the results

SET	Number of optima					Total CPU Time								
	KK	LH1	LH2	LH3	LH4	LH5	PBA	KK	LH1	LH2	LH3	LH4	LH5	PBA
1	4	9	7	8	8	7	10	2373.04	532.81	811.09	762.83	743.95	876.21	66.93
2	8	9	10	9	10	9	10	1293.31	431.94	235.41	445.88	198.84	417.49	321.45
3	8	10	10	10	10	10	10	1204.13	282.29	309.62	297.86	260.41	212.35	267.58
4	1	10	10	9	10	9	10	3453.36	640.58	809.19	1039.37	513.78	823.97	776.07
Average	21	38	37	36	38	35	40	2080.96	471.90	541.33	636.48	429.24	582.50	358.01

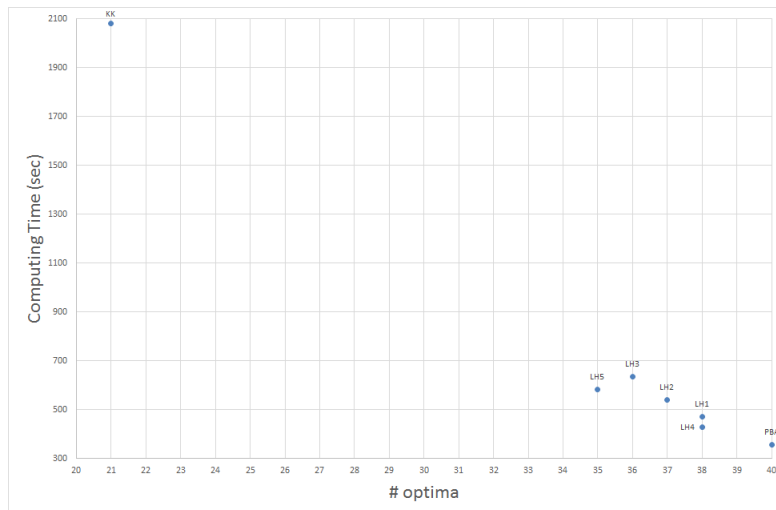


Figure 4: Graphical comparison of methods performances

The layout of the addressed instances, combined with the value of D_{max} and T_{max} , implies that at most one refuel per route is necessary in an optimal solution. Therefore, we have developed a modified version of PBA, named PBA-EMH, in which the paths, between two AFSs, have not been generated, yielding a faster approach in both the two phases thanks to a significantly decrement of the number of not dominated feasible paths. The PBA-EMH can be used only on instances in which the restriction of at most one refueling stop per route holds, while PBA is more general and can be always applied. Table 6 reports a comparison of the number of paths and pairs generated with PBA and PBA-EMH, showing that, exploiting the information on the maximum number of refueling stops, we can a priori exclude, on average, the 60% of the paths and the 47% of the pairs, without any loss of quality, but with a significant gain on computational effort, as shown in Figure 5.

5. Conclusions and future works

In this paper, we presented the Path Based Approach (PBA), a new exact method to efficiently solve the Green Vehicle Routing Problem (GVRP), based on the generation and combination of not dominated paths, each of them serving a subset of customers without intermediate refuels.

PBA was tested on four sets of benchmark instances available in the literature. The numerical results found were compared with those obtained by all

Table 6: Number of paths and pairs generated by the PBA and the PBA-EMH

INSTANCE	I	F	PBA		PBA.EMH	
			Paths	Pairs	Paths	Pairs
20c3sU1	20	3	9232	44501	3201	10907
20c3sU2	20	3	17019	66092	5400	15774
20c3sU3	20	3	17014	70092	5456	17030
20c3sU4	20	3	8951	38584	3036	9301
20c3sU5	20	3	11388	44813	3742	10579
20c3sU6	20	3	20283	96361	6451	18959
20c3sU7	20	3	20123	95788	6328	18405
20c3sU8	20	3	8673	37931	2984	9179
20c3sU9	20	3	18426	38247	5239	11875
20c3sU10	20	3	25771	90746	8200	23973
Average			17017	55453	5320	13825
20c3sC1	20	3	32972	154210	11008	38153
20c3sC2	19	3	29231	115700	8844	25635
20c3sC3	12	3	10772	35450	3738	11635
20c3sC4	18	3	30869	116394	9431	35672
20c3sC5	19	3	16636	28841	5614	12299
20c3sC6	17	3	1708	7179	642	1805
20c3sC7	6	3	113	306	46	89
20c3sC8	18	3	740	2781	308	801
20c3sC9	19	3	42836	186596	12704	36802
20c3sC10	15	3	845	3438	338	917
Average			13704	32146	4676	11967
S1-2i6s	20	6	29298	213590	11142	45480
S1-4i6s	20	6	23937	172640	9026	28650
S1-6i6s	20	6	20712	103474	6651	20764
S1-8i6s	20	6	13145	80370	5062	18393
S1-10i6s	20	6	58045	356176	22000	70497
S2-2i6s	20	6	69500	456200	18024	57026
S2-4i6s	19	6	67556	366766	26875	87456
S2-6i6s	20	6	4918	36757	1642	4566
S2-8i6s	16	6	1787	10576	610	1508
S2-10i6s	16	6	4423	27665	1672	4322
Average			22325	138057	7839	24707
S1-4i2s	20	2	8907	35998	3004	8594
S1-4i4s	20	4	9959	43969	3528	10834
S1-4i6s	20	6	23937	172640	9026	28650
S1-4i8s	20	8	37796	349736	12690	40555
S1-4i10s	20	10	41401	406457	13806	45458
S2-4i2s	18	2	30076	111105	9039	34241
S2-4i4s	19	4	46855	200142	17685	58651
S2-4i6s	20	6	67556	366766	26875	87456
S2-4i8s	20	8	90810	448072	40557	139262
S2-4i10s	20	10	89956	375867	59941	185366
Average			39599	274939	13248	43007

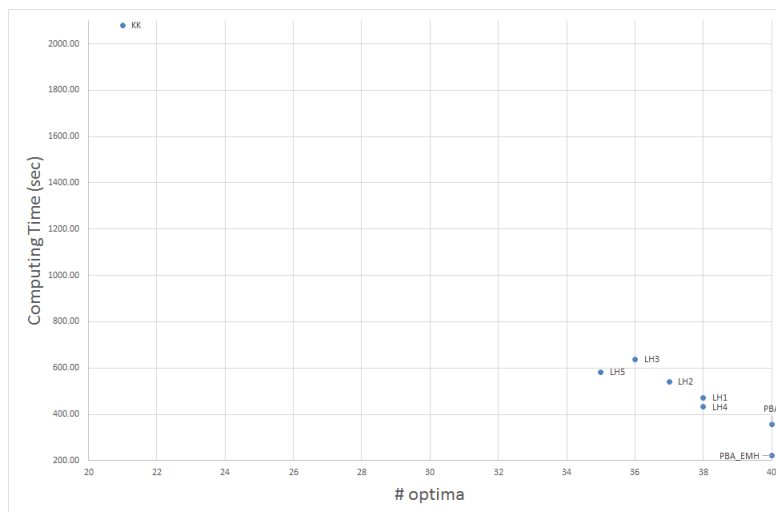


Figure 5: Graphical comparison of methods performances including PBA-EMH

the existing exact methods, tested on the same benchmarks. PBA was shown to outperform all the existing exact approaches in terms of both efficiency and effectiveness, solving to the optimality all the instances in a smaller computational time (on average, 358.01 seconds, against 429.25 seconds of the previous fastest method in the literature). Moreover, PBA revealed to be the only method suitable to certify the optimality of the solution found for all the tested instances.

Future research directions can go toward a generalization of PBA in order to solve other Vehicle Routing Problems (VRPs) with Intermediate Stops, (VRPIS), as the VRP with Cross Docking, the Waste Collection problem with intermediate disposals, and the Multi-Depot VRP with intra-depots routes, to cite a few. Moreover, PBA can be applied also to extended versions of the GVRP, e.g., with multiple recharging technologies, partial recharges and capacitated AFSs. Finally, PBA may be an approach suitable also to solve problems in which the number of customers per route is limited, i.e., the Drayage Problems.

References

- [1] J. Andelmin and E. Bartolini. An exact algorithm for the green vehicle routing problem. *Transportation Science*, 2017.

- [2] M. Bruglieri, S. Mancini, F. Pezzella, and O. Pisacane. The green vehicle routing problem with capacitated alternative fuel stations. In *Verolog 2017, Amsterdam, July 10th-2th*, 2017.
- [3] M. Bruglieri, S. Mancini, F. Pezzella, O. Pisacane, and S. Suraci. A three-phase matheuristic for the time-effective electric vehicle routing problem with partial recharges. *Electronic Notes in Discrete Mathematics*, 58:95–102, 2017.
- [4] M. Bruglieri, F. Pezzella, O. Pisacane, and S. Suraci. A variable neighborhood search branching for the electric vehicle routing problem with time windows. *Electronic Notes in Discrete Mathematics*, 47:221–228, 2015.
- [5] C. Cheng, P. Yang, M. Qi, and L.M. Rousseau. Modeling a green inventory routing problem with a heterogeneous fleet. *Transportation Research Part E: Logistics and Transportation Review*, 97:97–112, 2017.
- [6] S. Erdoğan and E. Miller-Hooks. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114, 2012.
- [7] Á. Felipe, M.T. Ortuño, G. Righini, and G. Tirado. 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, 2014.
- [8] A. Froger, J. Mendoza, O. Jabali, and G. Laporte. A matheuristic for the electric vehicle routing problem with capacitated charging stations. Technical Report CIRRELT-2017-31, CIRRELT, 2017.
- [9] A. Froger, J. Mendoza, O. Jabali, and G. Laporte. New formulations for the electric vehicle routing problem with nonlinear charging functions. Technical Report CIRRELT-2017-30, CIRRELT, 2017.
- [10] L. Grandinetti, F. Guerriero, F. Pezzella, and O. Pisacane. A pick-up and delivery problem with time windows by electric vehicles. *International Journal of Productivity and Quality Management*, 18(2-3):403–423, 2016.

- [11] M. Keskin and B. Çatay. Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 65:111–127, 2016.
- [12] Ç. Koç and I. Karaoglan. The green vehicle routing problem: A heuristic based exact solution approach. *Applied Soft Computing*, 39:154–164, 2016.
- [13] G. Laporte and M.M.B. Pascoal. Path based algorithms for metro network design. *Computers & Operations Research*, 62:78–94, 2015.
- [14] V. Leggieri and M. Haouari. A practical solution approach for the green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 104:97–112, 2017.
- [15] M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.
- [16] S. Madankumar and C. Rajendran. Mathematical models for green vehicle routing problems with pickup and delivery: A case of semiconductor supply chain. *Computers & Operations Research*, 2016.
- [17] S. Mancini. The hybrid vehicle routing problem. *Transportation Research Part C: Emerging Technologies*, 78:1–12, 2017.
- [18] D. Margaritis, A. Anagnostopoulou, A. Tromaras, and M. Boile. Electric commercial vehicles: Practical perspectives and future research directions. *Research in Transportation Business & Management*, 18:4–10, 2016.
- [19] A. Montoya, C. Guéret, J.E. Mendoza, and J. G. Villegas. A multi-space sampling heuristic for the green vehicle routing problem. *Transportation Research Part C: Emerging Technologies*, 70:113–128, 2016.
- [20] M. Schneider, A. Stenger, and D. Goeke. The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520, 2014.
- [21] M. Schneider, A. Stenger, and J. Hof. An adaptive vns algorithm for vehicle routing problems with intermediate stops. *OR Spectrum*, 37(2):353–387, 2015.

- [22] M. Soysal and M. Çimen. A simulation based restricted dynamic programming approach for the green time dependent vehicle routing problem. *Computers & Operations Research*, 2017.
- [23] P. Toth and D. Vigo. *Vehicle routing: problems, methods, and applications*. SIAM, 2014.
- [24] M. Wen, E. Linde, S. Ropke, P. Mirchandani, and A. Larsen. An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem. *Computers & Operations Research*, 76:73–83, 2016.
- [25] P.Y. Yin and Y.L. Chuang. Adaptive memory artificial bee colony algorithm for green vehicle routing with cross-docking. *Applied Mathematical Modelling*, 40(21):9302–9315, 2016.