

Article

SVND Enhanced Metaheuristic for Plug-In Hybrid Electric Vehicle Routing Problem

Xiaohui Li * , Xuemin Shi, Yi Zhao , Huagang Liang and Yuan Dong

School of electronics and control engineering, Chang'An University, Xi'an, 710064, China;

xueminshichd@gmail.com (X.S.); z1@chd.edu.cn (Y.Z.); hgliang@chd.edu.cn (H.L.); yuandong@chd.edu.cn (Y.D.)

* Correspondence: xiaohui.li@chd.edu.cn; Tel.: +86-29-82334640

Received: 11 December 2019; Accepted: 4 January 2020 ; Published: 7 January 2020



Abstract: Plug-in Hybrid Electric Vehicles (PHEVs), as a new type of environmental-friendly low cost transportation, have attracted growing interests for logistics. The path-planning optimization for PHEV has become a major challenge. In fact, PHEV-based routing optimization is a type of hybrid vehicle routing problem (HVRP). Compared with the traditional Traveling Salesman Problem (TSP) and Vehicle Routing Problem (VRP), the PHEV routing problem should consider more constraints, such as time limits, capacity constraints (including fuel tank capacity and battery capacity), electric stations, fuel stations and so forth. In this paper, a Mixed Integer Linear Programming formulation is presented and a novel hybrid metaheuristic approach (HMA_SVND) is proposed. Our method is a combination of memetic algorithm (MA), sequential variable neighborhood descent (SVND) and a revised 2_opt method. Comparative studies show that our proposed method outperformed previous works.

Keywords: hybrid vehicle routing problem; memetic algorithm; sequential variable neighborhood descent; plug-in hybrid electric vehicle

1. Introduction

Transportation plays an important role in the logistics system. Actually, transportation for logistic mainly contained two phases: sending raw materials from suppliers to manufacturers and the delivery of finished products from manufacturers to customers [1]. However, current transportation mostly employs fuel powered vehicles, which emit a lot of greenhouse gases (GHG) and harmful substances, such as carbon dioxide and sulfide, causing various environmental problems. Therefore, energy saving and emission reduction are more than necessary for logistic transportation. Recently, the conception of green logistics has been proposed. Green logistics is more environment-friendly. It has quickly become a hot research subject. By optimizing the usage of logistic resources, it focuses on restraining the environmental harm while reducing the whole logistic cost.

In order to build green logistics, the first objective is to make a reasonable layout and path planning. This is usually regarded as a Vehicle Routing Problem (VRP). The VRP is a well-known NP-hard combinatorial optimization problem. It was first introduced in 1959 [2]. The VRP problem is different from traditional routing problems such as Traveling Salesman Problem (TSP), it can be more complicated by taking consideration of multiple vehicles and various constraints. The classical VRP problem can be summarized as follows [3]—there is a certain number of customers with different cargo needs. Distribution centers provide cargo to customers. A fleet is responsible for distributing cargo and organizing appropriate driving routes. The aim is to satisfy the customer's needs and to achieve goals like minimizing the travel distance, reducing costs or optimizing travel time under certain constraints. Lin et al. [4] provided a review study on VRP and classified them into two categories—traditional VRP and Green Vehicle Routing Problem (GVRP) which is firstly proposed by Erdoğan and Miller-Hooks [5].

Compared to VRP, GVRP considers environmental factors in the whole route planning. GVRP aims to achieve the balance of economic and environmental interests.

The abbreviation of HVRP used in this work is derived from Mancini's [6] article, in which the Hybrid Vehicle Routing Problem is defined as an extension of the classical VRP where vehicles can work both electrically and with traditional fuel. Lin et al. [4] classified HVRP as a special case of GVRP.

Green vehicles or environment-friendly vehicles are also known as new energy vehicles (NEVs), of which, Battery Electric Vehicle (BEVs) is considered one of the most promising solutions to zero-emission transportation. Compared to traditional fuel powered vehicles, BEVs use renewable sources, they are not only much more environment-friendly, but also more energy efficient [6]. However, due to the limitation of the battery capacity, they require frequent visits to recharging stations along the route. At present, the major issue for BEV is that the number of charging stations in urban areas is quite limited globally. In addition, due to the physical characteristics of electric batteries, pure electric vehicles now have shorter transport distances in cold regions. Therefore, BEVs are ideal for last mile deliveries rather than long distance transportation. Hybrid electric vehicles (HEVs), on the contrary, are powered by both fuel and electricity [7]. It can be more practical for long distance logistics. A Plug-in Hybrid Electric Vehicle (PHEV) is the latest generation of HEVs. The PHEV used in this paper is not for inter-city logistics distribution, but for intra-city logistics distribution. PHEVs have been gradually applied in intra-city logistics in many countries. In terms of logistics, PHEVs can be more economical and environment-friendly than traditional fuel powered vehicles [8]. Furthermore, it can serve a much longer distance compared to pure electronic logistic vehicles. The difference between PHEV and ordinary HEVs is that the battery capacity of ordinary HEVs is very limited and its battery cannot be charged externally. The HEV's driving distance in pure electric mode is very short; on the contrary, a PHEV can be charged at any available electric power stations. It can operate in pure electric mode (zero-emission). After battery power is exhausted, it can automatically shift into hybrid mode (mainly fuel power). Therefore, PHEVs combine the advantages of fuel powered vehicles and pure electric vehicles. Hiermann et al. [9] studied an electric vehicle routing problem combining conventional, plug-in hybrid, and electric vehicles. To solve this problem, they have designed a sophisticated metaheuristic which combines a genetic algorithm with local and large neighborhood search. Björnsson and Karlsson [10] gave detailed explanations of the advantage of PHEV and BEV under different scenarios. Silva et al. [11] evaluated energy consumption, emissions and costs of PHEV.

Path planning for a plug-in hybrid vehicle is a typical Hybrid Vehicle Routing Problem (HVRP). Lin et al. [4] classified HVRP as a special case of GVRP. The traditional vehicle routing problem does not consider the access to the fuel stations, while the electric vehicle routing problem (EVRP) considers the access to the electric stations. The HVRP should further consider the access to both electric stations and fuel stations. In addition, HVRP takes into account the environmental impacts. The method and model in this paper are based on some assumptions—Firstly, the vehicle's speed keeps constant in the course of route driving. Secondly, PHEV has two energy sources, fuel and electricity, electricity has a high priority, and when the maximum driving capacity of the battery is exhausted, the vehicle will switch to the fuel mode. In this work, time limits, battery capacitance and fuel capacity are also the major considered constraints. The purpose is to consider the use of electricity and fuel reasonably according to the availability of electric or fuel stations while minimizing the total travel cost. A good PHEV routing solution can help enterprises or organizations reducing the overall travel cost by providing the optimal/near-optimal path planning within a certain time limit.

Contribution and innovation of this study are highlighted as follows:

- Compared with the existing HVRP studies on PHEV routing problem (Yu et al. [8]), this work deals with a complicated PHEV routing problem in which the types of charging power stations (fuel station or electric station) are provided to the vehicle. The vehicle can be refueled or charged during its delivery.

- The PHEV routing problem is formulated as a Mixed Integer Linear Programming model, different from previous research, various hybrid vehicles are considered.
- This study proposes a new hybrid metaheuristic which combines the memetic algorithm with a powerful local search technique SVND; the experimental results show that our algorithm obtains better results than previous ones.

The rest of this article is organized as follows. Section 2 reviews the relevant literature. Section 3 defines the problem. Section 4 describes the proposed memetic algorithm and SVND neighborhood search method. Section 5 reports the experimental results and the conclusion is given.

2. Literature Review

The Vehicle Routing Problem (VRP) was first introduced in 1959. Since then, many variants of the VRP problem have been studied. Recently, the environmental impacts of logistics transportation have grabbed more attention. Green logistics has therefore become a hot research topic. It was first proposed by Dekker et al. [12] and Sbihi et al. [13], and it takes environmental problems into combinatorial optimization problems. Indeed, green logistics has attracted increasing research attention. For example, the GVRP was proposed by Erdoğan and Miller Hooks et al. [5], and focuses on reducing emissions and eliminating environmental impacts. In their work, they considered the limited fuel capacity of vehicles and the possibility of refueling at fuel stations along the way. In recent years, there have been many studies on the variants of GVRP and there are some studies combining carbon emissions or energy consumption with different constraints and optimization objectives. For example, Messaoud et al. [14] proposed the Dynamic Green Vehicle Routing Problem (DGVRP), which combines the Dynamic Vehicle Routing Problem (DVRP) with GVRP. Leu et al. [15] took into account the distribution efficiency and carbon emissions and developed a cargo flow modeling method to analyze logistics planning. Ubeda et al. [16] conducted a study aimed at minimization of both traveling distances and pollutant emissions. For GVRP, there are two kinds of solutions based on exact and heuristic algorithms. Zhou et al. [17] proposed a Lagrangian Relaxation-Based Solution for GVRP. Zhang et al. [18] proposed a new tabu search named the RS-TS algorithm to solve the Vehicle Routing Problem with fuel consumption and carbon emissions. Demir et al. [19] used an adaptive large neighborhood search algorithm (ALNS) to minimize driving time and fuel consumption, respectively. Lately, many works have focused on the path planning problem of new energy vehicles. The path planning problem for electric vehicles is called the electric vehicle routing problem (EVRP). Schiffer et al. [20] proposed a location routing method for charging stations, which considered both the path of electric vehicles and the location of charging stations. On the basis of EVRP, Shao et al. [21] considered the impact of an electric vehicle's driving speed and cargo load on energy consumption. They adopted a hybrid genetic algorithm to solve the problem. For EVRP, latter works also considered time window constraints. Roberti and Wen [22] investigated the Electric Traveling Salesman Problem with Time Windows. Keskin and Çatay [23] discussed the electric vehicle routing problem with time-windows and partial recharging (EVRP-TWPR). In recent works on EVRP, the objectives have been to reduce energy consumption, cost, distance and so on. In terms of constraints, the layout of charging stations, partial charging, time window and variable speed are considered. For example, Mavrouniotis et al. [24] applied the ant colony algorithm (ACO) for EVRP which estimated whether the electric vehicle has a charging station in its working range. Schneider et al. [25] introduced a hybrid heuristic for EVRP with time windows and recharging stations. Hannan et al. [26] provided a comprehensive survey of HEVs' source combination, models, energy management systems and so forth. Wen et al. [27] discussed the plug-in electric vehicles (PEVs) charging selection problem.

Plug-in hybrid vehicles are regarded as one of the most environment-friendly solutions for logistic transportation. Actually, there are more related works that try to provide a solution for HVRP. The vehicle routing problem for new hybrid energy vehicles is termed HVRP, which was proposed by Mancini [6]. The author defined HVRP as an extension of GVRP, in which vehicles can use both electricity and fuel, the unitary travel cost is much lower for distances covered in electric mode, a

mixed integer linear programming formulation and a large neighborhood search based on metaheuristic are proposed. Yu et al. [8] proposed a simulated annealing (SA) algorithm based on restart strategy for HVRP. Hiermann et al. [9] presented a complex meta-heuristic method, which combines a genetic algorithm with a local and large-scale neighborhood search. Macrina et al. [28] proposed an iterative local search heuristic algorithm to optimize the routing of hybrid vehicles with electric engines and internal combustion engines. Sun and Zhou [29] proposed a Cost Optimization Algorithm (COA) based on the fact that plug-in hybrid electric vehicles have two power sources, so even along the same route, different power choices will lead to different energy consumption. In Tarek Abdallah's thesis [30], the vehicle routing problem for PHEV with time window constraint is studied. The objective of optimization is to minimize the fuel powered driving time within the time window. A Lagrange relaxation method is used to solve this problem, also a new tabu search algorithm is proposed. To our knowledge, Tarek Abdallah was the first to put forward the vehicle routing problem for PHEV.

Generally, HVRP is a combinatorial optimization problem. The solutions can be divided into three categories—neighborhood-centered, population-based and hybrid methods [31]. The first type of approach is generally proceeded by iteratively exploring neighborhoods of a single incumbent solution. Such as Simulated Annealing (SA) [8], Adaptive Large Neighborhood Search (ALNS) [32], Large Neighborhood Search (LNS) [5]. The second type is usually based on natural laws, such as Genetic Algorithms (GA), Evolutionary Algorithms (EA) and Memetic Algorithm (MA) [33]. The third combines the previous methods, it is a hybridization of the metaheuristic and a powerful local search technique. Vidal (2012) et al. [34] proposed a new approach based on a hybrid genetic algorithm for Multidepot and periodic vehicle routing problem. The algorithm combined population evolutionary search, meta-heuristic neighborhood search and adjusted population diversity management to improve the solution's general performance.

As a matter of fact, there are many studies about the vehicle routing problem of electric vehicles, but few articles about the vehicle routing problem of PHEV. To some extent, this paper presents a Mixed Integer Linear Programming model and provides a new algorithm for the vehicle routing problem of PHEV. In our previous work [35], we employed a memetic algorithm for solving the PHEV based HVRP problem by considering both electric and fuel stations. In order to obtain a better solution for a large-scale HVRP with PHEV, we have made some improvements on the previous algorithm. We proposed a novel hybrid memetic algorithm. It mainly combines memetic algorithm with sequential variable neighborhood descent (HMA_SVND). We tested our new approach on larger test instances and detailed comparative studies on open source benchmarks are performed.

3. Problem Description and Mathematical Formulation

Inspired by the works of Yu et al. [8], Erdoğan [5] and Mancini [6], we describe the problem as follows—the PHEV routing problem can be defined as an undirected complete graph $G = (V, E)$, where vertex set V is a combination of the customer set $I = \{v_1, v_2, \dots, v_n\}$, the depot v_0 , a set of fuel stations (S_f) and a set of electric stations (S_e). The fuel station or electric station can be visited several times by the vehicle. Each node j is characterized by a service time p_j . Each edge (v_i, v_j) is associated with a non-negative travel time t_{ij} and distance d_{ij} . This problem seeks to find at most m tours, visiting a subset of vertices including S_e and S_f when needed such that the total transportation cost is minimized. The studied problem considers two kinds of constraints: time constraint and capacity constraint (including fuel tank capacity and electric capacity). The time constraint means that the duration of the route in a feasible solution should not exceed the T_{max} . Admittedly, in certain cases, there is a possibility that all customers are visited by one vehicle. The following assumption is considered in this work.

1. All customers need to be visited and each customer can only be visited once.
2. There are at most m vehicles available and each vehicle must start and finish in the depot.
3. Vehicles can enter S_e or S_f when needed. The station type $Z(S_z)$ can be visited more than once or not visited at all.

4. There are no restrictions on the number of vehicles accessing electric or fuel stations.
5. When the vehicle visits the power station, the corresponding fuel or electric energy will be filled up.
6. Once the vehicle returns to the depot, both its fuel and battery will be filled to its maximum.
7. Each route may be covered using only the electric engine, or a part of it can be covered using traditional fuel propulsion, we give energy priority to electricity power.
8. Travel speeds are assumed to be constant over a link.
9. The travel time of each vehicle used in this problem should not exceed the predetermined maximum time T_{max} .

The mathematical formulation of PHEV routing problem is as follows:

Objective function:

$$Min \sum_{\substack{i,j \in V_0 \\ i \neq j}} \sum_{z \in Z} \sum_{k \in M} f_{ijkz} c_z. \tag{1}$$

Subject to:

$$\sum_{\substack{j \in V_0 \\ j \neq i}} \sum_{k \in M} x_{ijk} = 1 \quad \forall i \in I \tag{2}$$

$$\sum_{\substack{j \in V_0 \\ j \neq i}} \sum_{k \in M} x_{ijk} \leq 1 \quad \forall i \in S_0 \tag{3}$$

$$\sum_{\substack{j \in V_0 \\ j \neq i}} \sum_{k \in M} x_{jik} = \sum_{\substack{j \in V_0 \\ j \neq i}} \sum_{k \in M} x_{ijk} \quad \forall i \in V \tag{4}$$

$$\sum_{j \in V} \sum_{k \in M} x_{0jk} \leq m \tag{5}$$

$$\sum_{j \in V} \sum_{k \in M} x_{j0k} \leq m \tag{6}$$

$$\tau_j \geq \tau_i + (t_{ij} + p_i) \sum_{k \in M} x_{ijk} - T_{max}(1 - \sum_{k \in M} x_{ijk}) \quad \forall i, j \in V_0, \text{ and } i \neq j \tag{7}$$

$$0 \leq \tau_0 \leq T_{max} \tag{8}$$

$$t_{0j} \leq \tau_j \leq T_{max} - (t_{j0} + p_j) \quad \forall j \in V \tag{9}$$

$$y_{jkz} \leq y_{ikz} - f_{ijkz} + C(1 - X_{ijk}) \quad \forall i, j \in V_0, \forall k \in M, \forall z \in Z, \text{ and } i \neq j \tag{10}$$

$$f_{ijkz} \leq Q_{kz} \quad \forall i, j \in V_0, \forall k \in M, \forall z \in Z, \text{ and } i \neq j \tag{11}$$

$$f_{ijkz} \leq y_{ikz} \quad \forall i, j \in V_0, \forall k \in M, \forall z \in Z, \text{ and } i \neq j \tag{12}$$

$$\sum_{z \in Z} \frac{f_{ijkz}}{r_{kz}} = d_{ij} x_{ijk} \quad \forall i, j \in V_0, \forall k \in M \tag{13}$$

$$y_{jkz} = Q_{kz} \quad \forall i, j \in V_0, \forall k \in M, \forall z \in Z \tag{14}$$

$$x_{ijk} \in \{0, 1\} \quad \forall i, j \in V_0, \forall k \in M \tag{15}$$

$$y_{jkz} \geq 0 \quad \forall j \in V_0, \forall k \in M, \forall z \in Z \tag{16}$$

$$\tau_j \geq 0 \quad \forall j \in V_0 \tag{17}$$

$$f_{ijkz} \geq 0 \quad \forall i, j \in V_0, \forall k \in M, \forall z \in Z. \tag{18}$$

Objective (1) is to minimize the total cost of travel using PHEVs in a given day. Constraint (2) and (3) ensure that each customer is visited exactly once by one vehicle and each station may be visited at

most once, respectively. Route continuity is guaranteed by constraint (4) while a maximum number of routes is imposed by constraints (5) and (6).

The arrival time at each vertex by one vehicle is tracked through constraint (7). Constraint (8) ensures that each vehicle returns to the depot before the time limit T_{max} . Constraint (9) implies the lower and upper bound on arrival time at customer and station vertices.

Constraint (10) tracks a vehicle's energy level based on vertex sequence and type. If vertex j is visited right after vertex i by vehicle k ($X_{ijk} = 1$), then the vehicle's remaining energy level should minus the fuel and electric consumed on this route. Constraints (11) and (12) implies the upper bound of energy consumption of type z from vertex i to vertex j by vehicle k . Constraint (13) implies that vehicle k 's energy consumption from vertex i to vertex j is based on the distance and vehicle k 's energy consumption rate. Constraint (14) resets the power level of type z to Q_{kz} after visiting the depot or the station vertex of type z . Finally, the domain of the variables are specified by the constraints (15) to (18).

4. Methodology

A memetic algorithm is a hybridization of a genetic algorithm with a local search, so this work proposes a memetic algorithm using an SVND.

4.1. Memetic Algorithm

Memetic Algorithm (MA), a population-based metaheuristic, is an improvement of the evolutionary algorithm (EA). It is defined by the collaboration between cultural evolution and natural evolution to explore and exploit the search [36]. The concept of MA was inspired by Dawkin's notion of memes (a unit of cultural evolution), which consists of self-improvement [37]. The classical memetic algorithm is a combination of EA and local search procedure (LSP). The pseudo code of standard MA is given in Algorithm 1.

Firstly, an initial population is generated randomly and a local search procedure is applied to optimize it. Then, the fitness values of all the individuals are calculated by using an evaluation process. Afterwards, during the whole loop, the selection, crossover and mutate operators are employed to generate new offsprings. These offsprings are further optimized through the local search procedure, then these solutions are evaluated through the evaluation process. After that, the SelectNewPop operator tries to choose the best individuals for the population of the next generation. These steps are repeated until the stopping criterion is satisfied.

Algorithm 1: Structure of the Memetic Algorithm

Input: Data of problem and parameters of algorithm

Output: The best individual

Begin

$t \leftarrow 0$

$P(t) \leftarrow \text{InitPop}()$

$P(t) \leftarrow \text{LocalSearch}(P(t))$

Evaluate Fitness($P(t)$)

While *stopping criterion is not satisfied* **DO**

$P'(t) \leftarrow \text{Selection}(P(t))$

$P'(t) \leftarrow \text{Crossover}(P'(t))$

$P'(t) \leftarrow \text{Mutate}(P'(t))$

$P'(t) \leftarrow \text{LocalSearch}(P'(t))$

Evaluate Fitness($P'(t)$)

$P(t+1) \leftarrow \text{SelectNewPop}(P(t), P'(t))$

$t \leftarrow t+1$

End While

End

4.2. Representation and Evaluation

The solution consists of n customers and $q (q \geq 0)$ selected fuel or electric stations, where the power stations come from sets S_e and S_f . Each customer must be visited and can only be visited once, but each electric station and fuel station may be visited more than once or not at all. Therefore, the number of visited power stations in solutions may be different. Index 0 represents the depot, the i^{th} number in $1, 2, \dots, n$ denotes the i^{th} customer to be served. The following number of $n + 1, n + 2, \dots, n + m$ then denotes the index of power stations. Figure 1 shows two feasible solutions.

An example in Figure 1 shows the representation for an instance where 15 customers need to be visited and 4 power stations were considered, indexes 1 to 15 represent the customers, 16 and 17 electric stations, 18 and 19 fuel stations. As can be seen from Figure 1a,b, we adopt a chromosome coding method in which -1 is used as the delimiter between routes. In Figure 1a only 15 customers were visited without access to the energy station and 4 vehicles were used to visit all customers. In Figure 1b, all customers were visited by 3 vehicles, and the second vehicle and the third vehicle visit a electric station and a fuel station in their tour, respectively. Comparing two representations, the latter uses fewer vehicles because there are vehicles visiting the power station on the way, their driving distance capacity was extended. Of course, their driving time still should not exceed T_{max} .

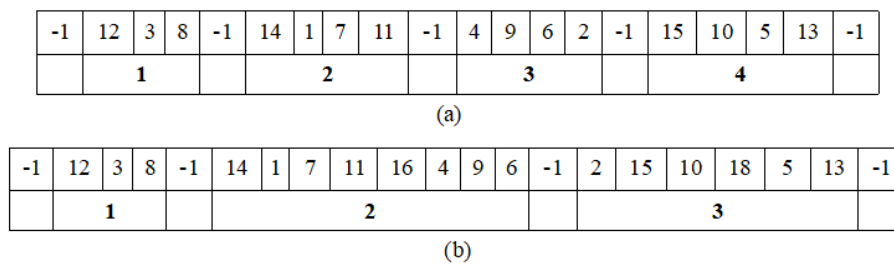


Figure 1. Example of the proposed representation. (a) is a feasible solution without accessing power stations. (b) is a feasible solution accessing power stations.

As we considered more constraints than GVRP [5] and HVRP [6] in this work, there would be more infeasible solutions during the computation of the objective function; however, the elimination of all infeasible solutions would harm the diversity of the population and the solution would highly likely fall into local optimum. Therefore, we introduced a penalty mechanism to maintain the diversity of population. The evaluation of the solution can be described as follows—the delimiter is a sign to distinguish different routes. For each route, we need to determine whether time constraint and vehicle energy capacity constraint are satisfied. We set the time constraint as a soft time constraint, and the solution beyond T_{max} is also acceptable. Only a certain penalty should be added to reduce the probability of the solution selected, the objective function of the solution with penalty P will become worse and be eliminated in the process of iterating to produce the new solution. The reason for introducing penalty P is to ensure the diversity of population. However, when the residual electric and fuel energy of the vehicle are not enough to support its return to the depot or to the nearest energy station, the solution is not feasible. The feasibility checking of the solution refers to the checking of the vehicle energy capacity constraint, which is a constraint condition. If the constraint is violated, the solution is not feasible. We added an infinite number to the objective function to cancel the probability of the solution being selected. The pseudo-code for evaluation procedure is shown in Algorithm 2, where the symbols $vn, EPCost_i, FPCost_i, time_spent_i$ represent the number of the route, the cost of electric, the cost of fuel and the time spent in route i , respectively.

Algorithm 2: Evaluation of the individual

Input: An individual
Output: The value of objective function
Begin
 Cost ← 0
for $i = 1$ to vn **do**
 Calculate $EPCost_i, FPCost_i$ of route i
 $Cost \leftarrow Cost + c_e \times EPCost_i + c_f \times FPCost_i$
 Calculate $time_spent_i$
 If $time_spent_i > T_{max}$ **then**
 Cost ← Cost + P
 End
 Solution-feasibility checking
 If unfeasible **then**
 Cost ← Cost + A large Penally
 End
End
 return Cost
End

4.3. Individual Initialization

Generally, the initial solutions are generated randomly but the convergence of the solution can be accelerated by using a heuristic. In this work, in order to speed up the convergence, we generate an initial population including a heuristic solutions and random solutions to explore the different regions of solution space. A single individual can be generated by a novel approach which is revised from Nearest Neighborhood Heuristic (NNH) [38] and the other individuals are generated randomly. The detailed procedure of this method is shown in Algorithm 3.

4.4. Select Individual for the Recombine Operator and The Mutate Operator

Inspired by Singh and Banghel [39], Singh and Gupta [40], to avoid premature convergence of programs, we have used a binary tournament individual selection based on the probability p_{better} . The candidate with better fitness is selected with this probability. It is a pre-set value, and this value is suggested to 0.8 [41]. The binary tournament selection takes two individuals randomly from the population, it means that in each selection process, we first generate a random number. If this random number is smaller than the p_{better} , we choose the better one in the two individuals, otherwise we choose the worst one. This process will be repeated twice to select two different individual as the parents for the recombine operator and the mutate operator. The pseudo code of the binary tournament selection is given in Algorithm 4.

Algorithm 3: Method to generate a signal individual

```

Input: The data of problem
Output: An initial solution  $R$ 
Begin
   $i = 0, count = 0, R = NULL$ 
   $vn = 1, route_{vn} = NULL$ 
  While  $count \leq n$  DO
     $j =$ the nearest unassigned node from  $i$ 
    If  $i = 0$  then
       $time\_spent = T_{ij} + T_{ji}$ 
    Else
       $time\_spent = time\_spent - T_{i0} + T_{ij} + T_{j0}$ 
    End
    If  $time\_spent \leq T_{max}$  then
       $route_{vn} = route_{vn} + j$ 
       $i = j$ 
       $count ++$ 
    Else
       $i = 0, vn ++$ 
       $route_{vn} = NULL$ 
    End
  End While
  for  $i = 1$  to  $mR$  do
    Calculate  $EP_i$  and  $FP_i$  of route  $i$ 
    If  $EP_i$  and  $FP_i >$ vehicle capacity then
       $route_{vn} = route_{vn} + S_e$  or  $S_f$ 
       $R = R + route_{vn}$ 
    End
  End
  return  $R$ 
End

```

Algorithm 4: The pseudo-code of binary tournament selection

```

Input: Individual population
Output: new population  $X$ 
Begin
  Select two employed solution  $S_1$  and  $S_2$  from the population
   $r = random()$ 
  If  $r < p_{better}$  then
     $X \leftarrow$ the best of  $S_1$  and  $S_2$ 
  Else
     $X \leftarrow$ the worst of  $S_1$  and  $S_2$ 
  End
  return  $X$ 
End

```

4.5. Recombine Operator

Before the recombine operator, we obtained two parent individuals by using binary tournament selection method. In this paper we attempt to exchange memes between two parents to obtain new individuals. The recombine operator was proposed firstly by Singh and Baghel [39] to solve a multiple traveling salesperson problem (MTSP) where the number of vehicles is fixed. In our studied problem, due to the uncertainty of vehicle number and the choice of the power station, the individual may have

different tour numbers and different individual lengths. Thus, a new revised recombine operator is proposed to solve the studied problem, the details of the reorganization operation are as follows:

Let x_1 and x_2 be the two selected parents to recombine operator and let m_1 and m_2 be the number of tours present in x_1 and x_2 , and mR is defined as the minimum tour number $mR = \min(m_1, m_2)$. The recombine operator can be divided into the following steps. The first step, sorting all tours by their distances in an ascending way within x_1 and x_2 , and then it constructs a offspring individual x' in an iterative way. In each iteration, it selects one of the two parent individuals randomly and moves the i^{th} tour to x' . This process is repeated in mR times in our case and the parents are selected with equal probability. Clearly, at the end of first stage, some customers still remain unassigned and they will be moved into an unassigned list.

In the second stage, we adopt the method proposed by Singh and Baghel [39] to generate good solutions. During each iteration, the customers in the unassigned list are reassigned one by one to x' . Each time it assigns a customer to a particular tour whose distance increases the least by this reassignment. To achieve this, we have to check all possible insertion positions in every tour. Simultaneously, in order to avoid having only the depot in the route of the generated offspring x' , a repair operation is performed. In this repair operation, we choose a tour randomly that has more than two customers, and move one customer to the route which has only one depot. The pseudo code of the recombine algorithm is given in Algorithm 5.

Algorithm 5: The pseudo-code of recombine.

Input: Two parent individuals x_1 and x_2
Output: The new offspring individual x'
Begin
 Sort the tours of x_1 and x_2 according to their distance
 mR =the minimum tour number of x_1 and x_2
for $i = 1$ to mR **do**
 r =random()
 If $r > 0.5$ **then**
 S =get tour i from x_1
 Else
 S =get tour i from x_2
 End
 add S to x'
 Delete the same city of S from both x_1 and x_2
End
 unassigned list←Unallocated city
for $j = 1$ to the size of unassigned list **do**
 Insert city j from unassigned list to a tour of x' by minimum
 distance increment
End
If there is only the depot in a tour **then**
 x' =repair(x')
End
 return x'
End

4.6. Mutate Operator

In the mutate operator, all tours in x are sorted in ascending order. Each customer from a tour i is moved to x' with varied probability P_{copy} , otherwise the customers are moved to the unassigned list. This process is repeated until all customers are moved from x . Then, the customers in the unassigned list will be reassigned to x' by the same process in the recombine operator.

Venkatesh and Singh [41] explained why they did not use the constant probability, but the probability related to the number of iterations and they defined P_{copy} in detail, it can be determined by the following equation:

$$P_{copy} = \frac{(P_{maxcopy} - P_{mincopy})}{iter_{max}} \times iter + P_{mincopy}. \quad (19)$$

Where $iter$ is the running time or the number of iterations, $iter_{max}$ is the maximum running time or the maximum number of iterations, $P_{maxcopy}$ and $P_{mincopy}$ represent respectively the maximum value and the minimum value of P_{copy} . The algorithm for mutate is given in Algorithm 6.

Algorithm 6: The pseudo-code of mutate

Input: Two selected individuals x
Output: The new offspring individual x'
Begin
 Sort the tours of x according to their distance
 TN =the tour number of x
for $i = 1$ to TN **do**
 r =random()
 If $r < P_{copy}$ **then**
 S =get tour i from x
 Else
 unassigned list=get tour i from x
 End
 add S to x'
 Delete the same city of S from x
End
for $j = 1$ to the size of unassigned list **do**
 Insert city j from unassigned list to a tour of x' by minimum distance increment
End
If there is only the depot in a tour **then**
 x' =repair(x')
End
 return x'
End

4.7. Sequential Variable Neighborhood Descent

Variable Neighborhood Search (VNS) is a metaheuristic based on a systematic change of the neighborhood structures within a search introduced by Mladenović and Hansen [42]. It is based on the idea of a systematic change of neighborhood both in a descent phase to find a local optimum and in a perturbation phase to get out of the corresponding valley [43]. It has been proved that it could solve various combinatorial optimization problems. The Sequential Variable Neighborhood Descent (SVND) is a sample variant of the basic VNS where the systematic changes of the neighborhood structures are performed in a deterministic way. The pseudo code of SVND is shown in Algorithm 7. It should be noted that there are two ways to explore local search, one is called first improvement strategy (a move made when an improvement in the neighborhood is once found) and the other is called the best improvement strategy (a move to the best solution in the neighborhood) [44]. In this work, we adopted the former.

The SVND in this paper is applied as a local search algorithm to improve the performance of a global search algorithm, and five neighborhood structures are proposed in this study ($l_{max} = 5$). Among all the new individuals produced, the SVND is used to improve the best one among them. The SVND method starts from a given solution x , and the l th neighborhood structure is applied to

generate a new solution x' , then the fitness value $f(x')$ is compared with the current value $f(x)$. If the new value is better, the solution x moves to x' and continue the search with premier neighborhood ($l \leftarrow 1$), otherwise the next neighborhood structure is applied with $l \leftarrow l + 1$. If the search operation runs l_{max} times without finding better values. The process will be terminated. Specifically, during the search process, an extra search technique 2_opt is applied when the new solution x' is better than the incumbent, it generates x'' and update the solution x only when $f(x'') < f(x)$. The 2_opt method and the neighborhood structure in the SVND algorithm are described in the following.

Algorithm 7: The pseudo-code of SVND

Input: the best offspring individual x
Output: The new offspring individual x'
Begin
 $l \leftarrow 1$
while $l < l_{max}$ **do**
 $x' \leftarrow lth_neighborhoodmove(x)$
if $f(x') < f(x)$ **then**
 $x \leftarrow x', f(x) \leftarrow f(x')$
 $l \leftarrow 1$
else
 $l \leftarrow l + 1$
end
 $x'' \leftarrow 2_opt(x)$
if $f(x'') < f(x)$ **then**
 $x \leftarrow x'', f(x) \leftarrow f(x'')$
end
end while
return x
End

4.7.1. 2_Opt Method

The 2_opt method used in this paper was proposed by Lin et al. [45]. It is a simplified form of the Lin-Kernighan algorithm [46], which is known as the k-opt method. In this paper, we can get a better solution by constantly comparing the distances of customers in the cycle. If the following equation is satisfied, then the indices of targets $i + 1$ and j should be swapped. Such comparisons can be repeated many times over a route, number of repetitions equals to the length of route-2. The same operation finally updates all routes. The pseudo-code for the 2_opt method is given in Algorithm 8.

$$d_{i,j+1} + d_{j,j+1} > d_{i,j} + d_{i+1,j+1} \quad (20)$$

4.7.2. Neighborhood Search Structure

The performance of the local search procedure depends on the choice of neighborhood structures. The size of the neighborhood is a key factor; the larger the neighborhood, the more likely it could contain better solutions [46]. We utilize five neighborhood structures in this work, for example, Swap, Insertion, Reverse, Insertion stations and Delete stations. The first three neighborhood structures are common changes in other VRP papers, and the latter two are unique to the hybrid vehicle routing problem in this paper. These operators are briefly explained below and illustrated in Figure 2.

1. **Swap:** The swap move is carried out by exchanging the positions of randomly selected i and j nodes of solution x . It should be noted that i and j may come from the same route or may come from different routes. That is to say, the swap operation may be an Intra-route movement or an Inter-route movement.

2. Insertion: The insertion is implemented by randomly selecting a node from index i and inserting it immediately following a randomly selected j node of x . Similar to swap, insertion operation may be an Intra-route movement or an Inter-route movement.
3. Reverse: A reverse is performed by randomly selecting i and j nodes from the same route of solution x and then reversing the substring in-between them.
4. Insertion Stations: An insert station is executed by choosing a station each from S_e and S_f , and then inserting them into the preceding random i index on solution x . Fuel stations, electric stations and insertion locations are randomly selected.
5. Delete Stations: A delete station means randomly taking out a power station from solution x . Before performing this operation, it is necessary to determine whether there is an power stations in the solution, if there are power stations in the solution, select a deletion from the existing power station, and if not, the operation does not make any changes to the solution.

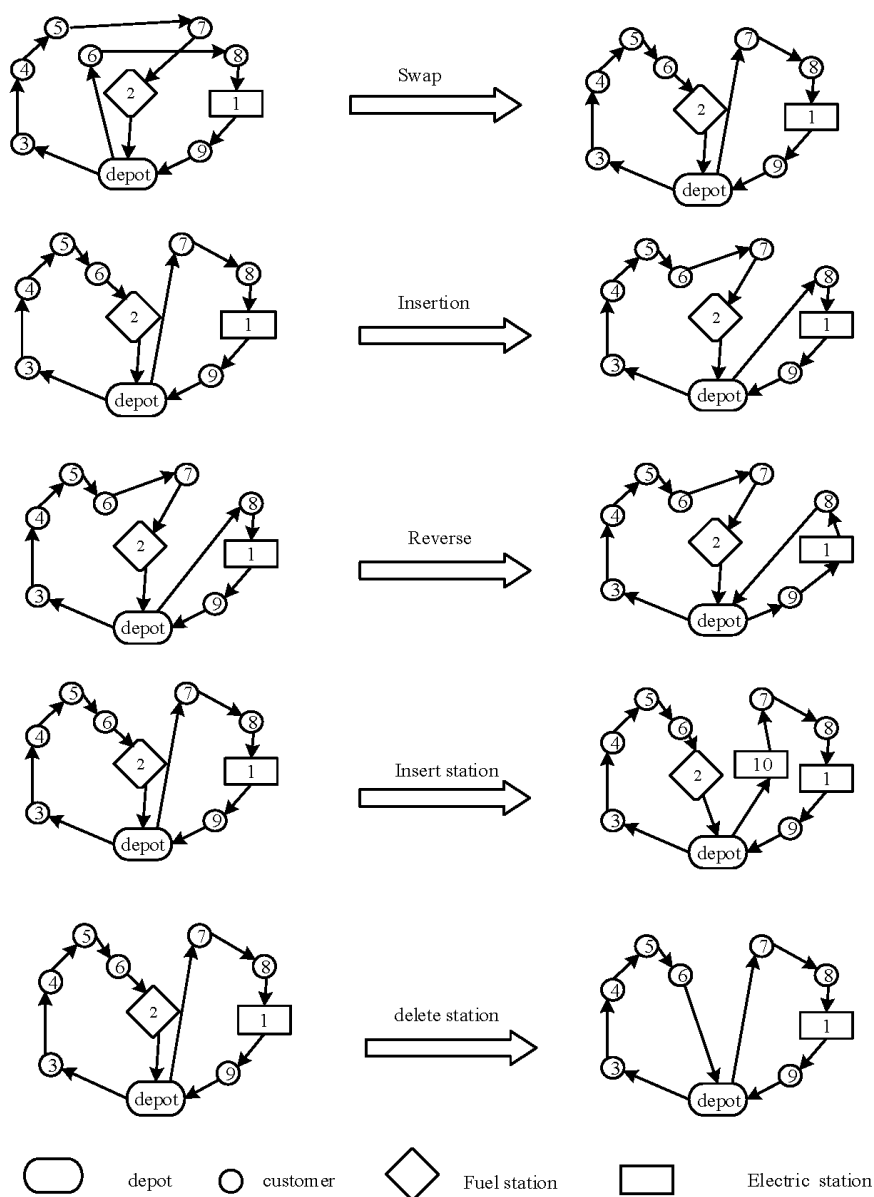


Figure 2. Neighborhood structures.

Algorithm 8: The pseudo-code of 2_opt method

```

Input: individual  $x$ 
Output: improved individual  $x'$ 
Begin
  for  $l = 1$  to  $R$  do
    for  $i = 1$  to  $n_l - 2$  do
      for  $j = i + 2$  to  $n_l$  do
         $d_1$  = total length for the 2 edges
         $d_2$  = total length of the edges when the targets are swapped
        if  $d_1 > d_2$  then
          Swap indices of targets in tour  $l$ 
        end
      end
    end
  end
End

```

4.8. Hybrid Memetic Algorithm Based on SVND for Solving PHEV Routing Problem

In conjunction with the above sections, the overall scheme of the proposed algorithm in this paper is shown in Figure 3. To make our solution more applicable to the PHEV routing problem, we made some customized improvement on the Memetic algorithm. First, for population initialization, we proposed a new algorithm based on the nearest neighborhood heuristic, it takes the factors of time, vehicle capacitance and fuel capacity as well as the access to the energy station into account. Secondly, we introduced penalty mechanism to maintain the diversity of population. Last, we improved the traditional 2_opt method with a cyclic comparison technique. The algorithm can be summarized as the following steps.

1. Initialization operation is used to generate a set of solutions, an initial solution is generated by rules and others are generated randomly. The total number of initial solutions is defined as P_s .
2. The 2_opt method is used to improve the generated solutions.
3. The objective function evaluation is carried out on these solutions.
4. The binary tournament selection operation is applied to choose the parents, then recombine operator or mutate operator is executed to generate new offspring. This step is repeated P_s times, in order to generate a new population with P_s individuals.
5. Evaluating the new population by using objective evaluation.
6. The SVND procedure is used to further improve the best individual at each generation, which searches all its neighborhoods according to a sequence and returns the best individual among all the searched neighbors.
7. Select the best individuals from the current population and the offsprings, they are considered as the population for the next generation, and the population size are always kept to P_s .
8. If the stopping criterion is satisfied, the algorithm will be terminated and export the result, otherwise go to step 4.

Steps 4 to 7 are executed in the iteration loop, in which the better solutions have a larger probability to be selected for the recombine operator or the mutate operator. It is noted that the recombine and mutation operators are used in a mutually exclusive manner, each offspring is generated by a recombine operator or a mutate operator but not both. The former is employed with probability P_c , otherwise the latter operator is used. The number of new individuals generated in step 4 of each iteration is P_s . In step 7, we arrange the total solutions in ascending order and select the non-repetitive P_s individuals in turn, if the objective values of the two solutions are identical but the compositions of the solutions

are different, then both solutions should be retained, provided that they will be arranged in P_s . Finally, the termination of the algorithm needs to satisfy the stopping condition.

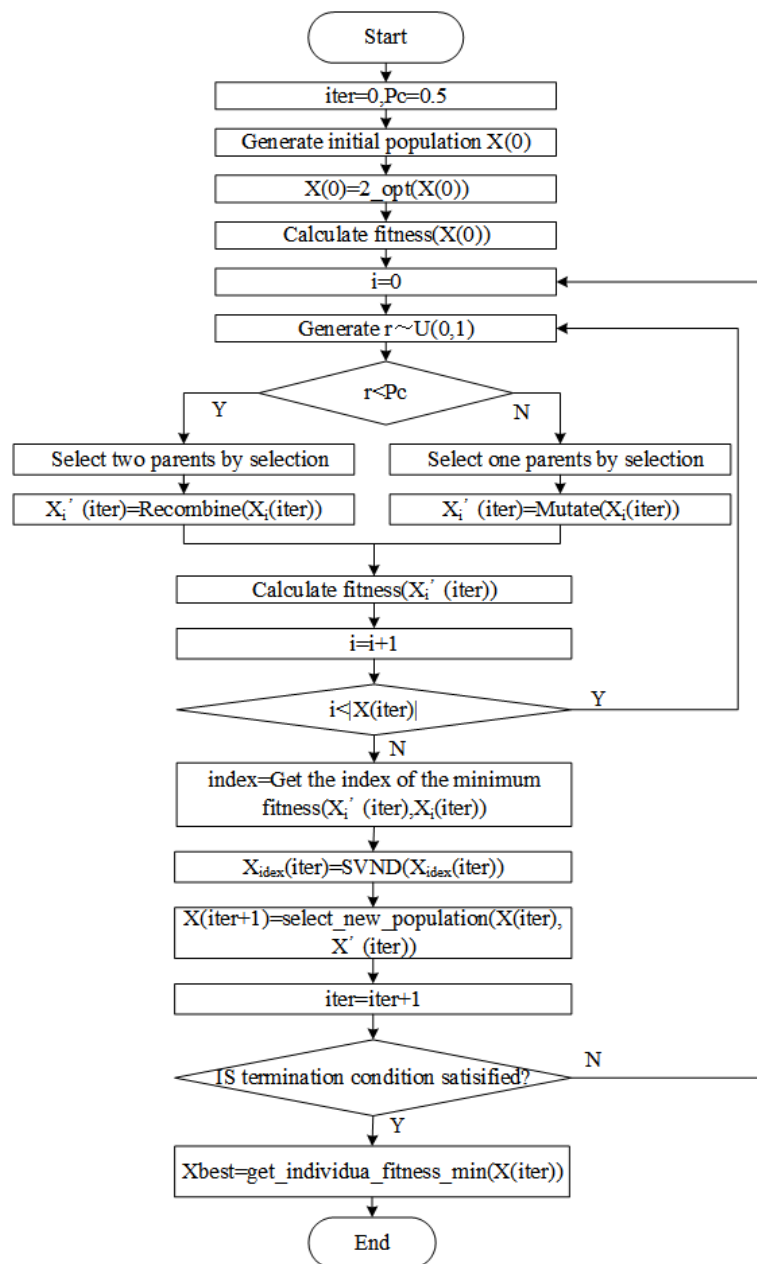


Figure 3. Flowchart: Hybrid Memetic Algorithm.

5. Computational Experiments

The experimental results and analysis are described in detail in this section. A comparative study is also conducted with the work of Yu et al. [8]. The proposed method was coded in C++ language and compiled on Microsoft Visual Studio 2017. the computing platform configuration are as following: Intel (R) Core (TM) i7 at 4.20 GHz, 16 GB of RAM with 64-bit platform under Windows 10 Operating System.

The vehicle data used in this paper are the same as the work of Yu et al. [8]. The relevant data are shown in Table 1. The test data we used for testing is a classic example of open source vehicle routing problem. It is an extension of the classical benchmark of VRP. The data of these instances are derived from Capacitated Vehicle Routing Problem (CVRP). This instances consists of datasets

for CVRP proposed by Augerat et al.(datasets A, B, and P), by Christofides and Eilon (for dataset E), by Fisher(for dataset F). All instances are available in <http://neo.lcc.uma.es/vrp/vrp-instances/>.

Table 1. Data of Vehicles Specification.

Specification	Total
Electric capacity	15 kwh
Maximum electric power can be used	10.5 kwh
Electric consumption rate	0.5 kwh/mile
Maximum electric distance	21 miles
Electric cost	US dollar 0.12/kWh
Fuel capacity	25 gallons
Fuel consumption rate	17.7 mpg
Maximum fuel distance	17.7*25 = 442.5 miles
Fuel cost	US dollar 4.18/gal
Velocity	40 mph
T_{max}	11 h
Penalty value P	25 dollars

In order to better study the impact of electric stations and fuel stations on PHEV routing problem research, we divided the test into three situations, taking different number of electric stations and fuel stations for testing. The first case consists of 0 electric station and 0 fuel station (E0F0). The second scenario considers 2 electric stations and 2 fuel stations (E2F2) and the third case has 4 electric stations and 4 fuel stations (E4F4). In order to facilitate the comparison with existing methods, we use the same station locations as in the work of Yu et al. [8]. Running tests were conducted on 14 kinds of test data under these three conditions, and their costs and distances were recorded. Specifically, for CVRP, it can be understood as a specific case of our E0F0 situation in this work where the fuel and electric stations are not considered. By replacing the time constraint with load constraint, some modifications to the proposed algorithm can be used to solve CVRP. Compared with HVRP defined by Simona Mancini, our optimization goal is to minimize the total cost and our article covers all the constraints of their articles and adds a few additional constraints. It is shown that the proposed algorithm can solve the problems with energy stations and general vehicle routing problems without energy stations.

Parameter sensitivity analysis has been done in the work of Venkatesh and Singh, Singh and Gupta. Therefore, in this work, our parameter settings are consistent with theirs which are $P_c = 0.5$, $P_{better} = 0.8$, $P_{mincopy} = 0.15$, $P_{maxcopy} = 0.9$, $iter_{max} = 1000$. When E2F2 and E4F4 are used, l_{max} is set to 5 and has five neighborhood structures. When E0F0 is used, l_{max} is set to 3. At this time, insert and delete station are not performed. For the layout of the fuel station and electric station, the methods we used are as follows—E2F2, 1 and 2 for charging stations, 3 and 4 for gas stations, then the customer sequence will start from 5, and -1 for depot. Similarly, we can get the distribution of E4F4.

Yu et al. [8] compared Plug-in Hybrid Electric Vehicle (PHEV) with Mild Hybrid Vehicles and Conventional Vehicles. The results suggested that PHEV has more advantages than the others. Thus, we considered the PHEV in our work.

This paper focuses on the PHEV routing problem, which considers a series of constraints in the VRP problem such as time limits, fuel capacity, electric capacity, fuel station and electric station. Table 2 presents the experimental results of the proposed algorithm for this problem with different power station arrangement (e.g., E0F0, E2F2 and E4F4). We have 14 instances tested under these cases, and each configuration is tested 10 times. The size of the population is 1000, the number of iterations is set to 100. With the increasing number of customers in test cases, the average CPU time of these instances are between 3 and 30 s. The best value of cost, the mean value of cost, the related mile value of the best obtained solution and the mean value of mile for each configuration are presented in Table 2. Since the objective of this paper is to minimize the total cost, the related mile in the table refer to the distances corresponding to the solution with best cost value, instead of the shortest distance in the 10 runs. In most cases, the related mile is less than the average mile in Table 2, but there are few exceptions, such

as instance B-35-K5 under E4F4 where the related mile is 408 miles and the average mile is 401.2 m. In this paper, electricity or fuel can be selected for each route, and the HVRP is an extension of GVRP, so we give priority to using electric energy. Therefore, as long as the electric stations are visited enough times along a route, there may be a long distance but low costs for the route.

In order to verify the performance of the proposed algorithm, we considered simultaneously the precision and the robustness. Table 3 shows the comparison results of HMA_SVND and SA_RS_CF to solve the 14 instances with different power station arrangements. The SA_RS_CF is the method proposed by Yu et al. [8]; it means that simulated annealing with a restart strategy and using the Cauchy Function to determine the acceptance probability of poor solutions. The best obtained solution with the lowest cost and related mile of the different methods are presented in this table. We proved that HMA_SVND stability analysis presented in Table 4 where the standard deviation and error value have been shown.

Table 2. Testing results of HMA_SVND for different instances.

Instances	E0F0				E2F2				E4F4			
	best cost	Aver. cost	related mile	Aver. mile	best cost	Aver. cost	related mile	Aver. mile	best cost	Aver. cost	related mile	Aver. mile
A-n33-k5	96.45	99.16	439	450.30	90.51	93.24	398	431.90	77.98	82.25	417	430
A-n46-k7	133.94	137.25	597	611.70	123.58	125.04	585	591.17	115.42	116.70	597	605.17
A-n60-k9	150.44	154.03	667	682.29	136.45	140.24	681	695.14	127.46	130.67	664	669.86
B-n35-k5	88.45	89.90	390	396	78.03	78.69	402	405.90	65.34	65.56	408	401.20
B-n45-k5	109.92	111.29	496	501.60	100.74	105.61	488	539.50	86.20	88.15	520	523.30
B-n68-k9	92.19	95.28	406	420.20	81.98	85.24	420	458.50	69.03	72.63	439	458.80
B-n78-k10	136.10	141.03	607	627.30	112.83	117.69	566	581.30	102.66	104.66	603	605.80
E-n30-k3	84.49	85.18	388	391.17	74.98	76.95	376	378.33	63.58	65.85	333	335.50
E-n51-k5	99.99	102.43	454	463.80	88.44	90.86	449	467.20	75.77	78.61	451	459.30
E-n76-k7	127.46	132.84	570	592.90	116.22	119	592	632.90	104.44	105.55	619	639.20
F-n72-k4	45.60	45.97	208	209.75	37.37	37.96	218	218.50	30.82	32.33	215	219.25
F-n135-k7	190.63	194.85	852	870.50	166.85	171.83	930	947.60	158.49	163.87	892	893.90
P-n76-k4	128.84	133.13	575	594	118.30	120.35	650	638.50	104.41	106.76	624	644.80
P-n101-k4	157.80	161.43	699	714	142.97	149.82	780	784.60	127.34	134.10	750	790.50

In Table 3, we have shown the comparison results of HMA_SVND and SA_RS_CF. The first line of Table 3 is explained. This problem is defined with 33 nodes, in the situation of E0F0, there are 1 depot and 32 customers to be visited, and this problem is considered as a classical CVRP problem. In the case of E2F2, there is 1 depot, 2 electric stations, 2 fuel stations and 28 customers, their indexes are arranged in the above order, in the case of E4F4, there is also a similar circumstance with 4 electric stations and 4 fuel stations. The cost value and mile value of HMA_SVND and SA_RS_CF correspond their best found solution. In the first line, the HMA_SVND has obtained better solutions than SA_RS_CF under both 3 situations. For most cases in Table 3, HMA_SVND has found better solution than the other (39 cases were better of the 42 configurations), the comparison results proved the effectiveness of the HMA_SVND algorithm.

In order to present the comparison results more clearly, we draw the bar chart shown in Figure 4, where Figure 4a–c are the results of the comparison under E0F0, E2F2 and E4F4, respectively. According to Figure 4, we can clearly observe the effectiveness of our proposed algorithm HMA_SVND, which outperforms the SA_RS_CF algorithm to a great extent. According to these graphs, it is not difficult to find that in some benchmark problems, our proposed algorithm can greatly improve the solution, some are slightly improved for most cases.

In order to prove the robustness of the algorithm, the standard deviation (SD) and error of HMA_SVND for tested configurations are shown in Table 4, the latter is defined by the following equation

$$Error = \frac{Aver - Opt}{Opt}, \times 100\%. \tag{21}$$

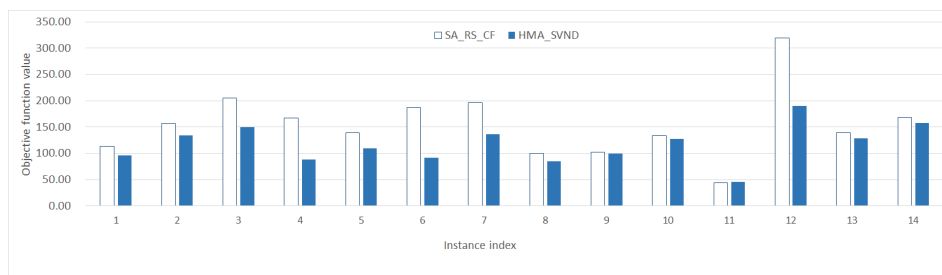
where Aver is the mean value of the results for 10 runs, Opt is the best value (related value for mile) for the 10 runs. As shown in Table 4, in the cases of E0F0, the maximum cost Error is 4.2% and the minimum cost Error is 0.8%, the SD values of cost are between 0.35 and 3.46, in the cases of E2F2, the cost SD values and cost error values are between 0.51 and 0.38, 0.8% and 4.8% respectively, in the cases of E4F4, the cost SD values are from 0.2 to 4.39, the cost error values are from 0.3% to 5.4%. In this study, our aim is to minimize the total cost, the value of mile is the total distance of the solution which has the minimum cost value. The total distance is not the considered point of our problem; therefore, the stability of mile values for the tested instances is less than that of the cost value. In a special case of the B-n35-K5 in E4F4, the mile error is -1.6% , this is because the related mile is greater than the average value in this case. The standard deviation and the error could be used to express the degree of data dispersion. The smaller the two values are, the more centralized distribution of experimental results is, on the contrary, the more decentralized the distribution is.

Table 3. Comparison results of HMA_SVND (proposed) and SA_RS_CF [8].

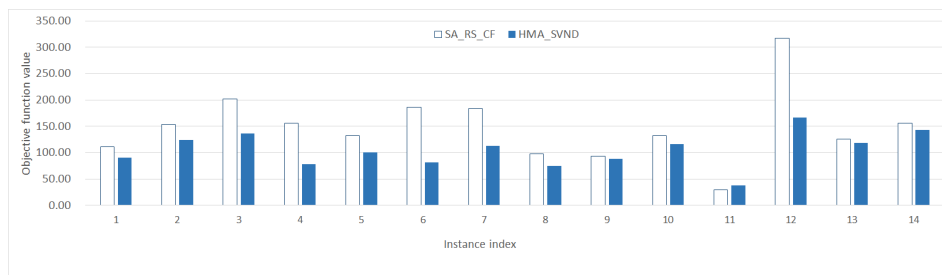
Instances	E0F0				E2F2				E4F4			
	SA_RS_CF		HMA_SVND		SA_RS_CF		HMA_SVND		SA_RS_CF		HMA_SVND	
	cost	mile	cost	mile	cost	mile	cost	mile	cost	mile	cost	mile
A-n33-k5	113.70	521	96.45	439	111.36	527	90.51	398	103.98	512	77.98	417
A-n46-k7	157.68	720	133.94	597	154.08	750	123.58	585	150	754	115.42	597
A-n60-k9	205.50	944	150.44	667	201.84	979	136.45	681	195.90	970	127.46	664
B-n35-k5	166.80	758	88.45	390	155.58	778	78.03	402	144.06	814	65.34	408
B-n45-k5	138.72	641	109.92	496	132.66	678	100.74	488	123.66	768	86.20	520
B-n68-k9	187.32	875	92.19	406	186.30	981	81.98	420	174	914	69.03	439
B-n78-k10	196.50	929	136.10	607	183.84	952	112.83	566	171.18	996	102.66	603
E-n30-k3	100.26	465	84.49	388	97.38	522	74.98	376	86.82	556	63.58	333
E-n51-k5	102.48	490	99.99	454	93.24	534	88.44	449	81.78	574	75.77	451
E-n76-k7	133.98	637	127.46	570	132.90	754	116.22	592	113.82	775	104.44	619
F-n72-k4	44.40	248	45.60	208	29.76	322	37.37	218	22.2	349	30.82	215
F-n135-k7	319.08	1487	190.63	852	317.64	1481	166.85	930	313.14	1616	158.49	892
P-n76-k4	138.84	673	128.84	575	125.46	726	118.30	650	114.84	741	104.41	624
P-n101-k4	168.86	813	157.80	699	155.88	870	142.97	780	154.08	981	127.34	750

Table 4. Standard deviation and Error on tested instances.

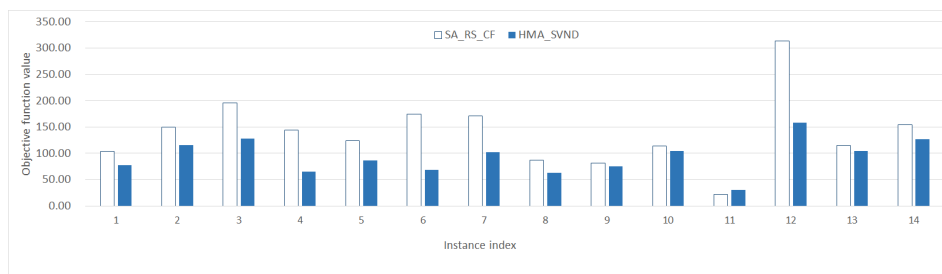
Instances	Standard Deviation						Error					
	E0F0		E2F2		E4F4		E0F0		E2F2		E4F4	
	cost	mile	cost	mile	cost	mile	cost	mile	cost	mile	cost	mile
A-n33-k5	2.44	10.40	3.38	26.69	3.75	9.94	2.8%	2.6%	2.9%	7.8%	5.4%	3.1%
A-n46-k7	2.84	11.88	0.86	7.14	1.10	10.19	2.4%	2.4%	1.1%	1.0%	1.1%	1.3%
A-n60-k9	2.68	11.22	2.18	21.42	2.37	22.16	2.4%	2.3%	2.7%	2.0%	2.5%	0.8%
B-n35-k5	1.81	7.57	0.51	10.96	0.20	8.36	1.7%	1.5%	0.8%	0.9%	0.3%	-1.6%
B-n45-k5	0.95	3.98	2.99	28.67	1.60	11.76	1.2%	1.1%	4.8%	10.5%	2.3%	2.4%
B-n68-k9	2.32	13.09	2.38	27.11	2.38	28.22	3.3%	3.5%	4.0%	9.2%	5.2%	4.5%
B-n78-k10	3.46	14.52	2.99	10.32	4.04	7.47	3.6%	3.3%	4.3%	2.7%	1.9%	0.5%
E-n30-k3	0.65	2.93	1.20	4.03	1.42	12.47	0.8%	0.8%	2.6%	0.6%	3.6%	0.8%
E-n51-k5	2.97	12.84	2.45	16.74	2.40	20.03	2.4%	2.2%	2.7%	4.0%	3.7%	1.8%
E-n76-k7	2.39	10.02	1.55	21.02	0.84	15.38	4.2%	4.0%	2.3%	6.9%	1.0%	3.2%
F-n72-k4	0.35	1.66	0.69	2.32	0.95	8.75	0.8%	0.8%	1.5%	0.2%	4.8%	1.9%
F-n135-k7	3.29	14.36	2.75	13.18	3.79	8.65	2.2%	2.2%	2.9%	1.8%	3.4%	0.2%
P-n76-k4	3.35	14.51	1.40	7.04	1.83	22.49	3.3%	3.3%	1.7%	-1.7%	2.2%	3.3%
P-n101-k4	2.81	11.98	3.36	14.63	4.39	24.45	2.3%	2.1%	4.8%	0.6%	5.3%	5.4%



(a) Best results comparisons under E0F0



(b) Best results comparisons under E2F2



(c) Best results comparisons under E4F4

Figure 4. The comparative results of the proposed HMA_SVND method and SA_RS_CF method of Yu et al. [8]

In terms of the robustness, Figure 5a,b graphically illustrate the box plot of the standard deviation and error of cost for different power station configurations. It can be observed that, in the case of E0F0 and E2F2, the stability of the algorithm is better when the number of energy stations increases, the robustness of the algorithm will become slightly worse.

In summary, compared to SA_RS_CF algorithm, our proposed HMA_SVND algorithm obtained a better performance in terms of precision and the robustness.

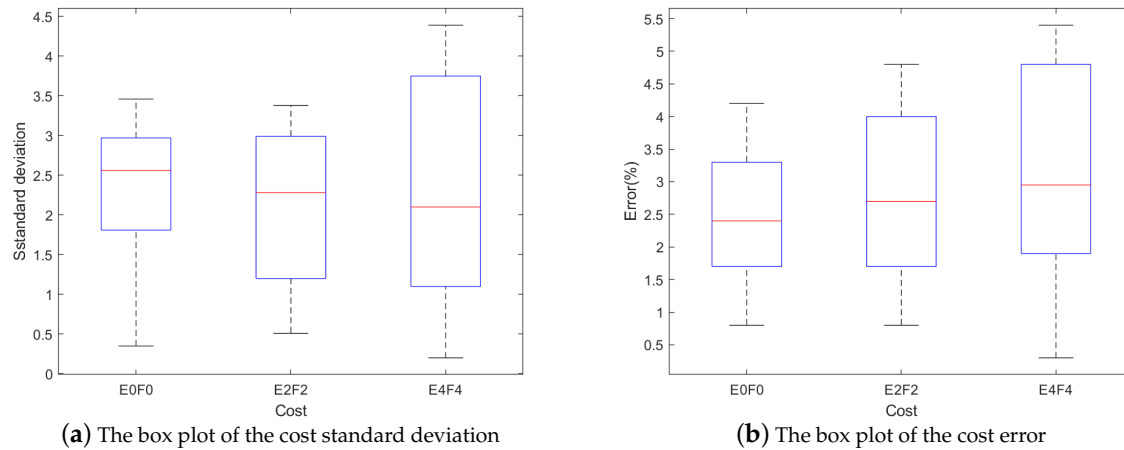


Figure 5. The graphics of the comparison result.

6. Conclusions

In this paper, we focused on solving the routing problem for PHEV, a new hybrid optimization algorithm which merges the memetic algorithm and the sequential variable neighborhood descent is proposed. The proposed approach exhibits a competitive performance in terms of precision and robustness, achieving superior results to prevalent algorithms. However, our studied problem is still based on some simplified assumptions. For example, the charging time of the charging station was not considered and the electric station is set to be always available. In real life, charging time and mode should be considered, and there may be queues at stations; vehicles may need to wait for charging and refilling in energy stations, so the cost of waiting should also be considered. At the same time, each customer may have a time window for their services. Therefore, in future research, we will concentrate on more comprehensive problems for real-life PHEV routing.

Author Contributions: The research work was conducted in collaboration with all authors. X.L., X.S. and H.L. defined the research theme and designed the proposed method. X.L., X.S. and Y.Z. implemented the prototype and wrote the article. X.S. and Y.D. discussed and evaluated the data test. Both authors have thoroughly contributed to reading and validating the manuscript. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: This work was partly supported by Fundamental Research Funds for the Xi'an Science and Technology Project (201805045YD23CG29), the Natural Science Basic Research Plan in Shaanxi Province of China(2018JM5156) and the Shaanxi Key Project of Research and Development Plan(2019ZDLGY03-01)

Conflicts of Interest: Authors declare no conflict of interest.

Abbreviations

The notations is introduced as follows:

Sets

Z	Set of power source, $Z = \{e, f\}$
M	Set of vehicles
I	Set of customers
I_0	Set of customers vertices and depot, $I_0 = I \cup \{v_0\}$
S_e	Set of electric stations
S_f	Set of fueling stations
Φ_e	Set of dummy electric stations
Φ_f	Set of dummy fuel stations
S_0	Set of stations vertices including depot, fuel stations, electric stations and dummy stations, $S_0 = \{v_0\} \cup S_e \cup S_f \cup \Phi_e \cup \Phi_f$
V	Sets of vertices(without depot) and dummy vertices, $V = I \cup S_0$
V_0	Sets of vertices and dummy vertices, $V_0 = I_0 \cup S_0$

Coefficients

n	Number of customers
m	Number of vehicles
p_j	Service time at vertex j ; if $i \in I$, then p_j is the service time at the customer vertex; if $i \in S_0$, then p_j is the refueling time or recharging time at the station vertex
d_{ij}	Distance from vertex i to vertex j
t_{ij}	Time of arrival at vertex j from vertex i
Q_{kz}	Vehicle k 's tank capacity of power type z
r_{kz}	Vehicle k 's consumption rate (gallons per mile) on power type z
c_z	Cost rate for power source type z , which includes c_e and c_f
T_{max}	Pre-determined maximum time for travel from depot until return to depot
C	A large positive constant value

Decision variables

x_{ijk}	Binary variable equal to 1 if vehicle k travels from vertex i to vertex j ; otherwise, 0
y_{jkz}	The remaining tank level of vehicle k 's power type z upon arrival to vertex j
τ_j	Time variable specifying the time of arrival of a vehicle at vertex j , initialized to zero upon departure from the depot
f_{ijkz}	Vehicle k 's consumption from vertex i to vertex j using power type z (electric or fuel)

References

1. Beamon, B.M. Supply chain design and analysis: models and methods. *Int. J. Prod. Econ.* **1998**, *55*, 281–294.
2. Dantzig, G.B.; Ramser, J.H. The truck dispatching problem. *Manag. Sci.* **1959**, *6*, 80–91.
3. Soysal, M.; Çimen, M.; Demir, E. On the mathematical modeling of green one-to-one pickup and delivery problem with road segmentation. *J. Clean. Prod.* **2018**, *174*, 1664–1678.
4. Lin, C.; Choy, K.L.; Ho, G.T.S.; Chung, S.H.; Lam, H.Y. Survey of green vehicle routing problem: past and future trends. *Expert Syst. Appl.* **2014**, *41*, 1118–1138.
5. Erdoğan, S.; Miller-Hooks, E. A green vehicle routing problem. *Transp. Res. Part E Log. Transp. Rev.* **2012**, *48*, 100–114.
6. Mancini, S. The hybrid vehicle routing problem. *Transp. Res. Part C Emerg. Technol.* **2017**, *78*, 1–12.
7. Doppstadt, C.; Koberstein, A.; Vigo, D. The hybrid electric vehicle-traveling salesman problem. *Eur. J. Oper. Res.* **2016**, *253*, 825–842.
8. Yu, V.F.; Redi, A.A.N.P.; Hidayat, Y.A.; Wibowo, O.J. A simulated annealing heuristic for the hybrid vehicle routing problem. *Appl. Soft Comput.* **2017**, *53*, 119–132.
9. Hiermann, G.; Hartl, R.F.; Puchinger, J.; Vidal, T. Routing a mix of conventional, plug-in hybrid, and electric vehicles. *Eur. J. Oper. Res.* **2019**, *272*, 235–248.
10. Björnsson, L.; Karlsson, S. Electrification of the two-car household: PHEV or BEV? *Transp. Res. Part C Emerg. Technol.* **2017**, *85*, 363–376.
11. Silva, C.; Ross, M.; Farias, T. Evaluation of energy consumption, emissions and cost of plug-in hybrid vehicles. *Energy Convers. Manag.* **2009**, *50*, 1635–1643.

12. Dekker, R.; Bloemhof, J.; Mallidis, I. Operations research for green logistics an overview of aspects, issues, contributions and challenges. *Eur. J. Oper. Res.* **2012**, *219*, 671–679.
13. Sbihi, A.; Eglese, R.W. Combinatorial optimization and green logistics. *Ann. Oper. Res.* **2007**, *5*, 99–116.
14. Messaoud, E.; Idrissi, A.E.B.E.; Alaoui, A.E. The Green Dynamic Vehicle Routing Problem in Sustainable Transport. In Proceedings of the 2018 4th International Conference on Logistics Operations Management (GOL), Le Havre, France, 10–12 April 2018.
15. Leu, J.; Krischke, A.; Lee, Y.; Lee, L.J.; Huang, Y. A Green Vehicle Routing Method for the Regional Logistics Center. In Proceedings of the 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Bangkok, Thailand, 16–19 December 2018; pp. 71–75.
16. Ubeda, S.; Arcelus, F.J.; Faulin, J. Green logistics at Eroski: A case study. *Int. J. Prod. Econ.* **2011**, *131*, 44–51.
17. Zhou, Y.; Lee, G.M. A Lagrangian Relaxation-Based Solution Method for a Green Vehicle Routing Problem to Minimize Greenhouse Gas Emissions. *Sustainability* **2017**, *9*, 776–790.
18. Zhang, J.; Zhao, Y.; Xue, W.; Li, J. Vehicle routing problem with fuel consumption and carbon emission. *Int. J. Prod. Econ.* **2015**, *170*, 234–242.
19. Demir, E.; Bektaş, T.; Laporte, G. The bi-objective pollution-routing problem. *Eur. J. Oper. Res.* **2014**, *232*, 464–478.
20. Schiffer, M.; Walther, G. The electric location routing problem with time windows and partial recharging. *Eur. J. Oper. Res.* **2017**, *260*, 995–1013.
21. Shao, S.; Guan, W.; Bi, J. Electric vehicle-routing problem with charging demands and energy consumption. *IET Intell. Transp.* **2018**, *12*, 202–212.
22. Roberti, R.; Wen, M. The Electric Traveling Salesman Problem with Time Windows. *Transp. Res. Part E Logis. Transp. Rev.* **2016**, *89*, 32–52.
23. Keskin, M.; Çatay, B. Partial recharge strategies for the electric vehicle routing problem with time windows. *Transp. Res. Part C Emerg. Technol.* **2016**, *65*, 111–127.
24. Mavrovouniotis, M.; Ellinas, G.; Polycarpou, M. Ant Colony Optimization for the Electric Vehicle Routing Problem. In Proceedings of the 2018 IEEE Sympos. Series on Comput. Intell (SSCI), Bangalore, India, 18–21 November 2018; pp. 1234–1241.
25. Schneider, M.; Stenger, A.; Goeke, D. The electric vehicle routing problem with time windows and recharging stations. *Transp. Sci.* **2014**, *48*, 500–520.
26. Hannan, M.A.; Azidin, F.A.; Mohamed, A. Hybrid electric vehicles and their challenges: A review. *Renew. Sustain. Energy Rev.* **2014**, *29*, 135–150.
27. Wen, C.K.; Chen, J.; Teng, J.; Ting, P. Decentralized plug-in electric vehicle charging selection algorithm in power systems. *IEEE Trans. Smart Grid* **2012**, *3*, 1779–1789.
28. Macrina, G.; Pugliese, L.D.P.; Guerriero, F.; Laporte, G. The green mixed fleet vehicle routing problem with partial battery recharging and time windows. *Comput. Oper. Res.* **2019**, *101*, 183–199.
29. Sun, Z.; Zhou, X. To save money or to save time: Intelligent routing design for plug-in hybrid electric vehicle. *Transp. Res. Part D Trans. Environ.* **2016**, *43*, 238–250.
30. Abdallah, T. The Plug-In Hybrid Electric Vehicle Routing Problem with Time Windows. Master's Thesis, The University of Waterloo, Waterloo, ON, Canada, 2013.
31. Vidal, T.; Crainic, T.G.; Gendreau, M.; Prins, C. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *Eur. J. Oper. Res.* **2013**, *231*, 1–21.
32. Avci, M.; Topaloglu, S. An adaptive local search algorithm for vehicle routing problem with simultaneous and mixed pickups and deliveries. *Comput. Ind. Eng.* **2015**, *83*, 15–29.
33. Moscato, P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. In *Caltech Concurrent Computation Program*; C3P Report; California Institute of Technology: Pasadena, CA, USA, 1989; Volume 826.
34. Vidal, T.; Crainic, T.G.; Gendreau, M.; Lahrachi, N.; Rei, W. A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems. *Oper. Res.* **2012**, *60*, 611–624.
35. Li, X.; Zhao, Y.; Shi, X.; Shi, T. A hybridization of Memetic algorithm with SVND for solving a Hybrid Vehicle Routing Problem. In Proceedings of the 2019 4th International Conference Mathematics and Artificial Intelligence (ICMAI), Chengdu, China, 12–15 April 2019.
36. Burke, E.K.; Hyde, M.R.; Kendall, G.; Ochoa, G.; Özcan, E.; Woodward, J.R. A Classification of Hyperheuristic Approaches. In *Handbook of Metaheuristics*; Springer: Heidelberg, Germany, 2010; pp. 453–477.

37. Dawkins, R. *The Selfish Gene*; Oxford University Press: Oxford, UK, 1977.
38. Goksal, F.P.; Karaoglan, I.; Altiparmak, F. A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. *Comput. Ind. Eng.* **2013**, *65*, 39–53.
39. Singh, A.; Baghel, A.S. A new grouping genetic algorithm approach to the multiple traveling salesperson problem. *Soft Comput.* **2009**, *13*, 95–101.
40. Singh, A.; Gupta, A.K. Two heuristics for the one-dimensional bin-packing problem. *OR Spectr.* **2007**, *29*, 765–781.
41. Venkatesh, P.; Singh, A. Two metaheuristic approaches for the multiple traveling salesperson problem. *Appl. Soft Comput.* **2015**, *26*, 74–89.
42. Mladenović, N.; Hansen, P. Variable neighborhood search. *Comput. Oper. Res.* **1997**, *24*, 1097–1100.
43. Hansen, G.; Mladenović, N. Variable neighborhood search. *Eur. J. Oper. Res.* **2008**, *191*, 593–595.
44. Mladenović, N.; Urošević, D.; Hanafi, S.; Ilić, A. A general variable neighborhood search for the one-commodity pickup-and-delivery travelling salesman problem. *Eur. J. Oper. Res.* **2012**, *220*, 270–285.
45. Lin, S.; Kernighan, B.W. An effective heuristic algorithm for the traveling salesman problem. *Oper. Res.* **1973**, *21*, 498–516.
46. Walsh, T.; Beek, P.V.; Rossi, F. *Handbook of Constraint Programming*; Elsevier Science: New York, NY, USA, 2006; pp. 1–978, ISBN 978-0444527264.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).