International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

Hybrid electric vehicle routing problem with mode selection

Lu Zhen, Ziheng Xu, Chengle Ma & Liyang Xiao

To cite this article: Lu Zhen, Ziheng Xu, Chengle Ma & Liyang Xiao (2019): Hybrid electric vehicle routing problem with mode selection, International Journal of Production Research, DOI: 10.1080/00207543.2019.1598593

To link to this article: https://doi.org/10.1080/00207543.2019.1598593

Published online: 29 Mar 2019.

International Journal of Production Research, 2019
https://doi.org/10.1080/00207543.2019.1598593

Hybrid electric vehicle routing problem with mode selection

Lu Zhen, Ziheng Xu, Chengle Ma and Liyang Xiao *

School of Management, Shanghai University, Shanghai, People's Republic of China

(*Received 7 June 2018; accepted 11 March 2019*)

With the development of green logistics, logistics companies gradually are paying attention to the application of hybrid electric vehicles (HEVs). HEVs have the advantages of low energy consumption and pollution, while their disadvantage mainly lies in their limited continuous driving range. Therefore, it is necessary to optimize the use of fuel during the distribution process. We study the mode selection system in HEVs based on the background of green logistics and the above characteristics of HEVs. The mode selection system can adjust the driving mode of the HEV according to different road conditions to obtain the optimal use of fuel. In this paper, we propose a new study of a hybrid electric vehicle routing problem with mode selection. This problem is formulated as a mixed integer linear programming model. An improved particle swarm optimization algorithm (IPSO) is developed to solve this problem. Extensive numerical experiments are conducted to validate the effectiveness of the proposed model and the efficiency of the proposed solution method. The experimental results show that our proposed algorithm not only obtains the optimal solution for some small-scale problem instances and some medium-scale problems but also solves some large-scale situations (one hundred customers, eleven vehicles, eleven charging stations, eleven gas stations and four modes) within an hour.

Keywords: Vehicle routing; hybrid vehicle; mode selection; particle swarm optimization; mixed integer linear programming

1. Introduction

Under the current shortage of petroleum resources and the requirement of sustainable development, finding new sources of energy instead of traditional fuel has become an international consensus. Under the environment of vigorously developing low-carbon economy, green logistics has become one of the important directions for the development of the modern logistics industry. According to relevant investigations, 60%-70% of CO and 40% of NO_x in the air come from automobile exhaust. The International Energy Agency reported that China's carbon emissions in 2015 were 618 tons, equivalent to 2% of the world's fuel emissions (Wu et al. 2017). Therefore, energy savings and emission reductions of automobiles will play a key role in solving the air pollution problem. Green vehicle routing problem has emerged, with the goal of reducing fuel consumption and reducing carbon emissions by deploying conventional or alternative fuel vehicles (AFVs) in transportation and logistics activities (Erdoğan and Elise,2012). Different from conventional vehicles, AFVs are powered by greener fuel sources, namely, biodiesel, natural gas, electricity, etc. For example, electric commercial vehicles (ECVs) can significantly reduce fuel consumption and reduce carbon emissions from logistics activities and thus they have outstanding environmental advantages. However, their overall industrial development is not yet mature, and they are limited to technologies such as fuel batteries. The problems of short distances and high cost have not yet been solved. As a transition product and a tradeoff between traditional products and ECVs, hybrid electric vehicles (HEVs) have become a global hot spot for the development of new environmental vehicles. HEVs have strong development prospects and market potential.

An HEV is powered by two power sources, it consumes both electricity and gasoline during driving. The energy consumption of an HEV on each road segment depends on the HEV driving modes, which include the following cases: mainly powered by the electric motor (battery-based mode); mainly powered by the engine (gasoline-based mode); the two are jointly driven (balance mode); or only powered by the engine (only gasoline mode). For example, an HEV can use a battery-based mode if there is a traffic jam on the road. An HEV can use the gasoline-based mode to drive if the traffic condition of the next road section is good. The mode selection system is added to the HEV to achieve optimal energy consumption. The HEV can choose the appropriate mode according to different road conditions. In logistics activities, the HEV also needs to go to the charging station to charge or to go to the gas station to refuel. In this paper, we consider the mode selection described above in our hybrid electric vehicle routing problem (HEVRP). The new study will be called HEVRP with mode

*Corresponding author. Email: xiaoliyang0509@gmail.com

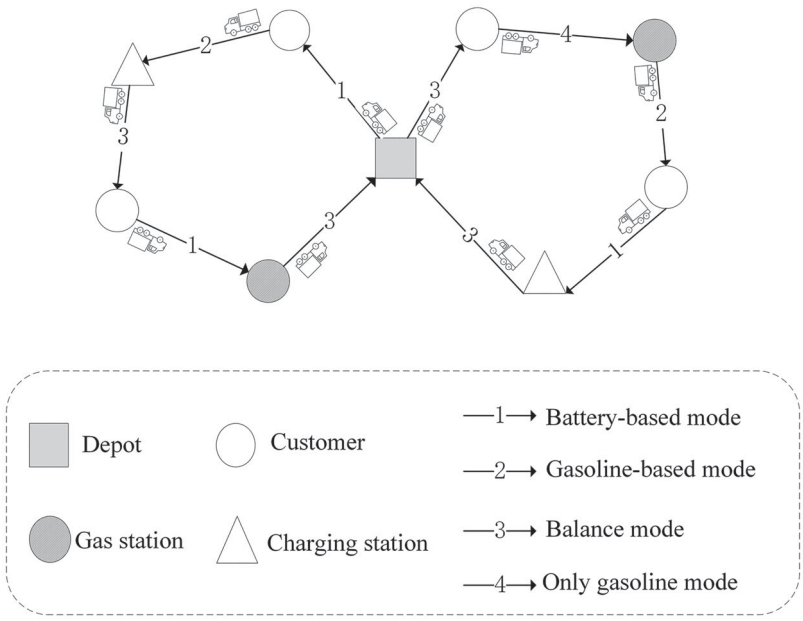


Figure 1. A case of a hybrid electric vehicle routing problem with mode selection.

selection (as shown in Figure 1), which is formulated as a mixed integer linear programming model. General commercial solution software, e.g. CPLEX, can only obtain optimal solutions for small-scale situations in a reasonable time. Thus, we developed an improved particle swarm optimization (IPSO) to solve large-scale problems in a relatively short time.

The rest of this article is organized as follows. Section 2 reviews related works. Section 3 describes the problem and formulates a mixed integer linear programming model. Section 4 proposes a heuristic algorithm. Section 5 shows the results of numerical experiments. A brief conclusion of the paper is given in Section 6.

2. Related works

The vehicle routing problem (VRP) was first proposed by Dantzig and Ramser (1959). In the following decades, the VRP has been continuously expanded and developed. Some works, such as Berger (2003), Prins, Labadi, and Reghioui (2009), Stenger et al. (2013) and Geetha (2014), developed some heuristics for variants of the VRP, while some exact algorithms were also developed for the VRPs, for example, by Baldacci, Christofides, and Mingozzi (2008) and Baldacci, Hadjiconstantinou, and Mingozzi (2017). Readers who are interested in VRPs can read excellent reviews from Laporte (2009), Toth and Vigo (2014).

As there are few studies on the HEVRP, this section mainly reviews studies on the electric VRP (EVRP). Schneider, Stenger, and Goeke (2014) proposed an EVRP with time windows, during which the electric vehicles can recharge in charging stations. The authors also proposed a heuristic algorithm based on a tabu search/variable neighbourhood search for this problem. Ángel et al. (2014) considered charging stations with different cost and recharging speeds, and then designed a heuristic algorithm that includes multiple local search heuristics and simulated an annealing algorithm. Hiermann et al. (2016) considered the effect of recharging time on the EVRP and proposed a hybrid heuristic that includes an adaptive large neighbourhood search and labelling procedure. Desaulniers et al. (2016) studied four cases of EVRP with time windows: (1) each route can only be charged once, and it is charged fully each time; (2) each route can be charged multiple times, and it is charged fully every time; (3) each route can only be charged once, but the battery can be allowed to be unfilled; and (4) each route can be recharged multiple times, and the battery is allowed to not be full. The authors proposed an exact branch-price-and-cut algorithm for these four cases and finally determined, through experiments, that the algorithm can solve these four cases. Montoya et al. (2017) expanded the existing model of EVRP, established a travel route network through three heuristic methods for travelling salesman problems, and then used a hybrid heuristic algorithm to segment the route generated by this large travel route network. Finally, an EVRP solution is formed by solving the set partitioning problem through assembling on the generated line.

Regarding the reference for HEVRP, we mainly refer to Nejad et al. (2016). They studied the plug-in HEVRP and proposed an algorithm to solve this problem. However, they did not consider the mode selection regarding recharging and

refuelling. Mancini (2017) introduced the hybrid VRP, considered that the vehicle can only be driven in either pure electric or pure gasoline power. Finally, they proposed a large neighborhood search to solve this problem. The same problem has been addressed in Yu et al. (2017) where a simulated annealing is proposed. Our research extends the mode of HEVs, considering the simultaneous use of gasoline and electricity, and proposes four modes of HEVs. Moreover, we consider the mode selection for each road segment, which sharply increases the complexity of the problem. To the best of our knowledge, there is no literature devoted to HEVRP with mode selection. The contributions of this paper are listed as follows. First, we propose the HEVRP with model selection. Second, we formulate this problem as a mixed integer linear programming model, with the objective of minimizing total travel cost of energy consumption. Then, an IPSO is developed to solve the problem.

3. Problem description and formulation

3.1. Problem description

We elaborate some features of HEVs that are related to VRP modelling before building the model. HEVs can run in different modes as long as the battery or the gasoline tank is not empty. Therefore, we need to decide in advance whether to go to gas stations or charging stations for fuel supplies. Following common VRP modelling techniques, we simplify several real-world characteristics and neglect the impact of load on fuel consumption.

The hybrid electric VRP with mode selection can be defined on a directed network $G = (V, E)$, where V is the set of nodes that is partitioned into a depot 0, a set of customers N, a set of charging stations G_e and a set of gas stations G_g, and $E = \{(i, j)|i, j \in V\}$ is an arc set. Each arc represents the road segment that connects vertex i and vertex j, where $i, j \in V$. There are a total of K HEVs in depot 0, and each HEV has a load capacity of Q. There is an underlying assumption for the model that the demand of each customer does not exceed Q. This assumption may hold for the most cases in reality. In case the customer demand d_i' exceeds Q, it will be decomposed into an integral multiple of Q and a remainder demand d_i: $d_i' = m \cdot Q + d_i$, where $m \in N^+, d_i < Q$. In this way, d_i will be scheduled for delivery with other customers' demands, $m \cdot Q$ quantities' demand are delivered by m extra trips to customer i.

In addition to load capacity restriction, each HEV also has a battery capacity of Q_e, and a fuel tank capacity of Q_g. Each HEV has a full battery and gasoline tank before departure. The HEV will be recharged fully when the HEV visits a charging station, and the HEV will be refuelled fully when the HEV visits a gas station. Our final goal is to minimize the total travel cost of energy consumption (fuel and electricity) of all the HEVs; thus, we use p^g to represent the price per gallon of gas; p^e represents the price per kW · h.

Since the energy consumed per road segment ($arc(i, j)$) depends on speed, we therefore establish a unit-distance gasoline consumption function $f_m^g(V_{ij})$ and a unit-distance battery consumption function $f_m^e(V_{ij})$ related to driving speed V_{ij}. Figure 2 is a function diagram of $f_m^g(V_{ij})$ and $f_m^e(V_{ij})$ in the four modes (battery-based mode, gasoline-based mode, balance mode, only gasoline mode). We define a variable d_{ij}, which represents the length of $arc(i, j)$. In addition, another variable g_m is used to indicate the amount of gasoline consumed per unit of distance in the standard case of mode m. The variable e_m is used

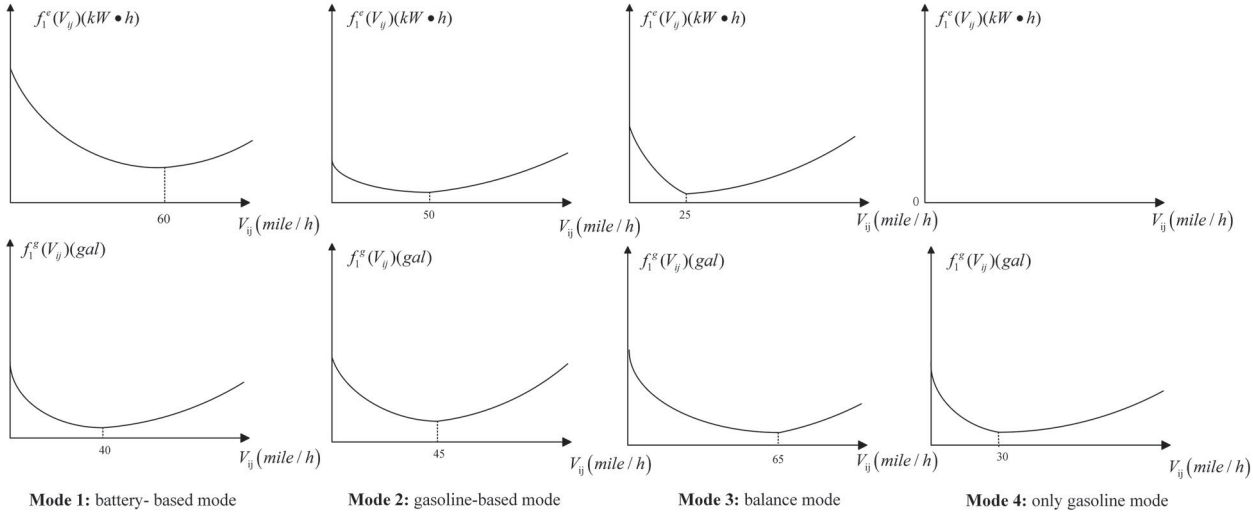


Figure 2. $f_m^g(V_{ij})$ and $f_m^e(V_{ij})$ function diagram in four modes.

to indicate the amount of battery consumed per unit of distance in the standard case of mode m. For example, the electric consumption of $arc(i,j)$ using mode 1 is equal to $d_{ij}f_1^e(V_{ij})e_1$. The gasoline consumption of $arc(i,j)$ using mode 1 is equal to $d_{ij}f_1^g(V_{ij})g_1$. We can define the variable $W_{ijm}^e(W_{ijm}^g)$ as the battery (gasoline) consumption when using mode m to travel $arc(i,j)$.

Some assumptions are listed as follows before formulating the model of the HEVRP with mode selection:

- All HEVs have the same properties and are available at the depot.
- All HEVs must start from the depot and return to the depot.
- The HEV can go to the gas station (charging station) for refuelling (charging) during a trip.
- The load of the HEV for each trip cannot exceed Q.
- The demand of each customer cannot exceed Q.
- Each customer is served exactly once by one vehicle.
- HEVs can only choose one mode to drive each $arc(i,j)$.
- When HEVs reach the customers or gas stations, the remaining battery keeps the same.
- When HEVs reach the customers or charging stations, the remaining gasoline keeps the same.

3.2. Model formulation

Indices and sets:

0	Depot
G_g	Set of gas stations
G_e	Set of charging stations
V	Set of nodes
N	Set of customers
K	Set of vehicles
R	Set of modes
p^e	Electricity price
p^g	Gasoline price
Q_e	Battery capacity
Q_g	Fuel tank capacity
Q	Load capacity
d_i	Demand of customer i.
w_{ijr}^e	Battery consumption when vehicle is using mode m to travel $arc(i,j)$.
w_{ijr}^g	Gasoline consumption when vehicle is using mode m to travel $arc(i,j)$.

Decision variables:

λ_{ik}^e	The amount of electricity when vehicle k reaches point i
μ_{ik}^e	The amount of electricity when vehicle k leaves point i
λ_{ik}^g	The amount of gasoline when vehicle k reaches point i
μ_{ik}^g	The amount of gasoline when vehicle k leaves point i
λ_{ik}^q	The amount of load when vehicle k reaches point i
μ_{ik}^q	The amount of load when vehicle k leaves point i
y_{ik}	Binary equals one if vehicle k reaches node i; otherwise, it equals zero
x_{ijkr}	Binary equals one if vehicle k visits customer j immediately after visiting customer i use mode r

3.2.1. Mathematical model

$$Minminze\, Z = \sum_{i\in V}\sum_{j\in V}\sum_{k\in K}\sum_{r\in R}(p^e w_{ijr}^e + p^g w_{ijr}^g)x_{ijkr} \tag{1}$$

Subject to:

$$\sum_{k\in K} y_{ik} = 1 \quad \forall i \in N \tag{2}$$

$$\sum_{k\in K} y_{ik} \le 1 \quad \forall i \in G_g \cup^{G_e} \tag{3}$$

$$\sum_{i\in V}\sum_{r\in R} x_{ilkr} = \sum_{j\in V}\sum_{r\in R} x_{ljkr} = y_{lk} \quad \forall l \in V, k \in k, r \in R \tag{4}$$

$$\sum_{i\in V}\sum_{r\in R} x_{ilkr} = \sum_{j\in V}\sum_{r\in R} x_{ljkr} \quad \forall l \in 0 \cup^{G_g} \cup^{G_e}, k \in K, r \in R \tag{5}$$

$$\sum_{i\in V}\sum_{r\in R} x_{ilkr} + \sum_{j\in V}\sum_{r\in R} x_{ljkr} \le M y_{lk} \quad \forall l \in 0 \cup^{G_g} \cup^{G_e}, k \in K, r \in R \tag{6}$$

$$\mu^e_{ik} \le Q_e + M(1 - y_{ik}) \quad \forall i \in G_e \cup^0 k \in K \tag{7}$$

$$\mu^e_{ik} \ge Q_e - M(1 - y_{ik}) \quad \forall i \in G_e \cup^0 k \in K \tag{8}$$

$$\mu^g_{ik} \le Q_g + M(1 - y_{ik}) \quad \forall i \in G_g \cup^0 k \in K \tag{9}$$

$$\mu^g_{ik} \ge Q_g - M(1 - y_{ik}) \quad \forall i \in G_g \cup^0 k \in K \tag{10}$$

$$\mu^e_{ik} - w^e_{ijr} \le \lambda^e_{jk} + M(1 - x_{ijkr}) \quad \forall i, j \in V, k \in K, r \in R \tag{11}$$

$$\mu^e_{ik} - w^e_{ijr} \ge \lambda^e_{jk} - M(1 - x_{ijkr}) \quad \forall i, j \in V, k \in K, r \in R \tag{12}$$

$$\mu^g_{ik} - w^g_{ijr} \le \lambda^g_{jk} + M(1 - x_{ijkr}) \quad \forall i, j \in V, k \in K, r \in R \tag{13}$$

$$\mu^g_{ik} - w^g_{ijr} \ge \lambda^g_{jk} - M(1 - x_{ijkr}) \quad \forall i, j \in V, k \in K, r \in R \tag{14}$$

$$\mu^q_{ik} \le \lambda^q_{jk} + M(1 - x_{ijkr}) \quad \forall i, j \in V, k \in K, r \in R \tag{15}$$

$$\mu^q_{ik} \ge \lambda^q_{jk} - M(1 - x_{ijkr}) \quad \forall i, j \in V, k \in K, r \in R \tag{16}$$

$$\mu^q_{0k} \le Q \quad \forall k \in K \tag{17}$$

$$\mu^q_{0k} = \sum_{i\in V} d_i y_{ik} \quad \forall k \in K \tag{18}$$

$$\mu^q_{ik} \le \lambda^q_{ik} - d_i + M(1 - y_{ik}) \quad \forall i \in N, k \in K \tag{19}$$

$$\mu^q_{ik} \ge \lambda^q_{ik} - d_i - M(1 - y_{ik}) \quad \forall i \in N, k \in K \tag{20}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ijkr} \le |S| - 1 \quad \forall S \subseteq N, \forall k \in K, \forall r \in R \tag{21}$$

$$\lambda^e_{ik} \ge 0 \quad \forall i \in V, k \in K \tag{22}$$

$$\mu^e_{ik} \ge 0 \quad \forall i \in V, k \in K \tag{23}$$

$$\lambda^g_{ik} \ge 0 \quad \forall i \in V, k \in K \tag{24}$$

$$\mu^g_{ik} \ge 0 \quad \forall i \in V, k \in K \tag{25}$$

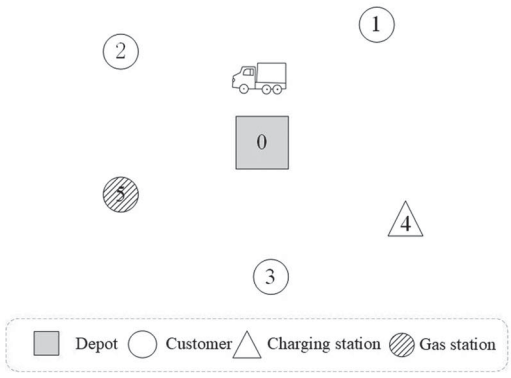
$$\lambda^q_{ik} \ge 0 \quad \forall i \in V, k \in K \tag{26}$$

$$\mu^q_{ik} \ge 0 \quad \forall i \in V, k \in K \tag{27}$$

$$y_{ik} \epsilon \{0, 1\} \quad \forall i \in V, k \in K \tag{28}$$

$$x_{ijkr} \in \{0, 1\} \quad \forall i \in V, j \in V, k \in K, r \in R \tag{29}$$

The objective function (1) minimizes the total travel cost of energy consumption. Constraint (2) guarantees that each customer can only be served by one HEV. Constraint (3) indicates that the same HEV can only pass through the same charging



W^e_{ij1}/W^g_{ij1}	0	1	2	3	4	5
0	0/0	15/3	16/2	2/2	18/4	16/3
1	15/3	0/0	6/1	17/2	3/3	2/8
2	16/2	6/1	0/0	15/2	10/2	13/3
3	2/2	17/2	15/2	0/0	12/2	18/4
4	18/4	3/3	10/2	12/2	0/0	12/3
5	16/3	2/8	13/3	18/4	12/3	0/0

W^e_{ij2}/W^g_{ij2}	0	1	2	3	4	5
0	0/0	10/4	18/3	10/1	18/3	14/5
1	10/4	0/0	2/4	12/3	10/1	16/4
2	18/3	2/4	0/0	2/3	4/5	8/7
3	10/1	12/3	15/2	0/0	3/8	4/8
4	18/3	10/1	4/5	3/8	0/0	2/7
5	14/5	16/4	8/7	4/8	2/7	0/0

Figure 3. Illustrative example of a HEV routing problem with mode selection.

station or gas station at most one time. Constraints (4)-(6) indicate that the number of arcs reaching and leaving the same node must be equal. Constraints (7) and (8) ensure that the battery is recharged fully when the HEV leaves the depot or charging stations. Constraints (9) and (10) ensure that the fuel tank is refuelled fully when the HEV leaves the depot or gas stations. Constraints (11) and (12) guarantee the HEV battery constraint from node i to node j. Constraints (13) and (14) guarantee the HEV gasoline constraint from node i to node j. Constraints (15) and (16) impose the consistency of the HEV load from node i to node j. Constraint (17) ensures that the load of the HEV when leaving the depot does not exceed Q. Constraint (18) indicates that the total customer demand served by vehicle k is equal to the cargo volume when vehicle k leaves depot. Constraints (19) and (20) show that when a vehicle leaves a customer which it has just served, the decrease of the load is equal to the demand of this customer. Constraint (21) is used for sub tour elimination constraints. Constraints (22)-(29) define the domains of decision variables.

3.3. An illustrative example

We present an example shown in Figure 3 to illustrate the HEVRP with mode selection. Node 0 represents the depot; nodes 1, 2 and 3 represent customers; node 4 is the charging station; and node 5 is the gas station. In this example, we give W^e_{ijm} and W^g_{ijm} in two modes. There is an HEV in the depot; the HEV has a battery capacity of 20 kW · h and a fuel tank of 20 gallons. To find the target cost, we assume that p^g equals \$2 per gallon and that p^e equals \$0.2 per kW · h. The goal is to minimize the total energy consumption cost. Most people use a greedy-based method to solve this problem, where they first find a path with minimum gasoline consumption from 0 to 1. In this example, the person who goes from the depot to customer 1 will choose mode 1; then, the final route will be $0-1-4-2-3-0$, and the mode sequence of each road will be $1-1-1-2-1$; the final energy consumption cost is \$30.80. However, the optimal solution is the path $0-1-2-3-0$, and the mode sequence of each road is $2-1-2-1$; the final energy consumption cost is \$24.

4. Algorithmic strategies

The model developed in Section 3 can be solved by some commercial software applications (e.g. CPLEX) for small-scale settings. We constructed an IPSO to solve large-scale situations. Different from the classic PSO, the IPSO is hybridized with a local search/variable neighbourhood search (LS/VNS) mechanism and label procedure. The LS/VNS mechanism obtains a better solution by systematically changing the neighbourhood structure during the search process. The role of label procedure is to assign a mode to each road segment.

4.1. Algorithm framework

This section lists our IPSO framework for the HEVRP with mode selection. PSO was proposed by Kennedy and Eberhart (1995). Particles represent candidate solutions in a PSO system. First, we randomly initialize the particle swarm, with a

length of $|N| + |G_e|+|G_s|$ for each particle, and we generate a random position for each particle that indicates which nodes each vehicle passes through. We use a label procedure to calculate the fitness of each particle. We then use the LS/VNS mechanism for the neighbourhood search and use a label procedure to evaluate the quality of the sequence. Finally, we update the particle's individual optimal value and global optimal value according to the particle's fitness. The outline of the method is given in Algorithm 1. There are two ways to update the particles: (*a*) is the feedback mechanism, (*b*) is the PSO position update formula; and *a* and *b* will be described in detail later.

Algorithm 1: IPSO framework

```
1:  Set ct ← 0, uc ← 0, f* ← ∞
2:  initialize swarm
3:  while ct < Maxepochs&uc < Maxepochs
4:  for each particle i
5:     f ← Evaluation of particle i (label procedure)
6:     f′ ← Education of particle i(LS/VNS)
7:     if f′ < f
8:        update particle positions(a)
9:     if f < f*
10:    f* ← f
11:       update best particle cost and best global cost
12: end for
13: update particle positions(b)
14: end while
15: output f*
```

4.2. Initialization

We first refer to the second line in Algorithm 1. According to the characteristics of this model, we use natural number coding (Zhen 2015). The depot's code is 0; $1, 2, \ldots, n$ represents the natural number assigned to each customer node; $n+1, n+2, \ldots, n+|G_e|$ represents the natural number assigned to each charging station; $n+|G_e|+1, n+|G_e|+2, \ldots, n+|G_e|+|G_s|$ represents the natural number assigned to each gas station. The position of each particle is generated randomly, which is represented as decimals between $[0, |K|)$. To be specific, the customer and station nodes associated with values between $[(k-1), k)$ are assigned to vehicle k, where $k \in K$. The initialized particle obtains its fitness through a label procedure, and the label procedure will be explained later. The initialized particle obtains its fitness through a label procedure, and the label procedure will be explained later.

4.3. LS/VNS mechanism

This mechanism refers to the sixth line in Algorithm 1. In this paper, we propose three methods of neighbourhood search, listed as follows:

- Exchange: for each sequence, randomly select two nodes and exchange their positions.
- Shift: for each sequence, randomly choose a point and move it to another location.
- 2-opt: the 2-opt algorithm was first proposed by Croes (1958). The operation of this algorithm is to reverse the sequence between two nodes.

Hansen, Lazić, and Mladenović (2007) first proposed the VNS algorithm. The algorithm begins from an initial solution and improves the quality of the solution by searching different neighbourhoods. The continuous increase in the neighbourhood range makes the algorithm have the ability to jump out of the local optimum. Particles that have been updated after the LS/VNS mechanism will feedback their positions to the optimal particle. This operation ensures that each iteration is updated with the best particle, therefore speed up the process. The outline of the LS/VNS mechanism is shown in Algorithm 2.

4.4. Label procedure

Referring to the fifth line in Algorithm 1, the label procedure is the core step of this paper. The purpose of label procedure is to assign a mode for each road segment and calculate the particle's fitness.

Algorithm 2: LS/VNS mechanism

```
S_0 ← initial sequence, flg ← true,
while flg is true
    flg ← false
    search a new neighbour S' of S_0
        Evaluation (Labelling Procedure)
        if R(S') < R(S_0)
            flg ← true; S_0 ← S'
            break
    end search
end while
```

Each node produces a series of labels for each vehicle's node sequence. Each label contains seven parts. The first part indicates the mode used from node i to node j. The second part shows the remaining battery power at the arrival node j. The third part represents the amount of gasoline remaining at node j. The fourth part indicates the cost of the label. The fifth part records whether it is eliminated. The sixth part records feasibility. The seventh part records the mode used by the upper road segment.

First, we generate a new cost function to evaluate each label. For a label with feasible solution, the feasibility indicator 'F' is denoted as 'F = Y'. Note that a solution may be labelled infeasible due to one of the load capacity, remaining battery or remaining gasoline restriction is violated: overload, running out of battery or gasoline. In the process of label procedure, when these violations are occurred, penalizations are resulted in cost function and the label is marked as infeasible, e.g. 'F = N'. Let ρ represent the sequence of nodes a vehicle passes through. $q(\rho)$ is the amount of load of node sequence ρ. $e(\rho)$ is the amount of battery of node sequence ρ. $g(\rho)$ is the amount of gasoline of node sequence ρ. $\theta_q, \theta_e, \theta_g$ are load penalty factors, battery penalty factors, and gasoline penalty factors, respectively. The new cost function is as follows:

$$\mathrm{C}(\rho) = Z(\rho) + (\theta_q \max(0, q(\rho) - Q) + \theta_e \max(0, e(\rho) - Q_e) + \theta_g \max(0, g(\rho) - Q_g) \tag{30}$$

To avoid the number of labels at the back node increasing too quickly, we propose a dominance rule to mitigate this situation. Let φ_1 and φ_2 indicate two labels under the same node. τ_e^1 and τ_e^2 are the remaining battery at the arrival node i, where $\tau_e^1 = max(0, \lambda_{ik}^e)$, $\tau_e^2 = max(0, \lambda_{ik}^e)$. τ_g^1 and τ_g^2 are the remaining gasoline at the arrival node i, where $\tau_g^1 = max(0, \lambda_{ik}^g)$, $\tau_g^2 = max(0, \lambda_{ik}^g)$. let α and β are the dominant parameters of battery and gasoline, respectively. φ_1 dominates φ_2 if and only if

$$C(\varphi_1) + \alpha\eta(\varphi_1, \varphi_2) + \beta\kappa(\varphi_1, \varphi_2) \leq \gamma C(\varphi_2) \tag{31}$$

$$C(\varphi_1) \leq C(\varphi_2) \tag{32}$$

$$\eta(\varphi_1, \varphi_2) = max(0, \tau_e^1 - \tau_e^2) \tag{33}$$

$$\kappa(\varphi_1, \varphi_2) = max(0, \tau_g^1 - \tau_g^2) \tag{34}$$

where $\gamma \geq 1$.

Parameter γ will be adjusted dynamically based on the number of labels for each node, Cattaruzza, Absi, and Feillet (2013) provide a formula for the dynamic change of γ.

$$\gamma = \begin{cases} \gamma + \dfrac{|\vartheta_i|}{1000\vartheta_{thr}} if\, |\vartheta_i| > \vartheta_{thr} \\ \gamma - \dfrac{\vartheta_{thr}}{1000|\vartheta_i|} if\, |\vartheta_i| < \vartheta_{thr} \end{cases} \tag{35}$$

where $|\vartheta_i|$ is the number of labels on node i produced by the label procedure, and ϑ_{thr} is a threshold parameter that indicates the number of labels associated with each node. If the result of the above formula value γ is less than 1, then let γ equal $\gamma = \max\{1, \gamma\}$.

Finally, we select the feasible and lowest cost label that has not been eliminated as the current solution in the last node. We use the data in the third section to provide an example, assuming that the sequence of nodes is $0-1-2-3-0$, and the result of the label procedure is shown in Table 1. Here, we assume that θ_q, θ_e and θ_g are equal to 2, that α is equal to 1, and that β is equal to 1.5.

Table 1. Generation of labels corresponding to node sequence.

1							2							3							0						
r	λ_{ik}^{e}	λ_{ik}^{g}	C	D	F	P	r	λ_{ik}^{e}	λ_{ik}^{g}	C	D	F	P	r	λ_{ik}^{e}	λ_{ik}^{g}	C	D	F	P	r	λ_{ik}^{e}	λ_{ik}^{g}	C	D	F	P
1	5	17	9	N	Y		1	0	16	14.2	N	N	1	1	0	14	53.2	Y	N	1	1	0	10	24	N	Y	2
2	10	16	10	N	Y		2	3	13	17.4	Y	Y	1	2	0	13	26.6	Y	N	1	2	0	11	39.6	Y	N	2
							1	4	15	13.2	N	Y	2	1	0	13	42.2	Y	N	2							
							2	8	12	18.4	Y	Y	2	2	2	12	19.6	N	Y	2							

Note: 'r' represents the mode, 'C' represents the cost, 'D' represents the result of dominance rule, 'F' represents feasibility, 'P' represents the predecessor mode. In addition, in the columns of 'D' and 'F', 'N' means 'No' and 'Y' means 'Yes'.

From the results, we finally choose the label with cost $c = 24$ associated with node zero corresponding to the solution consisting of four mode transitions; the mode transition sequence is $2 - 1 - 2 - 1$.

4.5. Update particle position and velocity

We now refer to the thirteenth line in Algorithm 1. From the initial group, particles constantly adjust their position and velocity according to their own and peer flight experience. The position and velocity of particles are updated as shown in equations (36) and (37).

$$V_i^{k+1} = wV_i^k + c_1 Rand()(pB_i^k - X_i^k) + c_2 Rand()(gB^k - X_i^k) \tag{36}$$

$$X_i^{k+1} = X_i^k + V_i^k \tag{37}$$

where V_i^k is the velocity of particle i in the $k-th$ iteration; X_i^k is the position of particle i in the $k-th$ iteration; pB_i^k is the individual extremum of particle i; gB^k is the global extremum; $Rand()$ is a random number on the interval $[0, 1]$; w is the inertia weight; c_1 is the cognitive coefficient, adjusting the flight steps to pB_i^k; and c_2 is the social coefficient, adjusting the flight step to gB^k. The velocity and position of the particles are limited to a specific range during the iteration. At the same time, pB_i^k and gB^k are updated constantly. Finally, output gB^k is the global optimal solution.

5. Numerical experiments

In this section, we conduct numerical experiments by using a PC (Intel Core i7, 2.80 GHz; Memory, 8.00G) to validate the effectiveness of our proposed models and the efficiency of our developed solution methods. These models and solution methods, including the IPSO, are implemented by CPLEX 12.6.2 with concert technology of C# (VS2015).

5.1. Algorithm setting

Our numerical experiments are performed on small-scale situations, medium-scale settings, large-scale problems, and some experiments with parameter settings. The parameters of customer location, vehicle capacity and customer demands are based on Solomon's benchmark (Solomon 1987). The location of charging stations and gas stations are generated randomly in a two-dimensional coordinate area where the customers are located. The parameters related to the algorithm after testing are shown in Table 2. $\alpha, \beta, \vartheta_{thr}, \theta_q, \theta_e, \theta_g$ are important parameters that affect the size of the neighbourhood search and the quality of the solution. So after testing, the final values are shown in Table 2. For convenience, the values of W_{ijm}^{e} and W_{ijm}^{g} in the study are proportional to d_{ij} and v_{ij}, and different ratios in different modes.

In order to evaluate the role of VNS mechanism, we remove VNS mechanism, and then compare with the proposed algorithm. Table 3 shows the results of comparing the proposed algorithm (i.e. IPSO with the VNS) and the algorithm without the VNS mechanism. In this comparative experiment, we tested a total of 15 instances for 5, 20 and 50 customers.

In Table 3, the average gap value '44.78%' shows that VNS contributes to a distinct improvement of solution quality. Moreover, the average time ratio of IPSO without VNS to the one with VNS is 154.57%, which indicates that the VNS mechanism can not only increase the quality of the solution but also reduce the computation time.

Table 2. Results of the parameter settings.

Parameter		Final value
θ_q	Load penalty factors	5
θ_e	Battery penalty factors	5
θ_g	Gasoline penalty factors	5
α	Dominance parameters of battery	2
β	Dominance parameters of gasoline	10
ϑ_{thr}	Threshold parameter	25
c_1	Cognitive coefficient	1
c_2	Social coefficient	1
w	Inertia weight	1.3
p^e	Unit price of electricity	0.6
p^g	Unit price of gasoline	6
Numparticles	Number of particles	20
Maxepochs	Maximum number of iterations	300

Table 3. Comparison of IPSO and IPSO (Without VNS).

Instance	IPSO		IPSO(without VNS)		Gap	Time Ratio
ID	Z_p	$T_p(s)$	Z_{vp}	$T_{vp}(s)$	$(Z_{vp} - Z_p)/Z_p$	T_{vp}/T_p
5-2-2-2-1	99.3	0.91	144.6	2.01	45.62%	220.88%
5-2-2-2-2	101.7	0.88	150.3	1.99	47.79%	226.14%
5-2-2-2-3	123.7	0.93	173.2	2.34	40.02%	251.61%
5-2-2-2-4	99.0	1.01	138.2	3.06	39.60%	302.97%
5-2-2-2-5	127.1	0.95	160.5	1.56	26.28%	164.21%
20-2-2-2-1	445.5	162.9	671.9	201.4	50.82%	123.63%
20-2-2-2-2	467.5	166.0	712.6	170.3	52.43%	102.59%
20-2-2-2-3	502.9	171.9	711.4	191.6	41.46%	111.46%
20-2-2-2-4	467.2	148.4	598.4	180.3	28.08%	121.50%
20-2-2-2-5	510.9	159.8	801.6	170.4	56.90%	106.63%
50-5-5-5-1	2130.7	566.6	3089.4	661.4	44.99%	116.73%
50-5-5-5-2	1886.9	620.4	2763.9	670.9	46.48%	108.14%
50-5-5-5-3	2234.8	601.4	3899.2	690.3	74.48%	114.78%
50-5-5-5-4	2936.4	533.2	3756.4	672.4	27.93%	126.11%
50-5-5-5-5	2433.6	550.8	3621.3	667.5	48.80%	121.19%
Average	971.1	245.7	1426.2	285.8	44.78%	154.57%

5.2. Small-scale situations

We tested a total of 15 instances for 5, 8 and 10 customers, using four modes for each instance. Here, the instance ID contains five parts: the first part is the number of customers, the second part is the number of vehicles, the third part is number of charging stations, the fourth part is the number of gas stations, and the fifth part is the instance index.

Table 4 shows the solutions obtained by CPLEX for small-scale problems. For the sake of model performance standalone, we also compare the CPLEX solutions with the solutions obtained by a 'decision rule'. The decision rule is intuitive and greedy, due the fact that the unit price of gasoline is ten times of electricity: the vehicle always uses the battery-based mode (uses the least gasoline consumption mode for each road). 'Z_c' is the solution obtained by CPLEX, 'Z_r' indicates the solution obtained by decision rule, 'T_c' represents the CPLEX computation time in seconds. In addition, $Rgap = (Z_p - Z_c)/Z_c$ is also provided.

From the results in Table 4, the solutions obtained by the decision rule are not ideal by comparing to the optimal ones. However, when the customer number is more than eight, the CPLEX cannot find the optimal solution within two hours. And it is obviously unacceptable in real life.

The results of comparing IPSO with CPLEX are illustrated in Table 5, where Z_p is the solution obtained by the IPSO, and T_p is the IPSO computation time in seconds. The obj gap $= (Z_p - Z_c)/Z_c$, LB gap $= (Z_p - Z_l)/Z_l$ and Time ratio $= (T_p/T_c)$ are provided. Here, Z_l is the lower bound of the model by relaxing vehicle's capacity restriction. From the results in Table 5, we can see that the IPSO performs well in small-scale settings: the average obj gap is equal to 0, when the average LB gap is equal to 3.47% and when the average time ratio is equal to 0.45%. Our IPSO can reach optimality for first ten small-scale

Table 4. Comparison of the CPLEX and the decision rule in small-scale settings.

Instance	CPLEX		Decision rule	R Gap(%)
ID	Z_c	$T_c(s)$	Z_r	$(Z_r - Z_c)/Z_c$
5-2-2-2-1	99.3*	237.09	112.4	13.19%
5-2-2-2-2	101.7*	259.48	119.6	17.60%
5-2-2-2-3	123.7*	288.62	138.4	11.88%
5-2-2-2-4	99.0*	264.91	113.0	14.14%
5-2-2-2-5	127.1*	301.11	154.3	21.40%
8-2-2-2-1	113.9*	3561.47	131.6	15.54%
8-2-2-2-2	130.6*	3711.32	150.7	15.39%
8-2-2-2-3	149.6*	3288.66	166.4	11.23%
8-2-2-2-4	115.1*	3789.91	140.7	22.24%
8-2-2-2-5	150.8*	3827.34	183.4	21.62%
10-2-2-2-1	178.1	7200.00	201.9	13.36%
10-2-2-2-2	188.9	7200.00	211.4	11.91%
10-2-2-2-3	201.4	7200.00	248.6	23.44%
10-2-2-2-4	180.3	7200.00	196.7	9.10%
10-2-2-2-5	208.9	7200.00	234.8	12.40%
Avg.	144.6	3701.99	166.9	15.63%

Note: '*' represents the optimal solution obtained by CPLEX.

Table 5. Performance of the IPSO for small-scale settings.

Instance	CPLEX	LB	IPSO		LB Gap(%)	Obj Gap (%)	Time ratio
ID	Z_c	Z_l	Z_p	$T_p(s)$	$(Z_p - Z_l)/Z_l$	$(Z_p - Z_c)/Z_c$	T_p/T_c
5-2-2-2-1	99.3*	97.6	99.3*	0.91	1.74	0	0.38%
5-2-2-2-2	101.7*	101.7	101.7*	0.88	0	0	0.34%
5-2-2-2-3	123.7*	111.5	123.7*	0.93	3.51	0	0.32%
5-2-2-2-4	99.0*	95.6	99.0*	1.01	3.56	0	0.38%
5-2-2-2-5	127.1*	120.8	127.1*	0.95	5.22	0	0.32%
8-2-2-2-1	113.9*	108.2	113.9*	19.11	5.27	0	0.54%
8-2-2-2-2	130.6*	122.6	130.6*	18.88	6.53	0	0.51%
8-2-2-2-3	149.6*	145.9	149.6*	19.13	2.54	0	0.58%
8-2-2-2-4	115.1*	109.8	115.1*	16.70	4.83	0	0.44%
8-2-2-2-5	150.8*	146.6	150.8*	18.11	2.86	0	0.47%
10-2-2-2-1	178.1	174.0	178.1	33.99	2.36	0	0.47%
10-2-2-2-2	188.9	182.6	188.9	38.62	3.45	0	0.54%
10-2-2-2-3	201.4	193.1	201.4	31.09	4.30	0	0.43%
10-2-2-2-4	180.3	173.6	180.3	35.27	3.86	0	0.49%
10-2-2-2-5	208.9	204.8	208.9	40.31	2.00	0	0.56%
Avg.	144.6	138.8	144.6	18.39	3.47	0	0.45%

Note: '*' represents the optimal solution obtained by CPLEX.

instances as CPLEX; for the rest five small-scale instances, IPSO can get the same solution values obtained by CPLEX. In addition, IPSO is significantly faster.

5.3. Medium-scale settings

The proposed model becomes intractable for CPLEX to solve directly when the number of customers in the situations is up to '10'. We tested a total of 15 instances for 15, 20, and 25 customers, and the ID explanation is the same as above. Table 6 shows the performance of the IPSO in medium-scale situations. For the effectiveness of the comparison algorithm, we use CPLEX to find the lower bound (LB) of the model and then compare the LB gap $= (Z_p - L_p)/L_p$ (Zhen 2016). From the results, we can see that the average LB gap is equal to 3.71%, which is similar to the small-scale experiment LB gap. The average solution time is equal to 141.5 (less than three minutes); the IPSO can obtain a relatively good solution in a relatively short time period. The IPSO performs well in medium-scale settings.

Table 6. Performance of the IPSO for medium-scale settings.

Instance	LB	IPSO		Decision rule	LB gap (%)	R Gap(%)
ID	L_p	Z_p	$T_p(s)$	Z_r	$(Z_p - L_p)/L_p$	$(Z_r - Z_p)/Z_r$
15-2-2-2-1	288.4	299.9	72.1	346.7	3.99	15.61%
15-2-2-2-2	301.4	308.9	69.7	380.5	2.49	23.18%
15-2-2-2-3	375.9	389.4	77.3	461.4	3.59	18.49%
15-2-2-2-4	299.8	310.3	80.1	394.5	3.50	27.14%
15-2-2-2-5	401.2	407.8	67.1	498.4	1.65	22.22%
20-2-2-2-1	420.5	445.5	162.9	560.4	8.53	25.79%
20-2-2-2-2	461.6	467.5	166.0	572.1	1.28	22.37%
20-2-2-2-3	479.6	502.9	171.9	612.3	4.86	21.75%
20-2-2-2-4	441.3	467.2	148.4	568.7	5.87	21.73%
20-2-2-2-5	501.6	510.9	159.8	622.6	1.85	21.86%
25-3-3-3-1	573.2	598.5	188.6	703.5	4.41	17.54%
25-3-3-3-2	581.9	611.9	190.6	741.9	5.16	21.25%
25-3-3-3-3	664.2	690.3	180.3	811.7	3.93	17.59%
25-3-3-3-4	612.0	633.7	191.6	767.3	3.55	21.08%
25-3-3-3-5	703.8	711.4	196.3	881.3	1.08	23.88%
Avg.	473.1	490.4	141.5	594.9	3.71	21.43%

5.4. Large-scale settings

In this section, we continue to test the performance of IPSO in large-scale settings. In the large-scale experiment, we tested 30–100 customers in increments of 5. We only tested one instance with the same number of customers; thus, we cancelled the index part in the instance ID, and the other parts are the same as described above, including the number of customers, the number of vehicles, the number of charging stations and the number of gas stations. There are four modes in all instances. To illustrate the practical application significance of the proposed algorithm, we conduct experiments comparing an intuitive decision rule and our algorithm. This practical rule is intuitive and commonly used by practitioners in reality.

Table 7 is the comparison of the IPSO and decision rule in large-scale settings; it has an average gap of 28.32%, which illustrates the practical application significance of the proposed algorithm. The results also show that IPSO can be solved for 100 customers with an average time of 1063.1 s.

In large-scale experiments, for comparison with other metaheuristics, we have chosen the standard genetic algorithm (GA) (Prins 2004; Cattaruzza, Absi, and Feillet 2013) which is commonly used in solving VRPs. We replaced the iterative mechanism in IPSO with the evolution mechanism of GA and keep the LS/VNS mechanism and label procedure. The result

Table 7. Comparison of the IPSO and the decision rule in large-scale settings.

Instance	Decision rule	IPSO		R Gap (%)
ID	Z_r	Z_p	$T_p(s)$	$(Z_r - Z_p)/Z_r$
30-3-3-3	1225.2	762.6	221.4	37.75
35-4-4-4	1456.7	1198.4	319.8	17.73
40-4-4-4	1922.0	1415.9	368.9	26.33
45-5-5-5	2563.3	1956.1	457.9	23.69
50-5-5-5	2897.5	2130.7	566.6	26.46
55-6-6-6	3441.8	2582.5	714.6	24.97
60-6-6-6	3728.7	2791.3	814.5	25.83
65-7-7-7	4122.6	3012.8	886.9	26.36
70-7-7-7	4818.2	3442.7	1010.3	28.55
75-8-8-8	5121.8	3599.6	1238.2	29.72
80-8-8-8	5588.0	3772.3	1327.8	32.49
85-9-9-9	5944.5	4121.0	1488.3	30.68
90-10-10-10	6699.7	4589.2	1956.6	31.50
95-11-11-11	7421.9	5011.4	2167.7	32.48
100-11-11-11	8083.6	5642.8	2406.9	30.19
Avg.	4335.7	3068.6	1063.1	28.32

Table 8. Performance of the IPSO and the GA for large-scale instances.

Instance	IPSO		GA		Gap (%)	Time ratio
ID	Z_p	$T_p(s)$	Z_g	$T_g(s)$	$(Z_g - Z_p)/Z_p$	T_g/T_p
30-3-3-3	762.6	221.4	762.6	350.6	0.00%	158.36%
35-4-4-4	1198.4	319.8	1198.4	490.5	0.00%	153.38%
40-4-4-4	1415.9	368.9	1433.6	690.5	1.25%	187.18%
45-5-5-5	1956.1	457.9	1950.7	880.1	− 0.28%	192.20%
50-5-5-5	2130.7	566.6	2144.5	920.4	0.65%	162.44%
55-6-6-6	2582.5	714.6	2588.1	1208.3	0.22%	169.09%
60-6-6-6	2791.3	814.5	2790.5	1533.4	− 0.03%	188.26%
65-7-7-7	3012.8	886.9	3033.4	1618.3	0.68%	182.47%
70-7-7-7	3442.7	1010.3	3455.1	1853.7	0.36%	183.48%
75-8-8-8	3599.6	1238.2	3602.9	1957.6	0.09%	158.10%
80-8-8-8	3772.3	1327.8	3768.4	2133.2	− 0.10%	160.66%
85-9-9-9	4121.0	1488.3	4145.6	2251.4	0.60%	151.27%
90-10-10-10	4589.2	1956.6	4622.3	2994.8	0.72%	153.06%
95-11-11-11	5011.4	2167.7	5005.9	3314.5	− 0.11%	152.90%
100-11-11-11	5642.8	2406.9	5648.7	3936.7	0.10%	163.56%
Average	3068.6	1063.1	3076.7	1742.3	0.28%	167.76%

is illustrated in Table 8, where the GA's objective value 'Z_g', the computation time 'T_g', Gap $= (Z_g - Z_p)/Z_p$, Time ratio $= (T_g/T_p)$ are provided.

Observing the results in Table 8, both IPSO and GA can also obtain the feasible solutions for large-scale instances. The difference between the solutions obtained by the two algorithms is small, but the calculation speed of IPSO is much faster than GA. To sum up, the above observation can indicate that IPSO and GA are appropriate for solving the formulated model, yet from the calculation time, IPSO is more suitable to solve this problem.

5.5. The effect of using HEVs

With the objective of minimizing total travel cost of energy consumption, the numerical experiments in this section are used to compare the effects of using HEVs with mode selection versus conventional vehicles. A conventional vehicle uses

Table 9. Comparison of using HEVs and conventional vehicles.

Instance	Hybrid electric	Conventional	Gap(%)
ID	Z_H	Z_C	$(Z_C - Z_H)/Z_H$
5-2-2-2	99.3	161.5	38.51%
10-2-2-2	178.1	313.2	43.14%
15-2-2-2	299.9	491.3	38.96%
20-2-2-2	445.5	765.8	41.83%
25-3-3-3	598.5	1034.7	42.16%
30-3-3-3	762.6	1369.9	44.33%
35-4-4-4	1198.4	2136.4	43.91%
40-4-4-4	1415.9	2889.3	51.00%
45-5-5-5	1956.1	3561.8	45.08%
50-5-5-5	2130.7	4023.7	47.05%
55-6-6-6	2582.5	4516.9	42.83%
60-6-6-6	2791.3	5368.1	48.00%
65-7-7-7	3012.8	5422.5	44.44%
70-7-7-7	3442.7	6312.9	45.47%
75-8-8-8	3599.6	7278.4	50.54%
80-8-8-8	3772.3	7333.1	48.56%
85-9-9-9	4121.0	7523.9	45.23%
90-10-10-10	4589.2	7812.5	41.26%
95-11-11-11	5011.4	8577.4	41.57%
100-11-11-11	5642.8	9865.2	42.80%
Avg.	2382.53	4337.9	44.33%

gasoline as the sole energy source. The comparison is tested on 20 instances with customer number increasing from 5 to 100, and the ID explanation is as stated in Section 5.2.

Table 9 shows the travel cost of energy consumption results of using HEVs (denoted by 'Z_H') and conventional vehicles (denoted by 'Z_C'). The cost results of the conventional vehicle are bigger than those for the HEV for all instances, where the average gap is equal to 44.33%. The benefit of using HEV is clear, it offers a much lower cost compared to the conventional vehicle.

Based on the above experimental results, some insights can be suggested both from the model and the experiments: (1) The HEVRP with mode selection can help logistic companies reduce their total travel cost by simultaneously determining the optimal route and the selection of modes on each road segment. However, the decision-making process is very complicated and time-consuming. The proposed IPSO is proved to be a powerful tool, both in terms of efficiency and effectiveness, to address the HEVRP with mode selection; (2) The adopted decision rule always uses the least-cost battery-based mode. Compared with our model, the results obtained by decision rule is relatively poor, and as the scale increases, the gap also increases. It reflects the outperformance of mode selection; (3) Moreover, this study also provides encouragement for logistic companies to replace their conventional vehicles with HEVs.

6. Conclusions

HEVs have the advantages of low energy consumption, low pollution, etc. and have a positive effect on solving the adverse impact on cities of urban distribution. Therefore, our paper combines the practical application of HEVs in the logistics industry and studies the HEVRP with mode selection, which minimizes total energy consumption cost. The contributions of this study are listed as follows: first, this paper makes an explorative study on a variant of VRP by considering battery capacity constraints, gasoline constraints and mode selection, which are seldom considered simultaneously in existing studies. Moreover, the HEVRP with mode selection is formulated as a mixed integer linear programming model to minimize the total energy consumption cost. For solving the proposed model in an efficient manner, we design an improved particle swarm algorithm (IPSO) in which a label procedure is involved. Finally, we conduct numerical experiments to test the efficiency of the IPSO. The experimental results show that our proposed algorithm not only obtains the optimal solution for some small-scale problem settings but also solves some large-scale problems (one hundred customers, eleven vehicles, eleven charging stations, eleven gas stations and four modes) within an hour.

The modern logistics industry plays an important role in many areas, such as product manufacturing (Eslami et al. 2018), supply chain (Dolgui et al. 2017), transportation (Simon and Zehtabian 2018), etc. HEVRP with mode selection can help reduce travel cost in these real-world production activities. This study proposes a discussion for a new variation of HEVRP, which considers the mode selection. Experimental results show the profitable use and a broad application field of HEVs, especially the ones with mode selection system. The problem of interest is extremely complicated, it simultaneously involves route optimization, recharging-refueling decision, and mode selection. Therefore, an efficient solution method is absolutely necessary to ensure that HEVs with mode selection is applicable in real life. The proposed method is proved to be a suitable choice. In the future research, we can further explore the possibility of partial battery recharging. Moreover, some stochastic events can be considered, such as uncertainty of demand and uncertainty of travel time due to road congestion.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Natural Science Foundation of China [grant numbers 71831008, 71671107, 71422007]. Thanks are due to the reviewers for their valuable comments.

ORCID

Lu Zhen http://orcid.org/0000-0001-5209-1109
Liyang Xiao http://orcid.org/0000-0003-4027-7701

References

Ángel, F., M. T. Ortuño, G. Righini, and G. Tirado. 2014. "A Heuristic Approach for the Green Vehicle Routing Problem with Multiple Technologies and Partial Recharges." *Transportation Research Part E Logistics & Transportation Review* 71: 111–128.

Baldacci, R., N. Christofides, and A. Mingozzi. 2008. "An Exact Algorithm for the Vehicle Routing Problem Based on the set Partitioning Formulation with Additional Cuts." *Mathematical Programming* 115 (2): 351–385.

Baldacci, R., E. Hadjiconstantinou, and A. Mingozzi. 2017. "An Exact Algorithm for the Capacitated Vehicle Routing Problem Based on a two-Commodity Network Flow Formulation." *Operations Research* 52 (5): 723–738.

Berger, J. 2003. "A new Hybrid Genetic Algorithm for the Capacitated Vehicle Routing Problem." *Journal of the Operational Research Society* 54 (12): 1254–1262.

Cattaruzza, D., N. Absi, and D. Feillet. 2013. "The Multi-Trip Vehicle Routing Problem with Time Windows and Release Dates." *Transportation Science* 50 (2): 177–185.

Croes, G. A. 1958. "A Method for Solving Traveling Salesman Problems." *Operations Research* 6 (6): 791–812.

Dantzig, G. B., and J. H. Ramser. 1959. "The Truck Dispatching Problem." *Management Science* 6 (1): 80–91.

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

Dolgui, A., M. K. Tiwari, Y. Sinjana, S. K. Kumar, and Y. J. Son. 2017. "Optimizing Integrated Inventory Policy for Perishable Items in a Multi-Stage Supply Chain." *International Journal of Production Research* 56 (1-2): 902–925.

Erdoğan, S., and M. Elise. 2012. "A Green Vehicle Routing Problem." *Transportation Research Part E: Logistics and Transportation Review* 48 (1): 100–114.

Eslami, Y., M. Dassisti, M. Lezoche, and H. Panetto. 2018. "A Survey on Sustainability in Manufacturing Organizations: Dimensions and Future Insights." *International Journal of Production Research*, doi:10.1080/00207543.2018.1544723.

Geetha, S. 2014. "Analysis of Vehicle Routing Problem Using Various Particle Swarm Optimization Techniques." *International Journal of Remote Sensing* 24 (22): 4251–4266.

Hansen, P., J. Lazić, and N. Mladenović. 2007. "Variable Neighbourhood Search for Colour Image Quantization." *Ima Journal of Management Mathematics* 18 (2): 207–221.

Hiermann, G., J. Puchinger, S. Ropke, and R. F. Hartl. 2016. "The Electric Fleet Size and mix Vehicle Routing Problem with Time Windows and Recharging Stations." *European Journal of Operational Research* 252 (3): 995–1018.

Kennedy, J., and R. Eberhart. 1995. "Particle swarm optimization." Paper presented at the IEEE international conference on neural networks, the University of Western Austrlia, 27 November-1 December.

Laporte, G. 2009. "Fifty Years of Vehicle Routing." *Transportation Science* 43 (4): 408–416.

Mancini, S. 2017. "The Hybrid Vehicle Routing Problem." *Transportation Research Part C: Emerging Technologies* 78: 1–12.

Montoya, A., C. Guéret, J. E. Mendoza, and J. G. Villegas. 2017. "The Electric Vehicle Routing Problem with Nonlinear Charging Function." *Transportation Research Part B Methodological* 103: 87–110.

Nejad, M., L. Mashayekhy, D. Grosu, and R. Chinnam. 2016. "Optimal Routing for Plug-in Hybrid Electric Vehicles." *Transportation Science* 51 (4): 1304–1325.

Prins, C. 2004. "A Simple and Effective Evolutionary Algorithm for the Vehicle Routing Problem." *Computers & Operations Research* 31 (12): 1985–2002.

Prins, C., N. Labadi, and M. Reghioui. 2009. "Tour Splitting Algorithms for Vehicle Routing Problems." *International Journal of Production Research* 47 (2): 507–535.

Schneider, M., A. Stenger, and D. Goeke. 2014. "The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations." *Transportation Science* 48 (4): 500–520.

Simon, E., and S. Zehtabian. 2018. "Scheduling Direct Deliveries with Time Windows to Minimise Truck Fleet Size and Customer Waiting Times." *International Journal of Production Research*, doi:10.1080/00207543.2018.1470696.

Solomon, M. M. 1987. "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints." *Operations Research* 35 (2): 254–265.

Stenger, A., D. Vigo, S. Enz, and M. Schwind. 2013. "An Adaptive Variable Neighborhood Search Algorithm for a Vehicle Routing Problem Arising in Small Package Shipping." *Transportation Science* 47 (1): 64–80.

Toth, P., and D. Vigo. 2014. *Vehicle Routing: Problems, Methods, and Applications*. Philadelphia: MOS-SIAM Series on Optimization.

Wu, Y., S. Zhang, J. Hao, H. Liu, X. Wu, J. Hu, P. W. Michael, J. W. Timothy, K. Z. Max, and S. Svetlana. 2017. "On-road Vehicle Emissions and Their Control in China: a Review and Outlook." *Science of the Total Environment* 574: 332–349.

Yu, V. F., A. A. N. P. Redi, Y. A. Hidayat, and O. J. Wibowo. 2017. "A Simulated Annealing Heuristic for the Hybrid Vehicle Routing Problem." *Applied Soft Computing* 53: 119–132.

Zhen, L. 2015. "Tactical Berth Allocation Under Uncertainty." *European Journal of Operational Research* 247 (3): 928–944.

Zhen, L. 2016. "Modeling of Yard Congestion and Optimization of Yard Template in Container Ports." *Transportation Research Part B: Methodological* 90: 83–104.