Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6635749, 10 pages
https://doi.org/10.1155/2021/6635749

Research Article

An Adaptive Memetic Algorithm for Dynamic Electric Vehicle Routing Problem with Time-Varying Demands

Na Wang,[1] Yihao Sun,[2] and Hongfeng Wang[2]

[1]*Department of Basic Computer and Mathematics, Shenyang Normal University, Shenyang 110034, China*
[2]*College of Information Science and Engineering, Northeastern University, Shenyang 110819, China*

Correspondence should be addressed to Hongfeng Wang; hfwang@mail.neu.edu.cn

Received 29 October 2020; Revised 21 January 2021; Accepted 22 February 2021; Published 4 March 2021

Academic Editor: Kai Wang

Dynamic electric vehicle routing problem (DEVRP) is an extension of the electric vehicle routing problem (EVRP) into dynamic logistical transportation system such that the demand of customer may change over time. The routing decision of DEVRP must concern with the driving range limitation of electric vehicle (EV) in a dynamic environment since both load degree and battery capacity are variable according to the time-varying demands. This paper proposes an adaptive memetic algorithm, where a special encoding strategy, an adaptive local search operator, and an economical random immigrant scheme are employed in the framework of evolutionary algorithm, to solve DEVRP efficiently. Numeric experiments are carried out upon a series of test instances that are constructed from a stationary VRP benchmark. The computational results show that the proposed algorithm is more effective in finding high-quality solution than several peer algorithms as well as significant in improving the capacity of the routing plan of EVs in dynamic transportation environment.

1. Introduction

In recent years, electric vehicles (EVs) have begun to widely apply into the logistical transportation systems due to their advantages of energy consumption and low pollution [1]. For example, there are about 2000 EVs, accounting for 10% in total, which are operated by the French electricity distribution company ENEDIS in 2016. However, EVs usually have a short driving range of 100–150 miles due to the technology bottleneck of battery [2]. Furthermore, the energy consumption of EVs is varied with the load degree, the speed, and even the road slope that may cause their driving range to be significantly lower [3]. In reality, the logistical transportation systems are often dynamic. For example, the request of a new customer arrives dynamically or the demand of an old customer can change over time. When deciding the route plan of an EV fleet in these dynamic environments, the influence of dynamic load degree upon the driving range must be considered due to the time-varying delivery requests. Figure 1 illustrates the route execution of a single EV in a dynamic logistical transportation system. Before an EV leaves the depot O time t_0, an initial route plans to visit the currently known requests (A, B, C, D, E). While EV executes its route, a new request X arrives at time t_1 and the initial route is adjusted to fulfill it. However, the increased load by new arrival request X limits the driving range of EV that causes its inevitable return to depot O once for recharging. Finally, the real executed route of EV is (A, B, C, X, O, D, E). Therefore, the routing plan of EVs in dynamic logistical transportation system becomes distinctively different from the traditional vehicle routing problem (VRP) in the literature as we will demonstrate throughout this paper.

In this paper, a special dynamic electric VRP (DEVRP) is investigated for the optimal routing plan of a fleet of EVs so that they can serve a set of customers with time-varying demands while minimizing the total driving distance of EVs. Obviously, DEVRP can be regarded as an extension of electric VRP (EVRP) into dynamic transport environment. Recently, a lot of researchers have begun to focus on EVRPs due to the wide application of EVs in the logistical transportation field [4–9]. Schneider et al. [10] studied an EVRP with time windows and changing stations in order to minimize the total travel distance by a homogenous EV fleet. Felipe et al. [11] proposed

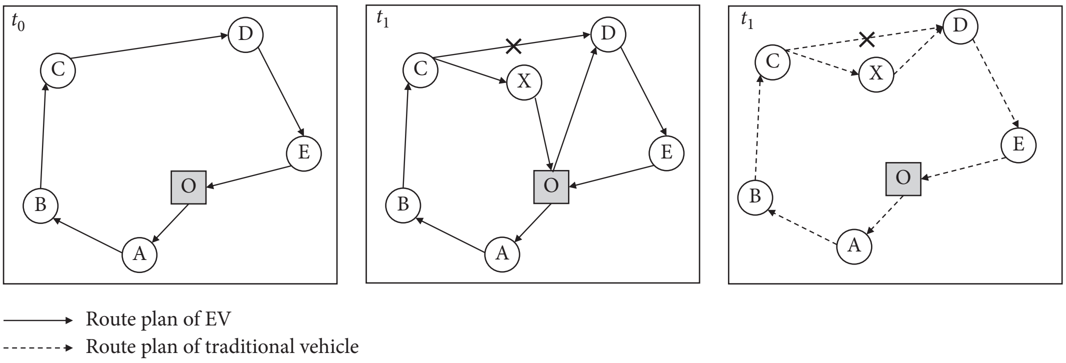

Figure 1: Example of dynamic EV routing.

several heuristics that were constructed within the framework of simulated annealing algorithm for an EVRP with multiple charging technologies and partial recharges. Goeke and Schneider [12] considered a full charging policy with a linear charging function approximation in the routing problem of a mixed electric and conventional vehicle fleet. Lin et al. [13] discussed the effect of vehicle load on battery consumption in EVRP. Keskin and Çatay [14] investigated the application of an adaptive large neighborhood search algorithm for an EVRP that allowed partial recharging. Montoya et al. [15] designed a hybrid metaheuristic for EVRP, in which the battery-charge level was defined as a nonlinear function of the charging time. Zhang et al. [16] utilized an ant colony algorithm for solving an EVRP with the objective of minimizing the energy consumption of EVs. Jie et al. [17] presented a two-echelon capacitated EVRP with battery swapping stations and utilized a hybrid algorithm that combined column generation and adaptive large neighborhood search to determine the delivery strategy under battery consumption limitations. It is noticeable that the current relevant literature on EVRP were concerned with the recharging operations of EVs along their route or the utilization efficiency of battery during the transport course. However, most researches only considered EVRPs as static optimization problems; that is, all customer demands were given as constant. As shown by the example in Figure 1, rerouting an EV must consider the influence of its driving range with load degree once a new customer demand arrives, which is not involved if a traditional vehicle is used.

According to the method proposed by Pillac et al. [18], there are four categories of VRPs. The first is the static and deterministic VRP, where all input is known beforehand and vehicle routes do not change once they are in execution [19]. The second is the static and stochastic VRP, which is characterized by input partially known as random variables (Gendreau et al.) [20]. The third is the dynamic and deterministic VRP, where part or all of the input is unknown and revealed dynamically during the design or execution of the routes [21]. The final is the dynamic and stochastic VRP that has part or all of the input unknown and revealed the exploitable stochastic knowledge dynamically during the execution of the routes [22].

The investigated problem in this paper belongs to the third category; however, the routing algorithm of DEVRP is significantly complicated due to the influence of EVs' driving range varying with their load degree in dynamic environment. For DEVRP, an effective optimization algorithm should track these environmental changes and adapt the best routing scheme of an EV fleet to the changes accordingly. Therefore, we will investigate the applications of an adaptive memetic algorithm (MA) for DEVRP in this paper. The contributions of this study can be summarized as follows:

(i) We extend EVRP into dynamic delivery transport environment, which is more general and practical, and present a special DEVRP, where the driving range of EV is limited by the variable load degree according to the time-varying demand requests.

(ii) We propose an adaptive MA method to solve the investigated DEVRP. The proposed method introduces a special individual representation strategy, an adaptive local search (LS) operator, and an economical random immigrant scheme into the framework of evolutionary algorithm (EA).

(iii) We validate the performance of the proposed method using a series of test instances constructed from a stationary VRP benchmark.

The rest of this paper is outlined as follows. Section 2 formally provides the investigated DEVRP. Section 3 describes the proposed MA method in detail. Section 4 constructs DEVRP test suites from a stationary VRP benchmark and evaluates empirically the performance of the proposed MA for DEVRPs. The final section concludes this paper with discussions on future works.

2. Problem Definition

In this section, we provide a formal description of the DEVRP, in which a homogenous fleet of EVs start from a single depot and deliver the goods to a set of customers. Each EV has a fixed load capacity and limited driving range. While EV is traveling, the battery charge level decreases proportionally with the distance traversed and the current load degree. All EVs must return to the depot for the battery recharging. The demands of customers may change during the design stage. As a consequence, the delivery routes may have to be revised to accommodate the corresponding changes of demands.

Let $I = \{1, \ldots, n\}$ be the set of nodes representing the customers and 0 a node representing the depot. Each customer has a time-varying demand q_i^t over the design horizon $[t, T]$ with $t > 0$. The DEVRP is defined on a directed and complete graph $G = (V, A)$, where $V = \{0\} \cup I$ and $A = \{(i, j): i, j \in V, i \neq j\}$ represents the set of arcs connecting vertices of V. Each arc (i, j) has two associated nonnegative values: a travel distance d_{ij} and a battery consumption b_{ij}. All EVs have a battery of capacity Q and a maximal load capacity $C_{\max}$. Note that b_{ij} can be calculated by the following formulation:

$$b_{ij} = \frac{q_{\text{cur}} - q_{\text{emp}}}{q_{\text{ful}} - q_{\text{emp}}} \cdot e_f \cdot d_{ij}, \quad (1)$$

where e_f represents the engine efficiency of EV, and q_{cur}, q_{ful}, and q_{emp} represent the load degrees with the current, full, and empty statuses of EV on arc (i, j), respectively.

The additional assumptions in the investigated DEVRP are given as follows:

(1) Each EV starts with the full-battery capacity from the depot

(2) Each EV route ends at the depot

(3) Each customer node is visited once by only an EV route

(4) Driving speed on each arc is constant

(5) The battery is recharged to full each time after returning to the depot

(6) No time window constraint is considered for the delivery service and the battery recharging

The purpose of DEVRP investigated in this paper is to decide an optimal delivery route plan so as to minimize the total travel distance of EVs as well as adapting to the time-varying demands of customers during the design stage. Considering that the traditional exact or approximation methods for large-scale VRPs are not very exciting, an adaptive memetic algorithm will be proposed for solving this problem efficiently and the detailed algorithmic designs will be given in the next section.

3. Proposed MA Method for DEVRP

3.1. General Framework of MA. In the EA community, MA can be regarded as a hybrid metaheuristic method inspired by Darwinian principles of natural evolution and Dawkins' notion of a meme, defined as a unit of cultural evolution that is capable of local refinements. Within the framework of a MA, EA operators are responsible for global rough search and LS operators (an LS is also called a meme) are used for local refinement. Due to the advantage of maintaining an efficient balance between exploration and exploitation, MAs have been successfully used to solve a lot of complex optimization problems [23–26]; however, they are rarely considered for dynamic optimization problems [27, 28].

The proposed MA in this paper is a class of EA-based hybrid metaheuristic, which can be expressed by the pseudocode in Figure 2, where pop_size, *pc*, and *pm* are the population size, crossover probability, and mutation probability, respectively. Within this MA, a population of pop_size chromosomes are generated randomly and then evaluated at the initialization step. Then, an elite chromosome, i.e., the chromosome with the best fitness, is improved by an LS operator. At each subsequent generation, the chromosomes undergo a binary tournament selection, where two chromosomes are selected randomly from the current population and the best one is taken for one parent, and the selected chromosome undergoes an order crossover (OX) operation with a probability *pc*. After crossover is executed, a 2-OPT mutation operator is performed for each newly generated offspring chromosome with a probability *pm*. Then, the pop_size best chromosomes among all the parents and offspring are selected to proceed into the next generation, and an elite chromosome in the newly generated population is refined in the LS operation. If the immigrant scheme is used, a certain number of immigrants are generated randomly and replace the worst chromosomes in the current population in order to improve the exploration capacity of MA in dynamic environment.

```
begin
  parameterize ($pop\_size$, $pc$, $pm$);
  $t = 0$;
  initializePopulation ($P(0)$);
  evaluatePopulation ($P(0)$);
  if the LS method is used then
    $elite$ = selectForLocalSearch ($P(0)$);
    execute the LS operation upon $elite$;
  repeat
    $P'(t)$ = selectForReproduction ($P(t)$);
    $P''(t)$ = crossover ($P'(t)$);
    mutate ($P''(t)$);
    evaluatePopulation ($P''(t)$);
    $P\ (t+1)$ = selectForSurvival ($P''(t) \cup P(t)$);
    if the LS method is used then
      $elite$ = selectForLocalSearch ($P(t)$);
      execute the LS operation upon $elite$;
    if the immigrant scheme is used then
      immigrate a number of random chromosomes into $P\ (t+1)$;
    $t = t + 1$;
  until a stop condition is met
end.
```

Figure 2: Framework of the proposed MA for DEVRP.

3.2. Individual Representation. How to express a chromosome using a proper encoding method is a basic work when designing an MA for VRP. A good representation scheme can help the algorithm obtain high-quality solutions easily, while an improper representation may make it hard for an algorithm to achieve even feasible solutions.

Prins [29] proposed a simple and effective encoding scheme of GA for VRP, where each chromosome is defined as a sequence of N customer nodes, just like in most GAs for traveling salesman problem (TSP). This encoding scheme can be viewed as a giant tour performed by one vehicle of infinite capacity if only the same vehicle performs all trips one by one. Then, an optimal split procedure can partition a given chromosome into the best VRP solution. Based on this encoding mechanism, we develop a permutation encoding scheme in MA for the investigated DEVRP. Note that the evaluation of a chromosome is time-costing since the splitting procedure, as shown in Figure 3, is known as an exact algorithm in $O(N)$. Thus, the evaluation number will be considered as the time index in the proposed MA for DEVRP in this paper.

```
$V_0 = 0$;
for $i = 1$ to $n$ do $V_i = +\infty$ endfor
for $i = 1$ to $n$ do
  $load = 0$; $cost = 0$; $dist = 0$; $j = i$;
  repeat
    $load = load + q_j$;
      if $i = j$ then
        $dist = d_{0,s_j} + d_{s_j,0}$; $cost = b_{0,s_j} + b_{s_j,0}$;
      else
        $dist = dist\text{-}d_{s_{j-1},0} + d_{s_{j-1},s_j} + d_{s_j,0}$;
        $cost = cost\text{-}b_{s_{j-1},0} + b_{s_{j-1},s_j} + b_{s_j,0}$;
      enif
      if $(load <= C_{max})$ and $(cost <= Q)$ then
        if $V_{i-1} + dist < V_j$ then
          $V_j = V_{i-1} + dist$;
          $P_j = i\text{-}1$;
        endif
        $j = j + 1$;
      endif
   until $(j > n)$ or $(load > C_{max})$ or $(cost > Q)$
endfor
Denotations:
  $S$: a sequence vector of visiting $n$ customers;
  $P$: a label vector of EVs.
```

Figure 3: Algorithm for the splitting procedure.

3.3. Local Search. In many heuristic methods for the routing problems, the inverse operation, where two nodes are selected from a segment which is reversed, is often employed as the local move technique, as shown in Figure 4. The quality of a neighbor generated by executing one inversion can be estimated by a parameter Δ, which is calculated by the formula $\Delta = (ac + b\ d) - (ab + c\ d)$. Obviously, the smaller the value of Δ is, the higher the quality of the new generated solution is.

In the original inversion scheme, the sequence between two different nodes chosen randomly in a single tour is reversed. Since there are existing multiple tours in the VRP solution, the inversion can be divided into two ways: single-tour-based inversion (SI) and multiple-tour-based inversion (MI). The former starts from choosing a single tour and reverses the sequence between two different nodes in the chosen tour. In the latter, the tours are not considered in isolation, and paths and nodes are allowable to exchange between different tours.

An example of SI and MI is presented in Figure 5, where node 3 and node 6 are selected from single (SI) and multiple (MI) tours and the sequence between them is reversed. It is easy to see that SI can always help a solution s make a single-tour local improvement, which means that one local move in SI always occurs in a smaller area around s, and MI can perform one move in a wider range upon s. Obviously, each local move makes a biased search and hence may be efficient for some classes of problems but not for others. Therefore, how to develop an efficient LS operator and avoid utilizing inappropriate local move becomes a very important issue.

Here, we design an adaptive LS (ALS) method for the investigated DEVRP according to an effective learning mechanism [30] in order to utilize SI and MI operators efficiently. In our proposed ALS method, SI and MI are both allowed to work together and selected by probability to generate a neighbor of an individual at every step of an LS operation on every iteration of the algorithm. In order to obtain the advantages of both of them during different periods when they are effective, an adaptive learning strategy is used to adjust their execution probabilities according to the improvement each inversion operator has achieved on every LS step.

Let p_{si} and p_{mi} denote the probabilities of applying SI and MI to generate a neighbor of an individual that is used for local search, respectively; $p_{si} + p_{mi} = 1$. At the start of this strategy, p_{si} and p_{mi} are both set to 0.5, which means giving a fair competition chance to each inversion operator. As each inversion move always makes a biased search, the inversion operator which produces more improvements should be given a greater selection probability. Let η denote the improvement degree of the selected individual when one inversion operator is used to refine it and η can be calculated by

$$\eta = \frac{f_{\text{imp}} - f_{\text{ini}}|}{f_{\text{ini}}}, \tag{2}$$

where f_{imp} is the final fitness of the selected individual for the local refinement after executing one LS operation and f_{ini} is its initial fitness before the local refinement. At each generation, the degree of improvement of each inversion operator is calculated when a predefined number of iterations is achieved, and then p_{si} and p_{mi} are recalculated to proceed with the local improvement in the next generation.

Suppose $\eta_{si}(t)$ and $\eta_{mi}(t)$, respectively, denote the total improvement of SI and MI at generation t. Their selection probabilities $p_{si}(t+1)$ and $p_{mi}(t+1)$ at generation $(t+1)$ can be calculated orderly by the following formulae:

$$\begin{aligned} p_{si}(t+1) &= p_{si}(t) + \delta \cdot \eta_{si}(t), \\ p_{mi}(t+1) &= p_{mi}(t) + \delta \cdot \eta_{mi}(t), \\ p_{si}(t+1) &= \frac{p_{si}(t+1)}{p_{si}(t+1) + p_{mi}(t+1)}, \\ p_{mi}(t+1) &= 1 - p_{si}(t+1), \end{aligned} \tag{3}$$

where δ signifies the relative influence of the degree of the improvement on the selection probability. The proposed

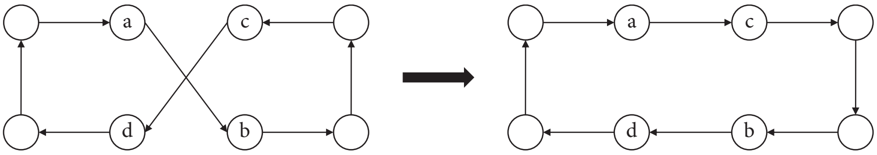

Figure 4: An example of inversion operator.

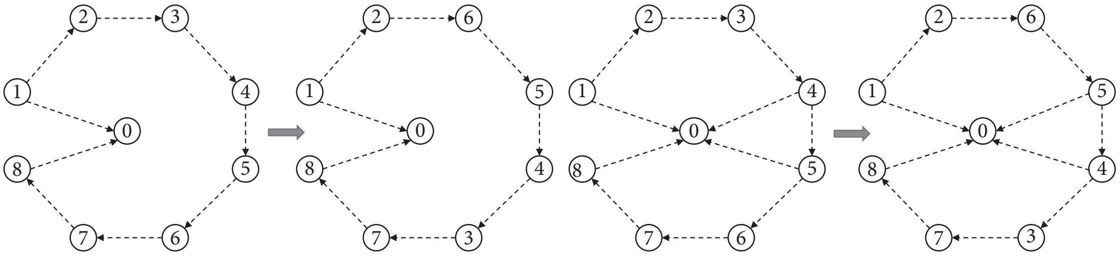

Figure 5: An example of SI and MI operators.

inverse-based LS operator can be expressed by the pseudocode in Figure 6.

From Figure 6, it can be seen that SI and MI are used to generate a certain number of s's neighbors, which compose a neighbor set S, by probability. For each neighbor s' in S, the corresponding value of $\Delta(s, s\prime)$ is calculated to estimate the effect of this local move. It is noticeable that the quality of s' is only estimated by the value of $\Delta(s, s\prime)$ rather than its fitness $f(s\prime)$, which differs from the general LS method as shown in Figure 1. This is because that the evaluation of fitness is too time-consuming (see the relevant description in Section 2.1), while the total evaluation number in each generation is always limited. Finally, the best neighbor s^* with the smallest value of Δ would be evaluated and replace s if its fitness is lower than $f(s)$.

```
begin
  if $p_{si}$ and $p_{mi}$ are not initialized then
    set $p_{si} = p_{mi} = 0.5$;
  set $\eta_{si} = \eta_{mi} = 0$;
  for $i$=1 to $ls\_size$ do
    set $S = \emptyset$ and $ns = 0$;
    while ($ns \leq max\_ns$) do
      if $random() < p_{si}$ then //SI is selected
        generate a neighbor $s'$ of $s$ using SI operator;
      else // MI is selected
        generate a neighbor $s'$ of $s$ using MI operator;
      calculate $\Delta(s, s')$;
      $S = S \cup \{s'\}$;
      $ns = ns + 1$;
    endwhile
    find a solution $s^*$ with the smallest value $\Delta$ of in $S$;
    evaluate $f(s^*)$;
    if $f(s^*) < f(s)$ then $s = s^*$;
    update $\eta_{si}$ and $\eta_{mi}$;
  endfor
  recalculate $p_{si}$ and $p_{mi}$;
end.
Denotations:
  $max\_ns$: the maximal allowable value of |S| per generation;
  $random()$: a pseudo-random number between 0 and 1;
  $\Delta(s, s')$: the corresponding value of $\Delta$ when $s$ moving to $s'$.
```

Figure 6: Pseudocode for the ALS method.

3.4. Increasing Population Diversity. The major problem when EAs are applied for dynamic optimization problems is that the converging population cannot adapt to the changing environments. The random immigrant scheme, where the population is partially replaced by the new-generated chromosomes at every generation, is a simple and efficient scheme for EAs in dynamic environments since it can introduce a constant diversity into the population [31, 32]. Obviously, it is more helpful to migrate random chromosomes into a converging population than a spread-out one. Thus, it is not necessary to always inject a constant number of random chromosomes into the population at every generation.

Here, we propose an economical random immigrant scheme for the proposed MA, where the immigrant ratio ri of every generation can be calculated based on the value of a population diversity index ξ. The index ξ can be calculated as follows:

$$\xi = \min\left\{\frac{\left|f_{\text{ave}} - f_{\text{best}}\right|}{f_{\text{ave}}}, 1\right\}, \tag{4}$$

where f_{best} and f_{ave} are the best and average fitness among the fitness values of the population, respectively. The index ξ is a fitness-based measurement of population diversity, and it can be seen as a measurement of the convergence degree of an algorithm. If $\xi \longrightarrow 1$, there is a high population diversity and therefore the algorithm is far from convergence; if $\xi \longrightarrow 0$, there is a low population diversity, which means that the convergence is approaching.

With the definition of ξ, the immigrant ratio ri can be calculated by the following formula:

$$ri = \min\{\alpha \cdot (1-\xi)\cdot(ri\_\max - ri\_\min) + ri\_\min, ri\_\max\}, \tag{5}$$

where $ri\_\max$ and $ri\_\min$ are, respectively, the predefined maximum and minimum value of ri and are preset constant to control the decreasing or increasing speed of ri. From this formula, it is easy to understand that less random immigrants could be introduced into the population in the presence of higher diversity (i.e., when $\xi \longrightarrow 1$) as a result of $ri \longrightarrow ri\_\min$. However, when the population is converging (i.e., when $\xi \longrightarrow 0$), $ri \longrightarrow ri\_\max$, which increases the level of random immigrants with a great degree.

4. Experimental Study

4.1. Dynamic Test Environment. In this section, the behaviors of investigated algorithms are examined on a series of DEVRPs which are constructed based on an instance from VRPLIB (http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/). In the basic VRP instance eil51, one depot node and $N = 50$ customer nodes are defined by the points in the plane and the cost for each edge (i, j) is the Euclidean distance between node i and node j. For each EV, the maximal load capacity is set to 160 and the capacity of battery is set to 20 kWh.

According to the generated method of dynamic optimization problems by Wang et al. [33], the dynamics of DEVRPs is defined as follows: every τ generations, the demands of $\rho \times N$ customers change according to a random Gaussian variable. More formally, a change of a single customer can be described as follows:

$$q_i(t) = \min\{\max\{q_{\min}, 15 + 8\cdot N(0,1)\}, q_{\max}\}, \tag{6}$$

where $q_i(t)$ denotes the demand of customer i at time t, $q_{\max} = 160$ and $q_{\min} = 1$ denote the allowable maximum and minimum demand of a customer, respectively, and $N(0,1)$ denotes a normal distribution random number in $(0,1)$. In fact, $q_{\max}$ is equal to the capacity of an EV, while $q_{\min}$ is set to 1 in order to ensure that the demand of each customer is nonnegative. Moreover, the mean demand and variance of all customers is 15 and 8, respectively, in the original data set. According to this constructing method, the parameter τ controls the speed of changes while $\rho \in (0.1, 1.0)$ controls the severity of changes. A bigger ρ means more severe changes while a smaller τ means more frequent changes.

In this study, the change speed parameter τ is set to 50, 100, and 200, respectively, which means that the environment changes very fast, in the moderate speed, and slowly, respectively. The change severity parameter ρ is set to 0.1, 0.5, and 1.0, respectively, in order to examine the performance of algorithms in dynamic environments with different severities: from slight change $\rho = 0.1$ to moderate variation $\rho = 0.5$ to intense change $\rho = 1.0$. In order to study the behavior of algorithms in randomly changing environment, we also set ρ to be a random number uniformly distributed in $(0.0, 1.0)$, i.e., $\rho =$ rand. In total, a series of 12 different dynamic problems are constructed from each stationary test problem. The dynamics parameter settings are summarized in Table 1.

Table 1: The index table for dynamic parameter settings.

τ	Environmental dynamic index			
50	1	2	3	4
100	5	6	7	8
200	9	10	11	12
$\rho \longrightarrow$	0.1	0.5	1.0	rand

4.2. Experimental Design. Experiments are carried out in order to study the major features of our proposed MAs and to compare their performance with several other peer algorithms. The following abbreviations represent the algorithms considered in this paper.

(1) SGA: A simple GA

(2) SGAr: A simple GA with restart from scratch whenever the environment changes

(3) RIGA: SGA with the traditional random immigrants' scheme

(4) MA-Inverse: MA with the simple reverse-based LS operator

(5) MA-Swap: MA with the simple swap-based LS operator

(6) MA-Insert: MA with the simple insert-based LS operator

(7) MA-ALS: MA with our proposed ALS operator

(8) MAr-ALS: MA-ALS with a simple restart scheme

(9) RIMA-ALS: MA with the ALS operator and our proposed economical random immigrant scheme

The following parameters are used in all algorithms: the population size (pop_size) is set to 100 for all MAs, but is set to 120 for SGA, SGAr, and RIGA because the LS operation in MAs may be executed by $ls\_size = 20$ steps per generation. It is easy to see that the total number of evaluations per generation is always 120 for all algorithms. The order crossover probability pc equals 0.8 and the inversion mutation probability pm is set to 0.2 for all GAs and MAs. The specific parameters in our proposed algorithms are set as follows: $\alpha = 1.0$, $ri\_\max = 0.1$, and $ri\_\min = 0.01$ for the random immigrant scheme, while $\Delta = 0.98$, $ls\_size\,1 = 5$, and $ls\_size\,2 = 5$ in the ALS operator.

For each experiment of an algorithm on a test problem, 30 independent runs were executed with the same set of random seeds. For each run of an algorithm on a DOP, 10 environmental changes were allowed and the best-of-generation fitness was recorded per generation. The overall offline performance of an algorithm is defined as the best-of-generation fitness averaged across the number of total runs and then averaged over the data gathering period, as formulated below:

$$\overline{F}_{BG} = \frac{1}{G}\sum_{i=1}^{G}\left(\frac{1}{R}\sum_{j=1}^{R} F_{BG_{ij}}\right), \tag{7}$$

where G is the number of generations i.e., $G = 10\cdot\tau$, $R = 30$ is the total number of runs, and $F_{BG_{ij}}$ is the best-of-generation fitness of generation i of run j.

4.3. Experimental Study on the Effect of LS Operators. In the experimental study on LS operators, we investigate the performance of MAs with different LS operators, with the aim of examining the effect of our proposed operator in Section 3. In particular, five different algorithms, including MA-ALS, MA-Inverse, MA-Swap, MA-Insert, and SGA, are tested on the stationary instance of DEVRP. For each run of an algorithm on each problem, the maximum allowable number of generations was set to 200. The experimental results are shown in Figure 7, where the data were also averaged over 30 runs.

From Figure 7, it can be seen that the performance of four MAs is always better than that of SGA on the stationary EVRP. This is because the LS operators can improve the performance of an algorithm significantly. The similar results have been obtained by many researchers. It is noticeable that the behavior of different MAs is different during the running course. MA-Insert, MA-Inverse, and MA-Swap significantly outperform SGA at the early iterative stage of running the algorithms, while their behavior becomes a little disappointing at the late iterative stage. However, MA-ALS almost performs much better than other algorithms on the whole course of running. This is because three different LS operators are employed and selected to execute local improvements for an individual with an adaptive mechanism in MA-ALS. Obviously, our proposed LS operator, which combines the features of two inverse operation in an adaptive learning strategy, can improve the individual more efficiently than each of the three single LS operators. The result that MA-ALS outperforms significantly other algorithms shows the efficiency of the proposed ALS operator.

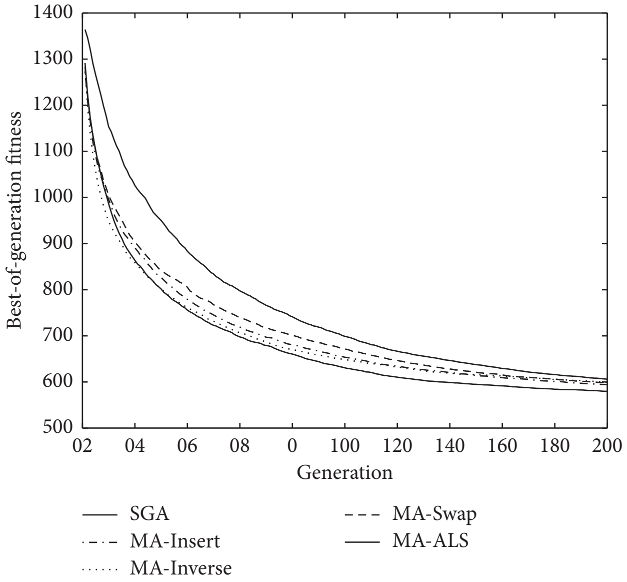

Figure 7: Experimental results of MAs with different LS operators on the stationary problem.

4.4. Experimental Study on the Performance of MAs on DEVRPs. In the experiments on DEVRPs constructed in Table 1, we attempt to compare the performance of our proposed algorithm RIMA-ALS with some other peer algorithms, including SGAr, RIGA, MA-ALS, and MAr-ALS described in Section 4.2.

The experimental results with respect to the overall offline performance are presented in Table 2 and plotted in Figure 8. The corresponding statistical results of comparing algorithms by the one-tailed t-test with 58 degrees of freedom at a 0.05 level of significance are given in Table 3. In Table 3, the t-test results regarding Alg. 1 – Alg. 2 are shown as "s+,", "s−," "+," or "−" when Alg. 1 is significantly better than, significantly worse than, insignificantly better than, or insignificantly worse than Alg. 2, respectively. From Table 2, Table 3, and Figure 8, several results can be observed and are analyzed below.

First, RIMA-ALS significantly outperforms other peer algorithms on most DEVRPs, as indicated in the relevant t-test results in Table 3. This result validates our expectation of the proposed MA for solving DEVRPs. In RIMA-ALS, the ALS operator can make a sufficient exploitation for the best-fitness chromosome. Many researches in the literature often utilize the LS technique to refine each chromosome in the current population. This is too costly or impossible for an MA in dynamic environments considering that the total cost per iteration in terms of evaluations is always limited. The ALS operator is only applied for the best-fitness chromosome in our proposed algorithm and can always help the algorithm keep tracking a changing high-quality solution, which is obviously feasible, especially on real-world applications. Moreover, the population is also introduced a certain degree of diversity by the economical random immigrant scheme, which can help the algorithm adapt well to environmental changes, especially when the environment is subject to significant changes. More relevant analysis on the effect of the operators in RIMA-ALS will be explained in the late experimental analysis.

Second, MAr-ALS performs much worse than MA-ALS and RIGA, but it is better than SGAr on all dynamic problems. Given the perfect restart scheme, MAr-ALS does not reuse any information from the past population, but only restarts to search the equivalent problem from a random initial state when an environmental change occurs. It is easy to understand that MAs or GAs with the restart scheme always take too much cost to reachieve the high-fitness solutions. The result that MAr-ALS significantly underperforms MA-ALS and RIGA on DEVRPs indicates the importance of reusing the useful information in the past population for MA in dynamic environments. The reason why MAr-ALS outperforms SGAr lies in that the ALS operator can help the algorithm obtain the high-fitness solution more quickly, which has been obtained in the experiment in the last section.

Third, MA-ALS always performs well on most DEVRPs, even outperforms RIMA-ALS on DEVRP when $\tau = 200$ and $\rho = 0.1$. This happens because that a new environment is close to the previous one when the value of ρ is very small. It is much advantageous to execute sufficient exploitation for the elite chromosome in the current population when a slight environmental change occurs in terms of severity. When ρ increases to 0.5 or 1.0, MA-ALS is significantly

Table 2: Experimental results of all peer algorithms on DEVRPs.

Dynamics		Algorithms				
τ	ρ	SGAr	RIGA	MA-ALS	MAr-ALS	RIMA-ALS
50	0.1	1026.93 ± 7.05	659.91 ± 15.15	616.16 ± 17.85	872.54 ± 10.28	610.53 ± 11.79
50	0.5	1022.74 ± 8.74	654.29 ± 12.97	616.40 ± 16.78	872.31 ± 10.55	608.07 ± 15.39
50	1.0	1023.24 ± 7.80	660.73 ± 13.88	621.13 ± 14.89	872.15 ± 8.97	612.98 ± 11.96
50	rand()	1022.43 ± 9.44	659.25 ± 20.74	616.70 ± 13.86	871.43 ± 9.34	606.87 ± 12.21
100	0.1	891.90 ± 8.14	613.56 ± 14.70	585.51 ± 15.45	777.13 ± 9.32	584.08 ± 12.20
100	0.5	892.85 ± 7.17	606.41 ± 10.91	589.58 ± 10.59	776.26 ± 6.86	582.60 ± 10.81
100	1.0	892.03 ± 6.72	610.34 ± 9.08	586.30 ± 12.12	771.01 ± 11.20	581.57 ± 11.15
100	rand()	893.47 ± 10.15	607.56 ± 12.51	588.22 ± 10.51	774.04 ± 9.83	583.07 ± 10.67
200	0.1	766.86 ± 8.25	586.05 ± 12.56	566.39 ± 9.74	687.77 ± 8.47	566.78 ± 12.63
200	0.5	765.06 ± 6.99	581.79 ± 13.65	566.90 ± 8.96	688.89 ± 9.67	566.41 ± 13.10
200	1.0	766.04 ± 6.46	582.32 ± 7.53	569.22 ± 7.30	686.49 ± 6.17	564.33 ± 9.02
200	rand()	767.29 ± 6.76	585.65 ± 9.15	567.11 ± 10.42	685.89 ± 8.56	566.82 ± 11.36

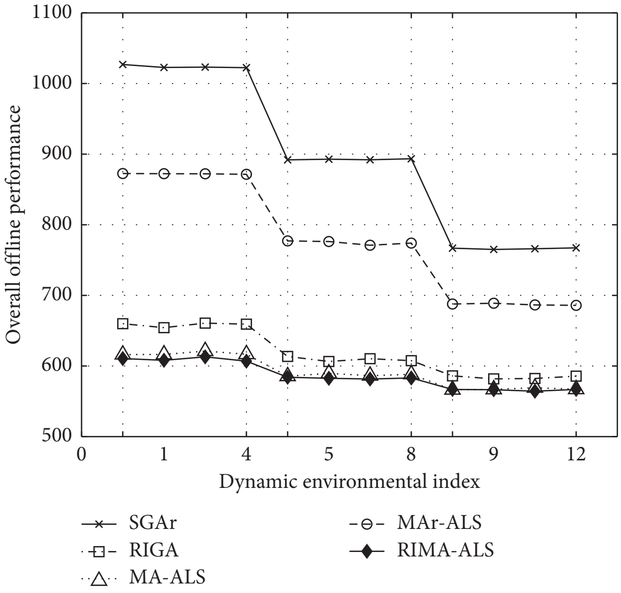

Figure 8: Experimental results of all GAs and MAs on DEVRPs.

Table 3: The t-test results of comparing algorithms regarding the overall offline performance.

t-test result	DEVRP instances			
$\tau = 50$, $\rho \Longrightarrow$	0.1	0.5	1.0	Rand()
RIMA-ALS–MAr-ALS	*s*+	*s*+	*s*+	*s*+
RIMA-ALS–MA-ALS	+	*s*+	*s*+	*s*+
RIMA-ALS–RIGA	*s*+	*s*+	*s*+	*s*+
MAr-ALS–MA-ALS	*s*−	*s*−	*s*−	*s*−
MAr-ALS–RIGA	*s*−	*s*−	*s*−	*s*−
MAr-ALS–SGAr	*s*+	*s*+	*s*+	*s*+
MA-ALS–RIGA	*s*+	*s*+	*s*+	*s*+
RIGA–SGAr	*s*+	*s*+	*s*+	*s*+
$\tau = 100$, $\rho \Longrightarrow$	0.1	0.5	1.0	rand()
RIMA-ALS–MAr-ALS	*s*+	*s*+	*s*+	*s*+
RIMA-ALS–MA-ALS	+	*s*+	*s*+	*s*+
RIMA-ALS–RIGA	*s*+	*s*+	*s*+	*s*+
MAr-ALS–MA-ALS	*s*−	*s*−	*s*−	*s*−
MAr-ALS–RIGA	*s*−	*s*−	*s*−	*s*−
MAr-ALS–SGAr	*s*+	*s*+	*s*+	*s*+
MA-ALS–RIGA	*s*+	*s*+	*s*+	*s*+

Table 3: Continued.

t-test result	DEVRP instances			
RIGA–SGAr	$s+$	$s+$	$s+$	$s+$
$\tau = 200$, $\rho \Rightarrow$	0.1	0.5	1.0	rand()
RIMA-ALS–MAr-ALS	$s+$	$s+$	$s+$	$s+$
RIMA-ALS–MA-ALS	−	$s+$	$s+$	$s+$
RIMA-ALS–RIGA	$s+$	$s+$	$s+$	$s+$
MAr-ALS–MA-ALS	$s-$	$s-$	$s-$	$s-$
MAr-ALS–RIGA	$s-$	$s-$	$s-$	$s-$
MAr-ALS–SGAr	$s+$	$s+$	$s+$	$s+$
MA-ALS–RIGA	$s+$	$s+$	$s+$	$s+$
RIGA–SGAr	$s+$	$s+$	$s+$	$s+$

beaten by RIMA-ALS, which shows the necessity of maintaining the sufficient population diversity for MAs in dynamic environments. Since it is usually not possible to judge the strength and weakness of an environmental change in advance, the proposed random immigrant scheme is a good choice when MAs are applied for DOPs; notice that RIMA-ALS always performs better than MA-ALS on all DEVRPs when ρ = rand.

Finally, the environmental parameters affect the performance of algorithms. The performance of all algorithms increases when the value of τ increases from 50 to 100 to 200. When τ becomes larger, algorithms have more time to find better solutions before the next change. However, the performance of algorithms hardly changes with the increasing or decreasing of the value of ρ.

5. Conclusion

In this paper, an adaptive MA is proposed for addressing a special DEVRP that is characterized by the time-varying customer demands during the route design stage, under the consideration of the influence of the variable load degree upon the driving range of EVs. In the proposed MA method, there are three major algorithmic schemes. The first is to develop a splitting procedure to translate an individual with simple permutation-based representation to a solution of DEVRP. The second is to propose an ALS operator that utilizes two different inverse-based LS operations, that is, SI and MI, in an adaptive cooperation way. The last is to employ an economical random immigrant scheme to maintain a sufficient population diversity for the proposed algorithm to adapt well to the changing environment.

From the experimental results based on a series of DEVRPs, which are constructed from a stationary instance of VRP, the following conclusions can be drawn on the dynamic test problems. Firstly, MAs with the ALS operator enhanced by the suitable diversity technique can exhibit a better performance in solving DEVRPs. For most test problems, RIMA-ALS, which hybridizes GA with the ALS operator and the economical random immigrant scheme, outperforms other peer algorithms. Secondly, the ALS operator can help the algorithm execute a robust local refinement since it employs multiple LS operators via an adaptive cooperation mechanism. In our experiments, MA-ALS performs much better than other MAs with a single LS operator for the stationary instance of VRP. Thirdly, maintaining a sufficient population diversity and reusing the past information efficiently are both important for MAs in dynamic environments. For most DEVRPs, RIMA-ALS always performs better than MA-ALS and MAr-ALS, while MA-ALS outperforms MAr-ALS. Finally, the environmental parameter can affect the performance of algorithms. In our experiments, algorithms perform better with the increase of the frequency of changes, while the influence of the severity of change seems very limited. Generally speaking, the experimental results indicate that the proposed algorithm, where an EA is hybridized with the ALS operator and an economical random immigrant scheme, seems a good optimizer for DEVRPs.

For the future work, it is straightforward to examine the performance of our proposed algorithm for more DEVRP instances. For example, a mixed fleet of EVs or multiple charging stations can be considered in DEVRPs. Another interesting research topic is to investigate the effect of hybridizing the other LS operations, such as those especially used for EVRPs, within this adaptive learning way. In addition, it is also valuable to carry out the sensitivity analysis on the effect of parameters, e.g., α, Δ, etc., on the performance of the proposed algorithm in the future.

Data Availability

The data used to support the findings of this study are available within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant nos. 61703290 and 71671032, the Natural Science Fund Guidance Program of Liaoning Province under Grant no. 2019-ZD-0478, and the Fundamental Research Funds for the Central Universities under Grant no. N180408019.

References

[1] P. R. Kleindorfer, A. Neboian, A. Roset, and S. Spinler, "Fleet renewal with electric vehicles at La Poste," *Interfaces*, vol. 42, no. 5, pp. 465–477, 2012.

[2] S. Pelletier, O. Jabali, and G. Laporte, “Battery electric vehicles for goods distribution: a survey of vehicle technology, market penetration, incentives and practices,” Technical Report CIRRELT-2014-43, CIRRELT, Montreal, Canada, 2014.

[3] C. De Cauwer, J. Van Mierlo, and T. Coosemans, “Energy consumption prediction for electric vehicles based on real-world data,” *Energies*, vol. 8, no. 8, pp. 8573–8593, 2015.

[4] A. Artmeier, J. Haselmayr, M. Leucker, and M. Sachenbacher, “The optimal routing problem in the context of battery-powered electric vehicles,” in *Proceedings of the 2nd International Workshop on Constraint Reasoning and Optimization for Computational Sustainability*, Bologna, Italy, June 2010.

[5] R. G. Conrad and M. A. Figliozzi, “The recharging vehicle routing problem,” in *Proceedings of the 2011 Industrial Engineering Research Conference*, Reno, NV, USA, May 2011.

[6] S. Erdoğan and E. Miller-Hooks, “A green vehicle routing problem,” *Transportation Research Part E*, vol. 48, no. 1, pp. 100–114, 2012.

[7] S. Pelletier, O. Jabali, and G. Laporte, “50th anniversary invited article-goods distribution with electric vehicles: review and research perspectives,” *Transportation Science*, vol. 50, no. 1, pp. 3–22, 2016.

[8] A. Afroditi, M. Boile, S. Theofanis, E. Sdoukopoulos, and D. Margaritis, “Electric vehicle routing problem with industry constraints: trends and insights for future research,” *Transportation Research Procedia*, vol. 3, pp. 452–459, 2014.

[9] G. Hiermann, J. Puchinger, S. Ropke, and R. F. Hartl, “The electric fleet size and mix vehicle routing problem with time windows and recharging stations,” *European Journal of Operational Research*, vol. 252, no. 3, pp. 995–1018, 2016.

[10] M. Schneider, A. Stenger, and D. Goeke, “The electric vehicle-routing problem with time windows and recharging stations,” *Transportation Science*, vol. 48, no. 4, pp. 500–520, 2014.

[11] Á. Felipe, M. T. Ortuño, G. Righini, and G. Tirado, “A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 71, no. 1, pp. 111–128, 2014.

[12] D. Goeke and M. Schneider, “Routing a mixed fleet of electric and conventional vehicles,” *European Journal of Operational Research*, vol. 245, no. 1, pp. 81–99, 2015.

[13] J. Lin, W. Zhou, and O. Wolfson, “Electric vehicle routing problem,” *Transportation Research Procedia*, vol. 12, pp. 508–521, 2016.

[14] M. Keskin and B. Çatay, “Partial recharge strategies for the electric vehicle routing problem with time windows,” *Transportation Research Part C: Emerging Technologies*, vol. 65, pp. 111–127, 2016.

[15] A. Montoya, C. Guéret, J. E. Mendoza, and J. G. Villegas, “The electric vehicle routing problem with nonlinear charging function,” *Transportation Research Part B: Methodological*, vol. 103, pp. 87–110, 2017.

[16] S. Zhang, Y. Gajpal, S. S. Appadoo, and M. M. S. Abdulkader, “Electric vehicle routing problem with recharging stations for minimizing energy consumption,” *International Journal of Production Economics*, vol. 203, pp. 404–413, 2018.

[17] W. Jie, J. Yang, M. Zhang, and Y. Huang, “The two-echelon capacitated electric vehicle routing problem with battery swapping stations: formulation and efficient methodology,” *European Journal of Operational Research*, vol. 272, no. 3, pp. 879–904, 2019.

[18] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia, “A review of dynamic vehicle routing problems,” *European Journal of Operational Research*, vol. 225, no. 1, pp. 1–11, 2013.

[19] G. Laporte, “Fifty years of vehicle routing,” *Transportation Science*, vol. 43, no. 4, pp. 408–416, 2009.

[20] M. Gendreau, G. Laporte, and R. Séguin, “Stochastic vehicle routing,” *European Journal of Operational Research*, vol. 88, no. 1, pp. 3–12, 1996.

[21] P. Jaillet and M. R. Wagner, “Generalized online routing: new competitive ratios, resource augmentation, and asymptotic analyses,” *Operations Research*, vol. 56, no. 3, pp. 745–757, 2008.

[22] N. Secomandi and F. Margot, “Reoptimization approaches for the vehicle-routing problem with stochastic demands,” *Operations Research*, vol. 57, no. 1, pp. 214–230, 2009.

[23] A. E. Fallahi, C. Prins, and R. Wolfler Calvo, “A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem,” *Computers & Operations Research*, vol. 35, no. 5, pp. 1725–1741, 2008.

[24] H. Wang, Y. Fu, M. Huang, G. Q. Huang, and J. Wang, “A NSGA-II based memetic algorithm for multiobjective parallel flowshop scheduling problem,” *Computers & Industrial Engineering*, vol. 113, pp. 185–194, 2017.

[25] H. Wang, M. Huang, and J. Wang, “An effective metaheuristic algorithm for flowshop scheduling with deteriorating jobs,” *Journal of Intelligent Manufacturing*, vol. 30, no. 7, pp. 2733–2742, 2019.

[26] H. Wang, I. Moon, S. Yang, and D. Wang, “A memetic particle swarm optimization algorithm for multimodal optimization problems,” *Information Sciences*, vol. 197, pp. 38–52, 2012a.

[27] H. Wang, D. Wang, and S. Yang, “A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems,” *Soft Computing*, vol. 13, no. 8-9, pp. 763–780, 2009.

[28] H. Wang, S. Yang, W. H. Ip, and D. Wang, “A memetic particle swarm optimisation algorithm for dynamic multi-modal optimisation problems,” *International Journal of Systems Science*, vol. 43, no. 7, pp. 1268–1283, 2012b.

[29] C. Prins, “A simple and effective evolutionary algorithm for the vehicle routing problem,” *Computers & Operations Research*, vol. 31, no. 12, pp. 1985–2002, 2004.

[30] Y. S. Ong and A. J. Keane, “Meta-lamarckian learning in memetic algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 2, pp. 99–110, 2004.

[31] S. Yang, “Genetic algorithms with memory- and elitism-based immigrants in dynamic environments,” *Evolutionary Computation*, vol. 16, no. 3, pp. 385–416, 2008.

[32] S. Yang, Y. S. Ong, and Y. Jin, *Evolutionary Computation in Dynamic and Uncertain Environments*, Springer-Verlag, Berlin, Germany, 2007.

[33] H. Wang, S. Yang, W. H. Ip, and D. Wang, “A particle swarm optimization based memetic algorithm for dynamic optimization problems,” *Natural Computing*, vol. 9, no. 3, pp. 703–725, 2010.