SAFE: Scale-Adaptive Fitness Evaluation method for expensive optimization problems

SAFE: Scale-Adaptive Fitness Evaluation Method for Expensive Optimization Problems

Sheng-Hao Wu, *Student Member, IEEE*, Zhi-Hui Zhan, *Senior Member, IEEE*, and Jun Zhang, *Fellow, IEEE*

***Abstract*—The key challenge of expensive optimization problems (EOP) is that evaluating the true fitness value of the solution is computationally expensive. A common method to deal with this issue is to seek for a less expensive surrogate model to replace the original expensive objective function. However, this method also brings in model approximation error. To efficiently solve the EOP, a novel scale-adaptive fitness evaluation (SAFE) method is proposed in this article to directly evaluate the true fitness value of the solution on the original objective function. To reduce the computational cost, the SAFE method uses a set of evaluation methods (EM) with different accuracy scales to cooperatively complete the fitness evaluation process. The basic idea is to adopt the low-accuracy scale EM to fast locate promising regions and utilize the high-accuracy scale EM to refine the solution accuracy. To this aim, two EM switch strategies are proposed in the SAFE method to adaptively control the multiple EMs according to different evolutionary stages and search requirements. Moreover, a neighbor best-based evaluation (NBE) strategy is also put forward to evaluate the solution according to its nearest high-quality evaluated solution, which can further reduce computational cost. Extensive experiments are carried out on the case study of crowdshipping scheduling problem in the smart city to verify the effectiveness and efficiency of the proposed SAFE method, and to investigate the effects of the two EM switch strategies and the NBE strategy. Experimental results show that the proposed SAFE method achieves better solution quality than some baseline and state-of-the-art algorithms, indicating an efficient method for solving EOP with a better balance between solution accuracy and computational cost.**

***Index Terms*—Crowdshipping scheduling, expensive optimization problem (EOP), fitness evaluation (FE) method.**

Manuscript received April 28, 2020; revised September 15, 2020 and December 8, 2020; accepted January 4, 2021. Date of publication January 14, 2021; date of current version May 28, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2102102; in part by the Outstanding Youth Science Foundation under Grant 61822602; in part by the National Natural Science Foundations of China (NSFC) under Grant 61772207 and Grant 61873097; in part by the Key-Area Research and Development of Guangdong Province under Grant 2020B010166002; in part by the Guangdong Natural Science Foundation Research Team under Grant 2018B030312003; in part by the Guangdong–Hong Kong Joint Innovation Platform under Grant 2018B050502006; and in part by the Hong Kong GRF-RGC General Research Fund under Grant 9042816 (CityU 11209819). *(Corresponding authors: Zhi-Hui Zhan; Jun Zhang.)*

Sheng-Hao Wu and Zhi-Hui Zhan are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Pazhou Laboratory, Guangzhou 510330, China (e-mail: zhanapollo@163.com).

Jun Zhang is with the Hanyang University, Ansan 15588, South Korea, and also with Victoria University, Melbourne, VIC 8001, Australia.

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TEVC.2021.3051608.

Digital Object Identifier 10.1109/TEVC.2021.3051608

I. Introduction

Evolutionary computation (EC) algorithm, inspired by evolution phenomenon and swarm intelligence in nature, is a population-based tool for solving optimization problems. Owing to the simplicity and gradient-free feature, multiple variants of EC algorithms have been developed over the decades, such as genetic algorithm (GA) [1]–[3], ant colony system (ACS) [4]–[6], particle swarm optimization (PSO) [7]–[9], and differential evolution (DE) [10]–[12], which have been applied to tackle with a wide range of optimization problems in operational research [13]–[15], artificial intelligence [16]–[18], industrial manufacturing [19]–[21], and many other fields [22]–[26]. Optimization is a process to optimize certain objective(s), with the satisfaction of a set of constraints [27]. EC algorithms begin with an initial population of feasible solutions (individuals) and iteratively generate new solutions (offspring) by reproduction operations (e.g., crossover and mutation). Then, by selection based on fitness evaluation (FE), individuals with better fitness values have a greater opportunity to survive into the next generation and consequently, high-quality solutions are propagated. Therefore, the FE is a significant process to drive the EC algorithm to approach the optimum. Noted that the fitness is dependent on the objective but there is also a difference. The objective function is related to the problem itself, which is the optimization target, while the fitness function is related to the algorithm design, which is designed to help the selection operation. Therefore, a suitable fitness function is needed to be built based on the objective function to connect the problem and the algorithm. If the objective function of the optimization problem is very clear and the fitness function is well designed, the EC algorithm will work well. However, many real-world problems are typically characterized by NP-hard, stochastic nature, and high computational cost [28], [29], which means the FE could be nondeterministic and computationally expensive. This is the expensive optimization problem (EOP) that challenges the traditional EC algorithms. Therefore, how to build a suitable fitness function for real-world EOP is very difficult.

To efficiently solve the EOPs by EC algorithms, a widely used method in the literature is to build a cheap fitness function by using surrogates to replace the original expensive objective function. The surrogate method can be classified into two categories: 1) data-driven method and 2) knowledge-driven method. The data-driven method [29]–[32] aims to build a surrogate model based on the collected historical

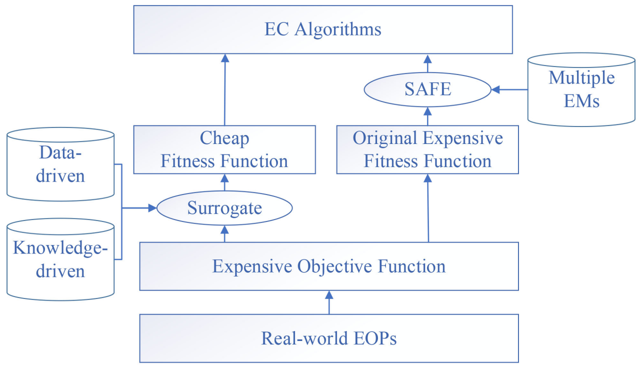


Fig. 1. Two ways to help EC algorithms efficiently solve EOPs.

data as a cheaper fitness function to replace the original highly expensive objective function. Different function fitting techniques, such as Kriging model [33] and artificial neural networks [34] have been studied to make full use of the limited data. Moreover, Li *et al.* [35] proposed a boosting strategy with a localized data generation method to produce synthetic data when the available data were not sufficient to build a satisfactory model. The knowledge-driven method [36]–[38], which is proposed by domain-specific experts, tries to simplify the original objective function to be a cheaper fitness function by numerical simulations based on knowledge-embedded mathematical models. The basic idea of the above two methods is to seek for a less expensive approximation model to replace the original expensive objective function. However, no matter the data-driven methods or the knowledge-driven methods, they more or less make the replaced/simplified cheaper fitness function somehow different from the original objective function of the real-world problem. This will unavoidably bring in model error which is harmful to practical applications. To reduce the model error of a single low-fidelity surrogate, some methods with multiple surrogates [35], [39] and multifidelity surrogates [40]–[42] are also proposed to balance the computational cost and the fidelity, which can reduce model error to some extent. However, the fitness function approximated by surrogates needs lots of data and training efforts. Moreover, the built surrogates may only be meaningful for the optimization process, but might not have actual meaning in the real-world applications as the original objective function does. That is, the solution that performs well in the surrogate fitness function does not necessarily work well in the real-world problem because of the surrogated model error.

To fill the gap between the expensive objective function of real-world problems and the FE for the EC algorithm, this article proposes a scale-adaptive FE (SAFE) method to enable the EC algorithm to efficiently solve the EOP directly on the original objective function. The major difference in solving EOPs between the proposed SAFE method and surrogate-based method is shown in Fig. 1. In the method on the left side of Fig. 1, the main contribution lies between the real-world objective function and the cheap fitness function, where a lot of surrogate methods are proposed. However, our work focuses on the right side of Fig. 1 whose main contribution lies between the fitness function and the algorithms, where the SAFE method is proposed to drive EC algorithms to efficiently deal with the original expensive fitness function. The SAFE method is a general framework that a set of different evaluation methods (EM) with different accuracy scales work together and cooperate adaptively to complete the FE process. Herein, the design of the SAFE method is inspired by the phenomenon in human cognition that people usually use different scales of resolutions to obtain different levels of cognitive information on the object. Therefore, human beings must combine multiple resolutions to make a comprehensive observation. Similarly, the SAFE method combines different EMs with different accuracy scales to make a comprehensive evaluation on the fitness function. When using SAFE, the fitness function can be the same as or have no significant difference with the original objective function, but the fitness value of the solution can be evaluated in different accuracy scales by different EMs in different evolutionary stages to reduce the computational cost. It is worth noticing that the multifidelity surrogate-based method has similarities to the SAFE method in balancing the solution accuracy and the computational cost. However, the major difference is that most multifidelity surrogate-based methods mainly consider only one EM and controls the evaluation fidelity by controlling the simulation parameters (e.g., maximum iterations) of this EM while the SAFE method adopts heterogeneous EMs with different accuracy scales to complete the FE. Hence, the SAFE method is a more general framework that not only the EM itself can be controlled, but also different EMs can cooperate.

The proposed SAFE method has three main advantages. First, the SAFE method is flexible because it adopts different EMs with different complexities that can evaluate the fitness function in different accuracy scales. Therefore, the SAFE is flexible to different FE accuracy requirements. Second, among different EMs, high-accuracy scale EM (HSEM) can obtain fitness value with higher accuracy but is computationally expensive, while low-accuracy scale EM (LSEM) has a less computational cost but results in fitness value with lower accuracy. Therefore, our proposed SAFE method is efficient because it combines different EMs and adaptively controls these EMs to calculate the fitness value based on different evolutionary stages and search requirements. Third, unlike the surrogate-based method, the SAFE method evaluates the solution directly on the fitness (i.e., objective) function which does not bring much distortion to the original problem. Consequently, the SAFE method is easy to apply to a class of EOPs for which some effective EMs have been proposed under the premise of remaining the objective function unchanged.

The EOPs widely exist in many fields like manufacturing and logistics systems in the smart city. For the study, a significant crowdshipping scheduling problem is put forward. In the crowdshipping system, the delivery tasks are completed by the professional fleet (PF) and idle occasional couriers (OC). The goal is to minimize the total delivery cost incurred by the PF and the compensation paid to the OC. Herein, the total cost is acting as the objective function. A crowdshipping delivery plan includes not only the "task assignment" determining which customer should be served by the PF or the OC, but also the "routing plan" for the PF. As the SAFE method uses a fitness function the same as the objective function, routes for

PF are required to plan to obtain the total cost. Obtaining the routing plan for PF is an optimization problem which is well known as the capacitated vehicle routing problem (CVRP). Since CVRP is NP-hard, it is time consuming and computationally expensive to get the optimal solution by existing exact methods. Many nonexact methods like greedy algorithm, variable neighborhood search (VNS), and ACS have been proposed in the literature and these CVRP solvers can be treated as different EMs in this article. Normally, they can be partitioned into different accuracy scales based on their search ability and computational cost. To be specific, the solvers with a higher accuracy scale tend to obtain better solutions but are more time consuming. Besides, parameters (e.g., maximum iterations) of the solver can be controlled to form more solvers with different accuracy scales. Obviously, there is a tradeoff between the solution quality and computational cost. Therefore, this article proposes to use the SAFE method to efficiently solve the crowdshipping scheduling problem, which would be useful and meaningful to the development of intelligent logistics systems in the smart city. Moreover, this problem is mathematically modeled as the CVRP with OC (CVRP-OC). Since the subproblem CVRP is well known and has been extensively studied with many available problem solvers (i.e., EMs), the crowdshipping scheduling problem (i.e., CVRP-OC) is very suitable to test and verify the SAFE method. The advantages and contribution of our work are summarized as follows.

1) Different from the previous research that usually changes the objective function of EOP, our proposed SAFE method manages a set of heterogeneous EMs on various accuracy scales to directly evaluate the original objective function as the fitness function. This is helpful to reduce the model error and is flexible in balancing the accuracy and computational cost.
2) To take advantage of various EMs, switching among different EMs is well designed as an inter-EM (i.e., among different EMs) management, including one-way and two-way EM switch strategies. This is helpful to switch to suitable EM adaptively according to the search requirements of different evolutionary stages.
3) To further help the SAFE method, a neighbor best-based evaluation (NBE) strategy is put forward to make full use of high-quality solutions to further reduce computational cost. As the NBE strategy is performed on the solutions within a particular EM, we refer to it as an intra-EM (i.e., within an EM) improvement.

The remainder of this article is organized as follows. Section II introduces the real-world EOP case study about crowdshipping scheduling. Section III develops the algorithm with the SAFE method to address the problem in detail. Section IV presents experimental studies. Section V concludes this article.

II. Crowdshipping Scheduling Problem

A. Background and Problem Formulation

Crowdshipping [43]–[47], also known as crowdsourcing delivery, is a new trend in the current logistics systems. In the crowdshipping mode, logistics companies not only employ a PF but also appeal to ordinary idle people referring to the OC, to work cooperatively to complete delivery tasks. In return, a small fee of compensation is offered to the OC. The purpose of the crowdshipping problem is to minimize the total delivery cost incurred by the PF and the compensation paid to the OC.

To be specific, the workflow is started with the arrival of orders (customers) to be served associated with information, such as customers' locations and demand (e.g., weights of the goods). The scheduler is required to decide which orders should be accomplished by the PF or the OC. Once a decision is made, which refers to a "task assignment," the customers for OC will be released and the cost can be calculated. The rest of the customers will be served by the PF which departs from the depot according to a "routing plan" in a given order, of which the total cost should be as small as possible. When all customers are satisfied, the PF will return to the depot. By the way, the load of vehicles in the PF should not exceed their capacity.

The problem concerning the crowdshipping system is mathematically modeled as CVRP-OC based on the well-known CVRP [48]. The CVRP-OC is defined on a complete undirected graph $G = (CUS, EDG)$. N is the number of customers and $CUS = \{c_0, c_1, \ldots, c_N\}$ is the set of *cus*tomer nodes which contains $N + 1$ elements. c_0 is the depot where all vehicles of the PF depart and finally return to. Each customer is associated with a demand which makes up $Demand = \{d_i \mid 0 \le i \le N\}$. Moreover, the demand d_0 for the depot is zero. All vehicles of PF have the same maximum load limitation, termed as *Capacity*. Here, the capacity of the OC is assumed to be infinite since many idle couriers constitute the OC as a whole. Combinations of pairwise customers constitute an *edg*e set $EDG = \{(i, j) \mid 0 \le i \ne j \le N\}$. Each edge is associated with a traveling cost termed as $Cost = \{w_{ij} \mid 0 \le i \ne j \le N\}$. Note that $i = 0$ or $j = 0$ means the edge that connects the depot and a customer. There are totally $K + 1$ vehicles available. The vehicle $k = 0$ means the OC and the vehicle $k > 0$ means the PF. The CVRP-OC can be formally put forward as follows.

Decision variables are defined as

$$x_{ijk} = \begin{cases} 1, & \text{if edge}(i, j) \text{ is traversed by vehicle } k \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ik} = \begin{cases} 1, & \text{if customer } i \text{ is served by vehicle } k \\ 0, & \text{otherwise.} \end{cases}$$

The objective is to minimize the total cost

$$\min \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=1}^{K} w_{ij} x_{ijk} + \sum_{i=0}^{N} \rho \cdot w_{0i} y_{i0} \tag{1}$$

$$\text{s.t.} \sum_{i=0, i \ne j}^{N} \sum_{k=1}^{K} x_{ijk} = y_{jk} \quad \forall j = 1, \ldots, N \tag{2}$$

$$\sum_{j=0, j \ne i}^{N} \sum_{k=1}^{K} x_{ijk} = y_{ik} \quad \forall i = 1, \ldots, N \tag{3}$$

$$\sum_{k=0}^{K} y_{ik} = 1 \quad \forall i = 1, \ldots, N \tag{4}$$

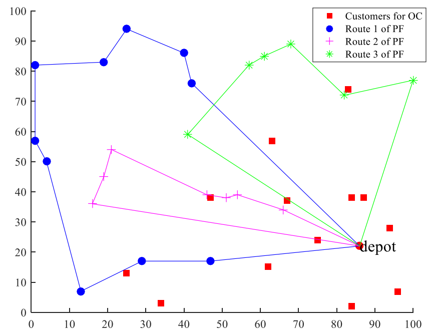


Fig. 2. Example of the crowdshipping delivery plan.

$$\sum_{i=0}^{N} d_i y_{ik} \leq \text{Capacity} \quad \forall k = 1, \ldots, K \tag{5}$$

$$\sum_{k=0}^{K} y_{0k} = K + 1 \tag{6}$$

$$\sum_{i=0}^{N} \sum_{j=0}^{N} x_{ij0} = 0 \tag{7}$$

$$u_{ik} - u_{jk} + N \cdot x_{ijk} \leq N - 1 \quad \forall i, j = 1, \ldots, n, i \neq j \quad \forall k = 1, \ldots, K. \tag{8}$$

In (1), the left term in the objective function is the cost incurred by the PF according to a routing plan and the right term is the compensation cost incurred by the OC. The ρ is the compensation coefficient that can be multiplied with the cost w to obtain the compensation cost. Constraints (2) and (3) guarantee the flow conservation rule for each node in the graph. Constraint (4) guarantees that each customer should be served by exactly one vehicle. Constraint (5) prevents the load of each vehicle from exceeding capacity. Constraint (6) requires that each vehicle should depart from the depot to begin a trip. Constraint (7) explicitly demonstrates that there is no need to plan routes for the OC since every route is a single-customer trip. Constraint (8) eliminates isolated subtours to ensure validity.

B. Solution Representation

The decision process for the CVRP-OC can be divided into two stages: 1) assigning customers to OC/PF and 2) planning routes for the vehicles in PF. For the first stage, it is a scheduling problem. Assignments for this problem can be encoded into a 0-1 string of length N and each position in the string represents a customer with 0 meaning the corresponding customer served by the OC and 1 by the PF.

For the second stage, planning routes for the PF is the CVRP. Many researchers have studied the problem and many problem solvers have been proposed. CVRP solvers can be briefly categorized as exact methods and heuristic methods. Exact methods [49], [50], such as branch-and-bound [51] and column generation [52] solve the problem based on a mixed-integer programming model. However, the time for solving the problem increases significantly as the problem size becomes larger. Considering the real-time requirement in practical applications, adopting exact methods can hardly work. Heuristic methods, such as greedy algorithm, VNS [53]–[55], and ACS [4], [14] are methods for obtaining a suboptimal solution in a reasonable time while the global optimum is not guaranteed.

Overall, a solution S (i.e., crowdshipping delivery plan) for the CVRP-OC includes the assignment A, the routing plan R, and the total cost C as the fitness value, which also acts as the objective function in the proposed model. The evaluation process of the solution S takes an assignment A as input and outputs the routing plan R for PF and corresponding total cost C. Hence, evaluating a solution is equal to evaluating the assignment of the solution. An example of a crowdshipping delivery plan is shown in Fig. 2.

Obviously, evaluating fitness value for a solution S requires the returning result of the routing optimization process which accounts for the major computational cost and makes the solution expensive to evaluate.

C. CVRP Solvers

For the CVRP, we briefly introduce three types of solvers which are mainly proposed by domain-specific experts and researchers.

Greed+2opt: The algorithm constructs the first route for a vehicle beginning with a random customer and iteratively adds the next customer into the route according to a greedy criterion (i.e., minimum distance from the currently served customer to the unvisited customers). Until the capacity restriction is violated, the vehicle will return to the depot and then a new route for the next vehicle available is constructed in the same way repeatedly. After all the customers have been served, *2opt* operation [56] is performed on every route of the plan to eliminate the intersections to further improve the quality of routes. Notably, once the first customer is randomly selected, the rest of the route is deterministic by the greedy rule. Hence, multiple plans with a different first-selected customer can be constructed to obtain a better routing plan. To better exploit the different accuracy scales of the greedy algorithm solver, we name *Greed+2opt–1* as the solver in which one plan is constructed and *Greed+2opt–N* as the solver in which N plans with a different first-selected customer are constructed, and then the best plan is output as the result.

VNS+2opt: VNS [53]–[55] is essentially a local search algorithm, in which a feasible solution is constructed at first. Then multiple neighborhood solutions are generated based on the current solution by some operations which are artificially designed. If some of the produced solutions outperform the original one, the best one is used as the current solution in the next generation. Moreover, jumping out techniques are incorporated to avoid being trapped by the local optima. Similar to the previous solver, each solution is refined by a *2opt* operation after all the routes are constructed.

ACS+2opt: ACS [4], [14] is a population-based iterative approach in which several routing plans are constructed independently according to probability-based rules. The routing plan with lower cost is rewarded by strengthening the probability to select the edges that are involved in the routing plan. In every routing plan construction iteration, the probability matrix

Algorithm 1 BasicEvaluation(A, EM)

input: A, EM // A = assignment to be evaluated, EM = evaluation method
output: R, C // R = routing plan for A, C = total cost for A
param: N, $PARAM(solvers)$ // N = number of total customers
//$PARAM(solvers)$ = parameter setting for solvers

BEGIN
  IF EM = $Greed + 2opt - 1$
    Construct R by the greedy algorithm and obtain C
  ELSEIF EM = $Greed + 2opt - N$
    Construct R by the greedy algorithm N times, output R with the smallest C
  ELSEIF EM = $VNS + 2opt$
    Construct R by VNS and obtain C
  ELSEIF EM = $ACS + 2opt$
    Construct R by ACS and obtain C
  END
END

helps the algorithm to pick promising edges and high-quality routing plans are propagated.

The reason for stating several CVRP solvers is that the optimization process of the routes for PF has a significant impact on the evaluation of the assignment. Each CVRP solver can be viewed as an EM for the assignment. The solvers vary from many aspects, such as computational cost and solution quality. For demonstration, experiments on CVRP instances are carried out and the results are presented in Table S.I in the supplementary material. The results show that with more plans being constructed, $Greed + 2opt - N$ surpasses $Greed + 2opt - 1$ in terms of solution quality and stability, but consequently, the more computational time is required. $ACS + 2opt$ and $VNS + 2opt$ achieve better performance than the previous two greedy algorithms in terms of solution quality, but the execution time of the solvers grows significantly with the problem size. Notably, it is hard to draw a conclusion that either $ACS+2opt$ or $VNS+2opt$ can obtain better solution quality. It seems that their ability to solve an instance is dependent on the problem structure (e.g., number of customers and customers' distribution). We define the EM for a solution (i.e., assignment) that is high-accuracy but more time consuming, such as $ACS + 2opt$ and $VNS + 2opt$ as the HSEM while the lower accuracy and cheaper ones, such as $Greed+2opt-1$ and $Greed+2opt-N$ are denoted as the LSEM. Algorithm 1 shows the procedure to evaluate the assignments by various EMs. To reduce ambiguity with the NBE introduced in Section III-B, the evaluation by the EM mentioned in the following context refers to the basic evaluation shown in Algorithm 1.

III. SAFE Method

To address the crowdshipping scheduling problem, which is computationally expensive, our proposed SAFE method can be summarized as inter-EM management and intra-EM improvement. For the inter-EM management, we adopt two EM switch strategies to automatically control the various EMs to switch in different evolutionary stages. The basic idea of the SAFE method is to use LSEM to find a promising region and further refine the search with HSEM. For the intra-EM improvement, the NBE strategy is carried out to make use of the correlation between the well-performed solution and its neighbor solution. The NBE strategy can make significant benefits for the HSEM in the later evolutionary stage when most of the individuals in the population are similar. Hence, expensive re-evaluations of the same solutions and similar solutions are avoided in which way the computation cost is reduced. Moreover, the archives for multiple EMs are carried out with operations, including archive insertion, archive reduction, and archive migration to efficiently manage high-quality evaluated solutions.

A. Inter-EM Management by SAFE

1) Motivation for SAFE: The design of the SAFE method is inspired by the phenomenon in human visual cognition. When people observe objects, they usually use different scales of resolutions for different levels of cognition. That is, different distances from the observing point to the object lead to different resolutions (i.e., scales). More specifically, a remote observing distance causes a coarse resolution that the general structure of the object can be observed while the details of the object could be difficult to be captured. On the contrary, a close observing distance causes a fine resolution that the detail of the object can be captured while lacking a general understanding of the object. To provide a comprehensive understanding of the object, it may be better to combine observative information on different scales. A simple idea is to analyze the object at a coarse resolution and then gradually increase resolution. This way of handling multiple resolutions has been utilized in the image processing area and brings in great success [57], [58].

Therefore, we adopt a similar idea that uses multiple EMs of different accuracy scales to effectively solve the EOP. In the SAFE method, the EC algorithm can switch between EMs with different accuracy scales to evaluate individuals to achieve a tradeoff between solution accuracy and computational cost. Without significant modification to the original EC algorithm, the SAFE method is embedded into the beginning of every generation to determine a suitable EM for this generation. The SAFE method is adaptively controlled by indicators CF_1 and CF_2 that can reflect the potential of the next EM. Once a switch condition is met, the current EM for the population is regarded as not suitable for the evolution anymore and the EM should be switched.

Switching between various EMs is the decision-making process. The main issue is to decide when to switch EM according to a criterion and which EM should be selected as the next EM. To start with, the EM pool contains four kinds of EMs which are $Greed + 2opt - 1$, $Greed + 2opt - N$, $VNS + 2opt$, and $ACS + 2opt$. For the sake of simplicity, they are denoted as EM_1, EM_2, EM_3, and EM_4, respectively. They are arranged in an order that the EM with higher rank has higher accuracy for evaluating an assignment (i.e., solution). It is not easy to define whose accuracy is higher between EM_3 and EM_4 because their performance is dependent on the problem structure. Hence, they are arranged in ascending order according to the computational cost. Besides, reachable EMs of each EM can be defined by a directed graph. For example, if the current EM is EM_2 and its reachable EMs are EM_3 and EM_4, then the manager should decide in what condition the algorithm should switch to EM_3 or EM_4.

Two EM switch strategies are proposed to achieve adaptive control. In the one-way EM switch strategy, the algorithm starts from an LSEM and switch to HSEM according to the

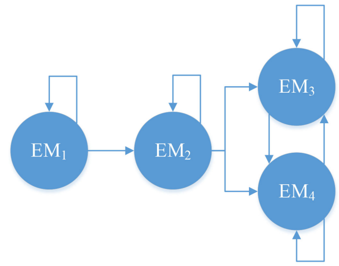


Fig. 3. Topology structure of one-way EM switch strategy.

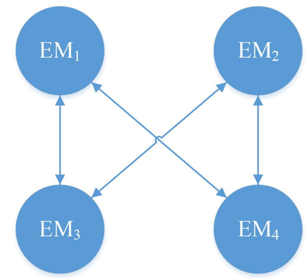


Fig. 4. Topology structure of two-way EM switch strategy.

contribution factor CF_1. The key feature is that once the LSEM is switched to HSEM, there is no chance to turn back to LSEM. In the two-way EM switch strategy, the algorithm can switch between LSEM and HSEM and adopts two contribution factors CF_1 and CF_2 to decide which EM to switch to. Both the one-way and two-way EM switch strategies are carried out when the evolutionary process has stagnated for a number of consecutive generations, which is indicated by a parameter named η to represent the maximal stagnated generations for triggering the switch.

2) One-Way EM *Switch Strategy:* In the one-way EM switch strategy, the reachable EMs of every EM are shown in Fig. 3. The EC algorithm initializes with the EM_1 and undergoes the evolutionary process. The switch condition is defined as the stagnation of the population exceeding the threshold η without improvement on the global best of the current EM. After the switch, the stagnation of the population is reset to zero indicating the population entering a new EM. The contribution factor CF_1 is put forward to decide which EM the algorithm should switch to. The calculation of CF_1 is obtained by a "pre-evaluation" process on the current population described as follows.

First, all the individuals in the current population are evaluated by the NBE strategy using the current EM and all the next reachable EMs. Note that the evaluation based on the NBE strategy is to reduce the computational cost, which will be described in Section III-B. The global best individuals of the current EM (i.e., $\mathrm{EM}_{\mathrm{cur}}$) and each next EM (i.e., $\mathrm{EM}_{\mathrm{next}}$) are denoted as $F_{\mathrm{best}}(\mathrm{EM}_{\mathrm{cur}})$ and $F_{\mathrm{best}}(\mathrm{EM}_{\mathrm{next}})$, respectively. Then, the CF_1 value of this $\mathrm{EM}_{\mathrm{next}}$ is calculated as

$$\mathrm{CF}_1(\mathrm{EM}_{\mathrm{next}}) = F_{\mathrm{best}}(\mathrm{EM}_{\mathrm{next}}) - F_{\mathrm{best}}(\mathrm{EM}_{\mathrm{cur}}). \tag{9}$$

As our objective is to find the solution with the minimal total cost, a negative CF_1 value indicates that the next EM may be more suitable than the current EM. Therefore, the next EM that has the smallest negative CF_1 value is selected as the final next EM. However, it should be noted that due to the stochastic nature of different heuristic algorithms, it is not guaranteed that a higher accuracy scale EM can improve the current population. Therefore, if none of the $F_{\mathrm{best}}(\mathrm{EM}_{\mathrm{next}})$ is better than $F_{\mathrm{best}}(\mathrm{EM}_{\mathrm{cur}})$, i.e., none of the CF_1 value is smaller than zero, the algorithm continues with the current EM until the next switch condition is met.

The key feature of the one-way EM switch strategy is that once the LSEM is switched to the HSEM, there is no chance to turn back. The task for LSEM is to find promising regions and the HSEM is adopted to further exploit.

3) Two-Way EM Switch Strategy: The LSEM (e.g., EM_1) typically has features of lower accuracy, a higher degree of uncertainty, and more importantly, less computationally expensive. Hence, it is possible to take advantage of the uncertainty and cheapness of LSEM and adopt it to search for another promising region. Based on such motivations, the two-way EM switch strategy is proposed. The reachable EMs for every EM are defined as the topology shown in Fig. 4.

The EC algorithm starts with an LSEM. When the switch condition is met, the next EM will be selected to switch to. As its name indicates, the two-way EM switch strategy has two EM switch mechanisms. The first is the switch from LSEM to HSEM. That is, if the current EM is LSEM (i.e., EM_1 or EM_2), it is better to choose an EM with a higher accuracy that shows potential in improving the population quality. Hence, the contribution factor CF_1 is adopted to try to switch to an HSEM. The process is the same as the one-way EM switch strategy does.

The second is the switch from HSEM to LSEM. That is, if the current EM is HSEM (i.e., EM_3 or EM_4) and the evolution meets the switch condition, then it is the intention to locate another promising region by switching from HSEM to LSEM. We denote the population size as ps, the fitness value of the ith individual in the current population evaluated by the next EM as $F_i(\mathrm{EM}_{\mathrm{next}})$, then the contribution factor CF_2 is calculated as

$$\mathrm{CF}_2 = \sum_i \sum_{i \neq j} \frac{I\big(\big(F_i(\mathrm{EM}_{\mathrm{next}}) - F_j(\mathrm{EM}_{\mathrm{next}})\big) \cdot \big(F_i(\mathrm{EM}_{\mathrm{cur}}) - F_j(\mathrm{EM}_{\mathrm{cur}})\big) \geq 0\big)}{ps \cdot (ps - 1)/2} \tag{10}$$

where $I(\cdot)$ is an identity function and equals to 1 if the condition in the parentheses is satisfied and 0 otherwise. The indicator CF_2 is designed to measure the alignment degree between the current EM and the next EMs. First, every pair of individuals in the population is compared. If individual i is better than individual j under current EM while the next EM agrees on the comparison of this pair, the $I(\cdot)$ is 1 and is added up to the result of CF_2. The more agreements on the comparison emerge, the larger CF_2 it should be and results in a larger degree of alignment between the current EM and the next EM. Since the purpose is to increase the possibility to find another different promising region, the selection criterion is choosing the next EM with the smallest CF_2 to switch to. In such a way, the next EM tends to disagree with the

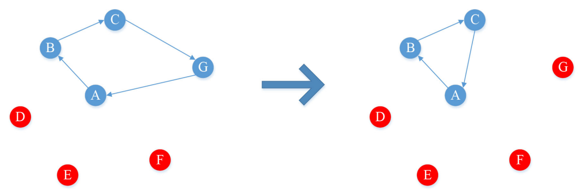


Fig. 5. OpExclude operation.

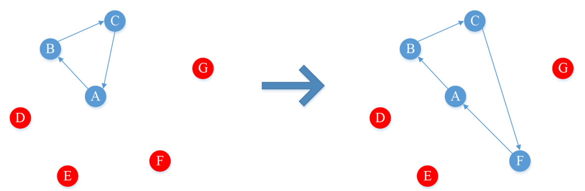


Fig. 6. OpJoin operation.

current judgement on the comparison between individuals and improves the jumping out ability.

B. Intra-EM Improvement by NBE

In this section, we introduce the NBE strategy in detail. First, in order to save computational cost, the NBE strategy uses high-quality evaluated solutions to evaluate the fitness of the new solution and reduces re-evaluations of the duplicate and similar solutions. To achieve this goal, two operations (i.e., OpJoin and OpExclude) are adopted. Furthermore, to manage the evaluated solutions, the archive is put forward to store them with operations, including archive insertion, archive reduction, and archive migration.

1) NBE Strategy: Adopting the HSEM to evaluate solutions might be computationally cheaper than the exact methods but still leads to a higher computational cost than LSEM. The issue becomes prominent when the problem size grows larger. Considering a CVRP instance with 135 customers, the time for solver $ACS + 2opt$ which is EM_4 to solve the case is about 1 s. At the population level, it takes about 1 min to evaluate the population of size 50. It is supposed that the optimization running time is restricted to 10 min. As a result, at most 10 generations can be executed for the optimization which might be far from enough to take advantage of the high accuracy of HSEM. To address the issue, it is possible to consider the correlation between similar solutions and make full use of high-quality solutions. The reason is that high-quality routing plans are more likely to share some common good structures. Therefore, when evaluating a new solution, its similar and high-quality evaluated solution, which is named neighbor best solution (NBS) in this article, can be used. That is, only a slight change is needed to reconstruct the *NBS* to obtain the new solution and its corresponding fitness value. Therefore, the NBE strategy leads to an intra-EM (i.e., within one EM) improvement.

To proceed on, two operations named OpExclude and OpJoin are defined and shown in Figs. 5 and 6 with blue points representing customers served by the PF and red points served by the OC. It is supposed that the two assignments are very similar with a slight difference in the assignment of the customer G in Fig. 5. The total cost of the left assignment is already known which is evaluated by the current EM. The OpExclude is to eliminate the customer G in routes and makes it served by an OC. Then, the vacant customers C and A in the PF routing plan are linked. The change of cost by edge deletion and insertion and compensation is added to the total cost of the left assignment to obtain the total cost of the right assignment. Likewise, when turning the customer F to be served by PF in Fig. 6, OpJoin selects the edge CA that causes the least increase on the total cost and adds the customer F into the routes for the PF. Afterward, the total cost of the new assignment can also be derived.

The procedure of the NBE strategy is shown in Algorithm 2. Given the *NBS* which contains the assignment *NBS.A* with known total cost *NBS.C* and the routing plan *NBS.R*, the current EM, and the assignment *A* to be evaluated, the similarity between two assignments is calculated with hamming distance metric at first. If their distance exceeds a threshold named neighborhood distance *nd*, the assignment will be evaluated by the current EM, as line 16. Otherwise, the two assignments *A* and *NBS.A* are similar and therefore the *A* can be evaluated based on *NBS* to reduce computational time, as lines 3–13. The basic idea is to modify the route of *NBS* to make it become the same as the route of *A* and then its resultant cost can be regarded as the fitness value of *A*. First, the 0-1 strings of *NBS.A* and *A* are compared to find customers that should be removed from the PF or inserted into the PF in the *NBS.A*. That is, those customers served by PF in *NBS.A* (i.e., $NBS.A == 1$) but not in *A* (i.e., $A == 0$) are denoted as *rmvCustomer* (line 5) and are removed from the PF route of the *NBS* by the OpExclude operation (line 6). Then, those customers served by PF in *A* but not in *NBS.A* are denoted as *insCustomer* (line 7) and are inserted into the PF route of the *NBS* by the OpJoin operation (line 8). Note that, the customers in the *rmvCustomer* and *insCustomer* lists are shuffled by random permutation before they are used by the OpExclude and OpJoin. By performing the OpExclude and OpJoin operations on the customers one by one, the final new assignment is obtained together with its fitness value. After all the customers in the *rmvCustomer* and *insCustomer* lists are settled, *2opt* operation is performed on every route in the routing plan for PF to refine the solution (line 13). However, if an insertion operation is unsuccessful (i.e., no routes in the plan can afford the inserting customer's delivery service anymore due to the restriction of capacity), the NBE strategy fails and the assignment will be evaluated by the EM to get a legal routing plan and corresponding total cost (line 10).

2) Archive Management: The *NBS*s of different EMs are maintained by their corresponding archive which is denoted as Ar_{EM}. The archive is managed by the archive insertion, archive reduction, and archive migration operations as introduced as follows.

Archive Insertion: Algorithm 3 shows the procedure of inserting a solution *S* into the archive which stores *NBS*s. First, all solutions' assignments in Ar_{EM} similar to the given solution's assignment *S.A* are found according to the threshold *nd* which defines the similarity between assignments. If no similar assignment in Ar_{EM} exists, the given assignment is inserted into the archive. Otherwise, the given solution's total cost *S.C*

Algorithm 2 NBE(A, NBS, EM)

```
input: A, NBS, EM
    // A=assignment to be evaluated, NBS.A = neighbor best assignment
    // NBS.R = PF route for NBS.A, NBS.C = total cost for NBS.A
    // EM = evaluation method
output: C, R
param: nd        // nd = neighborhood distance
BEGIN
  IF HammingDistance(A, NBS.A) < nd
    // A == 0 indicates the set of customer served by OC
    // INTERSECT is the set operation
    rmvCustomer = FindCustomers(A == 0, NBS.A == 1);
    [C, R] = OpExclude(rmvCustomer, NBS.C, NBS.R);
    insCustomer = FindCustomers(A == 1, NBS.A == 0);
    [C, R] = OpJoin(insCustomer, C, R);
    IF any customer of insCustomer is inserted unsuccessfully
       [C, R] = BasicEvaluation(A, EM); // Algorithm 1
    ELSE
       // all selected customers are removed and inserted successfully
       [C, R] = 2opt(R);
    END
  ELSE
    [C, R] = BasicEvaluation(A, EM); // Algorithm 1
  END
END
```

Algorithm 3 ArchiveInsertion(S, Ar, EM)

```
input: S, Ar, EM
output: Ar
param: nd // nd = neighborhood distance
  BEGIN
    SNBS = FindAllSolutions(HammingDistance (S.A, Ar_EM.A) < nd);
    // SNBS = the set of NBS relative to S
    IF SNBS is empty
      // no similar assignment is found
      Insert S into Ar_EM;
    ELSE
      IF S.C < min(SNBS.C)
        Insert S into Ar_EM;
        Delete SNBS from Ar_EM;
      END
    END
  END
```

Algorithm 4 ArchiveReduction(Ar, EM)

```
input: Ar, EM
output: Ar
param: MaxArSize // MaxArSize = maximal archive size
  BEGIN
    IF Size(Ar_EM) > MaxArSize
      eliminate (Size(Ar_EM) - MaxArSize) solutions with largest total
       cost from Ar_EM
    END
  END
```

should be compared to all its neighbor solutions' costs in the archive. In the case that the incoming solution outperforms all its neighbor solutions, the incoming solution replaces all its neighbor solution in Ar_{EM}. The purpose is to avoid duplicate or similar assignments in the archive.

Archive Reduction: Archive reduction operation is used to prevent the archive from exceeding the storing limitation *MaxArSize* which indicates the maximal number of *NBS*s that can be stored. At the end of every generation of the EC algorithm, if the current EM's archive size Size(Ar_{EM}) exceeds *MaxArSize*, it eliminates the worst (Size(Ar_{EM})–*MaxArSize*) solutions with the largest total cost from the archive. The pseudocode is shown as Algorithm 4.

Algorithm 5 ArchiveMigration(Ar, EM_{cur}, EM_{next})

```
input: Ar, EM_cur, EM_next
output: Ar
param: None
 1:  BEGIN
 2:    MS_1 = FindBestSolution(Ar, EM_cur);
 3:    // find the solution with the smallest total cost in the archive
 4:    MS_2 = RouletteSelection(Ar, EM_cur);
 5:    // randomly select a solution based on the fitness in a roulette scheme
 6:    [MS_1.C, MS_1.R] = BasicEvaluation(MS_1.A, EM_next); // Algorithm 1
 7:    [MS_2.C, MS_2.R] = BasicEvaluation(MS_2.A, EM_next); // Algorithm 1
 8:    Ar = ArchiveInsertion(MS_1 and MS_2, Ar, EM_next);
 9:    Ar = ArchiveReduction(Ar, EM_next);
10:  END
```

Archive Migration: When the switch occurs, the population will be pre-evaluated by the NBE strategy with a different set of *NBS*s of the next EMs. It should be noted that since the *NBS*s in the archives of EM_{next} are not updated during the evolutionary process with EM_{cur}, the *NBS*s of EM_{next} can be outdated. Therefore, this operator is adopted to migrate the high-quality solutions in the Ar of EM_{cur} to the Ar of EM_{next} to make the NBE of the next EM more efficient when the EM switch is triggered. The migrated solutions are named MS. Two MSs in Ar_{EM} (i.e., MS_1 and MS_2) are selected to be evaluated again by the next candidate EM. The MS_1 is the solution with the best fitness value in the archive of the current EM (line 2), and the MS_2 is selected with the roulette scheme probabilistically (line 4) based on the fitness of solutions in the archive of the current EM. Then, the selected MSs are evaluated by EM (lines 6 and 7) and then inserted into the archive of the next EM by the archive insertion operation followed by the archive reduction operation. Afterward, the population can be pre-evaluated by the NBE using the next EMs. The pseudocode is shown as Algorithm 5.

3) Whole Algorithm: In this section, we briefly discuss the whole algorithm. An easy-used and commonly used EC algorithm, GA, is adopted as the optimizer and is incorporated with the SAFE method and the NBE strategy to solve the CVRP-OC. The resultant algorithm is termed as GA-SAFE. The assignment of a solution is encoded as the 0-1 bit string with 0 meaning the customer served by the OC and 1 by the PF. The output of the evaluation for an assignment includes the fitness value and the routing plan of PF.

In terms of initialization, the initial population consisting of candidate assignments (i.e., solutions) is randomly generated with the probability of 0.5. That is, for the bit string (i.e., assignment), each bit (i.e., customer) is set to 1 with a probability of 0.5 or 0 otherwise. It is noted that heuristic initialization strategies can be incorporated based on *a priori* knowledge of the problem instance. For example, if the compensation coefficient for the OC is rather low, assigning more tasks to the OC than the PF might be more profitable. Therefore, setting a lower probability might cause the population easier to locate a promising area. However, the initialization strategies in our paper are considered to be irrelevant to the compensation scheme. Our work aims to optimize the total cost of the CVRP-OC under whatever compensation scheme. Thus, a trivial initialization scheme is adopted. As

Algorithm 6 GA-SAFE

input: *Customers*, *Cost*, *Capacity*, *Demand*, *depot*
output: *bestAssignment*, *bestCost*, *bestRoutes*
param: *ps*, EM, *nd*, η

```
BEGIN
Initialize the population with size ps and evaluate with EM_cur;
  WHILE stopping criterion is not met
    IF switch condition is met
      // SAFE method and archive migration
      Determine the candidate EM_next for the EM_cur according to the
       topology structure;
      Ar = ArchiveMigration(Ar, EM_cur, EM_next);
      Pre-evaluate current population by NBE with EM_next;
      Calculate the contribution factor for all candidate EM_next;
      Select next EM to switch to according the selection criterion
       (i.e., CF_1, CF_2);
    END
    Selection operation;
    Crossover operation;
    Mutation operation;
    FOR all individuals in the population
      // NBE strategy
      Find the nearest individual in the Ar of EM_cur as the neighbor
       best solution NBS;
      [individual.C, individual.R] = NBE(individual.A, NBS, EM_cur);
      Ar = ArchiveInsertion(individual, Ar, EM_cur);
    END
    Ar = ArchiveReduction(Ar, EM_cur);
    Global best update and elitist preservation process;
  END
END
```

for the selection operation, the commonly used roulette selection is adopted based on the reciprocal of the fitness values. The offspring reproduction process includes two operations: 1) crossover and 2) mutation. For the crossover operation, we adopt the two-point crossover operation that randomly selects two points (i.e., alleles) in the bit string (i.e., chromosome) to act as the beginning point and the ending point of the segment. Then, the segments of the bit strings are exchanged. For the mutation operation, we adopt the simple mutation operation that randomly selects a point in the bit string and then turns it into the opposite value. To improve search efficiency, the elitist preservation strategy is supplemented after the evaluation of the population and global best update process. Finally, the whole algorithm is given as Algorithm 6.

IV. Experimental Studies

A. Problem Instances and Parameter Settings

In the proposed model of CVRP-OC, the problem instances from the CVRP which have been widely studied and tested can be used to form the test instances for the CVRP-OC. All the test instances can be downloaded on the website http://akira.ruc.dk/~keld/research/LKH-3/. The basic components of the CVRP instances are the weight matrix for the connected edges, the capacity of the vehicles, the demand for each customer, and the maximal number of vehicles available. Unlike the CVRP, the customers in the CVRP-OC instance are served by the PF and the OC. The optimization task is to search for a crowdshipping delivery plan (i.e., solution) of which the total cost is as small as possible. Notably, whenever the assignment is fixed, the routing problem for the customers assigned to the PF which are a subset of the customers in the instance is a CVRP with a smaller size than its corresponding CVRP-OC. If all customers are assigned to the PF, the routing problem makes no difference to the original instance. Hence, CVRP-OC is a variant of the CVRP with a much larger solution space. The tested instances are divided into medium-size ones with 25 instances and large-size ones with 10 instances, which are shown in Table I. The key features are manifested in the instance name with the first letter indicating the abbreviation of the test suite, n (i.e., N in the CVRP-OC model) indicating the total number of customers, and k (i.e., K in the CVRP-OC model) indicating the total number of vehicles. To allow the algorithms to fully perform the search ability, the stopping criterion is set to be the maximum running time. The running time is 300 s for the medium-size instances and 600 s for the large-size instances. The experiments are carried out on the machine of Intel Xeon CPU E3-1225 v5 @3.30 GHz. Each algorithm runs 30 independent trials on each instance to obtain the average result to reduce random statistical error.

TABLE I
CVRP-OC Problem Instances

MEDIUM-SIZE					LARGE-SIZE	
A-n55-k9	B-n50-k8	E-n51-k5	F-n45-k4	P-n60-k15	X-n157-k13	X-n303-k21
A-n60-k9	B-n57-k7	E-n76-k7	F-n72-k4	P-n65-k10	X-n190-k8	X-n331-k15
A-n65-k9	B-n63-k10	E-n76-k15	F-n135-k7	P-n70-k10	X-n214-k11	X-n359-k29
A-n69-k9	B-n67-k10	E-n101-k8	P-n51-k10	P-n76-k5	X-n237-k14	X-n411-k19
A-n80-k10	B-n78-k10	E-n101-k14	P-n55-k15	P-n101-k4	X-n261-k13	X-n459-k26

For simplicity, as assumed in Section II-B, the simple compensation scheme that is linearly dependent on the distance from the depot to the destination is adopted. Moreover, the compensation coefficient ρ in (1) is set to be 0.8 which is relatively high. In such a way, it is possible to increase the problem difficulty for the optimization process since with a higher coefficient more customers tend to be served by PF. As a result, the evaluation for the assignment tends to be more computationally expensive due to the larger size CVRP for solvers (i.e., EM) to solve. Generally speaking, setting a higher compensation coefficient ρ poses a greater challenge of balancing computational cost and solution quality to the EC algorithms.

The neighborhood distance *nd* for the NBE strategy is set as

$$nd = \begin{cases} N/2, & \text{if EM} = ACS + 2\text{opt} \\ N/5, & \text{if EM} = VNS + 2\text{opt} \\ N/10, & \text{otherwise} \end{cases} \tag{11}$$

where N is the total number of customers. Since a larger *nd* tends to activate the NBE strategy to help the evaluation process, a larger *nd* is required in HSEM to fully make use of the evaluated *NBS* because using HSEM too frequently is economically unacceptable. On the contrary, setting a smaller value of *nd* for LSEM might cause more individuals to be evaluated by the EM. This way, individuals are less likely to fall in the neighborhood of solutions in the archive and the diversity can be maintained with the uncertainty of the LSEM. The population size *ps* is 50 for GA-SAFE and all the other peer algorithms. Two SAFE topology structures of different EM

TABLE II
EXPERIMENTAL RESULTS OF GA-SAFES AND THE BASELINE ALGORITHMS ON MEDIUM-SIZE INSTANCES

Algorithm	GA-Greed-1		GA-Greed-N		GA-VNS		GA-ACS		GA-SAFE-I		GA-SAFE-II	
Instance	best	mean	best	mean	best	mean	best	mean	best	mean	best	mean
A-n55-k9	1082.51	1136.02	1077.31	1128.00	1082.55	1134.71	1074.43	1135.50	1080.46	1130.33	**1062.03**	**1099.47**
A-n60-k9	1385.24	1461.29	1385.54	1455.72	1365.58	1442.53	1396.43	1471.04	1364.30	1422.17	**1330.79**	**1380.52**
A-n65-k9	1289.04	1358.10	1290.45	1335.35	1249.77	1323.36	1285.98	1356.65	1259.30	1331.97	**1246.48**	**1288.20**
A-n69-k9	1253.19	1295.57	1228.31	1279.71	1192.62	1270.69	1228.90	1277.25	1189.49	1270.68	**1166.19**	**1231.33**
A-n80-k10	1956.49	2072.62	1938.79	2056.37	1955.92	2063.69	1957.97	2104.38	1896.96	1960.58	**1864.67**	**1917.90**
B-n50-k8	1220.44	1282.55	1222.83	1275.87	1221.89	1280.19	1235.14	1294.78	**1211.36**	1254.35	1213.89	**1232.68**
B-n57-k7	1150.24	1296.25	1153.03	1263.59	1153.59	1242.85	1167.82	1263.53	1110.29	1228.90	**1106.89**	**1160.98**
B-n63-k10	1541.99	1626.45	1518.64	1597.98	1521.69	1610.10	1552.05	1637.79	1492.00	1573.60	**1472.80**	**1530.18**
B-n67-k10	1084.40	1147.12	1101.98	1137.33	1086.61	1144.50	1084.36	1146.71	1078.54	1134.13	**1063.90**	**1100.10**
B-n78-k10	1324.30	1407.90	1315.96	1375.87	1312.63	1376.13	1316.04	1387.42	1295.05	1356.21	**1252.74**	**1314.32**
E-n51-k5	545.70	573.79	538.76	565.49	533.93	559.22	538.79	578.76	534.42	563.61	**530.63**	**554.59**
E-n76-k7	750.06	798.79	746.29	781.97	730.01	772.70	748.97	789.65	725.68	777.54	**700.76**	**746.39**
E-n76-k15	1032.32	1072.71	1037.66	1067.70	**1030.72**	1061.63	1042.85	1072.28	1044.95	1068.73	1032.91	**1052.40**
E-n101-k8	948.00	1009.96	950.25	1012.53	936.10	994.15	937.59	998.62	890.90	980.25	**880.86**	**949.08**
E-n101-k14	1207.86	1275.68	1173.35	1248.02	1187.18	1242.18	1196.76	1235.13	**1161.06**	1225.19	1171.41	**1211.79**
F-n45-k4	658.56	680.79	650.33	666.71	**644.19**	664.72	644.25	669.63	**644.19**	676.23	645.28	**661.47**
F-n72-k4	260.46	281.97	259.25	281.05	243.19	268.08	241.04	275.53	**240.73**	267.11	**240.73**	**257.33**
F-n135-k7	1287.58	1406.69	1308.90	1383.12	1262.36	1366.19	**1187.11**	1351.56	1246.68	1335.33	1217.72	**1300.99**
P-n51-k10	719.32	741.71	719.91	739.19	720.16	740.04	724.01	741.56	**714.11**	733.53	716.04	**730.22**
P-n55-k15	852.47	864.09	848.61	861.87	849.02	863.62	852.96	865.74	848.61	859.19	**847.13**	**857.36**
P-n60-k15	917.75	936.31	917.54	938.71	924.09	934.91	920.54	938.55	**911.94**	929.32	915.38	**929.16**
P-n65-k10	819.05	859.31	818.59	854.46	819.13	852.54	**809.65**	852.27	828.15	855.18	821.29	**841.09**
P-n70-k10	866.16	902.64	870.59	901.46	872.25	900.11	873.48	903.48	865.26	898.25	**844.06**	**876.67**
P-n76-k5	678.32	733.59	688.04	729.12	683.04	725.06	698.54	731.13	669.18	732.61	**666.71**	**698.51**
P-n101-k4	798.10	841.41	788.49	828.82	786.29	816.78	795.88	829.71	**699.99**	785.15	704.42	**779.91**

switch strategies (i.e., one-way and two-way EM switch strategies) are adopted to form two variants of GA-SAFE, which are named GA-SAFE-I and GA-SAFE-II, respectively. The parameter η of the switch condition, which indicates the consecutive generations of the population without improvement, is set to 20. For the CVRP solvers $VNS+2opt$ and $ACS+2opt$, the initialization scheme indicates the method to construct a solution that is used as the initial solution and to initialize the pheromone matrix. The initialization scheme for solvers $VNS + 2opt$ and $ACS + 2opt$ is set to $Greed + 2opt - N$ to obtain a good initial solution. The rest of the parameters for the proposed algorithm GA-SAFE and CVRP solvers representing various EMs are shown in Table S.II in the supplementary material.

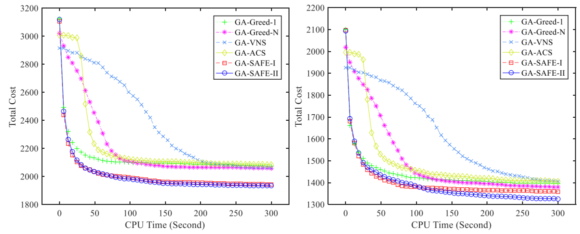
B. Comparisons With Some Baseline Algorithms

In this section, the proposed algorithms GA-SAFE-I and GA-SAFE-II are compared to the GA variants which adopt the unique EM throughout the whole optimization process. These GA variants are named GA-Greed-1, GA-Greed-N, GA-VNS, and GA-ACS. They, respectively, use the CVRP solvers $Greed+2opt-1$, $Greed+2opt-N$, $VNS+2opt$, and $ACS+2opt$ to plan the routes for PF when evaluating assignments, respectively, which actually can be regarded as four baseline algorithms. To enable a fair comparison, the NBE strategy is preserved in all the algorithms. Since the peer algorithms use the NBE strategy with a single EM, the archive only stores the *NBS* for one EM and the archive migration operation is not needed.

The results on the medium-size instances are shown and comparedin Table II. The experimental results show that GAs with SAFE (i.e., GA-SAFE-I and GA-SAFE-II) generally outperform the ones without SAFE, such as GA-Greed-1. The experimental results show that the SAFE method for GA brings much higher quality solutions to the problem. In terms of the mean value of the total cost shown in Table II, GA-SAFE-II reaches the best performance on 23 instances. As for the best value of the total cost, either GA-SAFE-I or GA-SAFE-II ranks the first place on 21 instances. Especially on instances with larger sizes, such as A-n80-k10 and P-n101-k4, GA-SAFEs perform significantly better than GA variants without SAFE.

Notably, GA-VNS appears to be rather competitive among the GA variants without SAFE. The reason might attribute to the similarity to the traditional neighborhood searching techniques, such as VNS. Researches show that such neighborhood search techniques perform well on medium-size problems [54], [55]. Also, the experimental results of four solvers on CVRP instances in Table S.I in the supplementary material support this conclusion. Hence, due to the low computational cost of the transformation operations, GA-VNS appears to be outstanding among GA variants without SAFE on medium-size problems.

In addition, Wilcoxon's rank sum test on the medium-size instances is also conducted to show the significance level of comparison between the GA-SAFEs and four peer algorithms. The comparison results are shown in Table S.III of the supplementary material. It is apparent that GA-SAFE-II significantly outperforms GA-Greed-1,



(a) (b)

Fig. 7. Convergence curve of the algorithms on CVRP-OC instances. (a) A-n80-k10. (b) B-n78-k10.

GA-Greed-N, and GA-ACS on all instances. Also, the performance of GA-SAFE-I is significantly better than or similar to that of GA variants without SAFE, showing the effectiveness of the SAFE method. Compared to the competitive algorithm GA-VNS, the GA-SAFE-I is significantly better than GA-VNS on 7 out of 25 instances and gets a draw on the rest of the instances while the GA-SAFE-II significantly outperforms GA-VNS on 24 out of 25 instances.

As for the large-size instances, the mean and the best values of the total cost of 30 independent trials are shown in Table S.IV and their Wilcoxon's rank sum tests are shown in Table S.V in the supplementary material. It can be observed from Tables S.IV and S.V that both GA-SAFE-I and GA-SAFE-II substantially surpass the peer algorithms and the advantage of the SAFE method on large-size instances is even more obvious than that on medium-size instances. Moreover, on the large-size instances, the GA-SAFE-II performs slightly better than GA-SAFE-I on 9 out of 10 instances. Hence, it seems that GA with SAFE is superior to the ones without SAFE and GA-SAFE-II is superior to GA-SAFE-I.

Furthermore, Fig. 7 shows the convergence curve of the compared algorithms on two representative instances, and some of the others are plotted in Fig S.1 in the supplementary material. The figures show that our proposed GA-SAFE algorithms converge faster than the compared ones. More importantly, our GA-SAFEs achieve better final results in terms of solution quality.

C. Comparisons With the State-of-the-Art Algorithm

The multistart heuristic algorithm MATHOD proposed in [43] can be used as a state-of-the-art compared algorithm. Since the mathematical model in our paper is slightly different from the one in [43], the original MATHOD can not be directly applied to the CVRP-OC. Some modifications have been made to the original MATHOD. Specifically, the integer programming process in MATHOD for the "JUMP" process is replaced by a simple random reassignment on the customers' order. Also, the initialization process in the MATHOD uses the same way as the GA-SAFE does. To ensure fairness, other parameter settings and the algorithm process is consistent with the literature. Finally, the modified version of the algorithm is denoted as MATHOD-M. The results, including best and mean cost are shown in Table S.VI of the supplementary material. Moreover, the results of Wilcoxon rank sum test for the comparisons are shown in Table III. It seems that our proposed algorithm

TABLE III
WILCOXON RANK SUM TEST BETWEEN GA-SAFE AND THE STATE-OF-THE-ART ALGORITHM MATHOD-M ON MEDIUM-SIZE INSTANCES

Proposed Algorithm	GA-SAFE-I	GA-SAFE-II
Peer algorithm	MATHOD-M	MATHOD-M
GA-SAFE is better (+)	11	21
=	11	3
GA-SAFE is worse (-)	3	1

TABLE IV
COMPARISONS BETWEEN GA-SAFE AND THE GA VARIANTS WITH DIFFERENT EM SWITCH STRATEGIES

Algorithm	Peer Algorithm	Best (Win/Lose)	Mean (Win/Lose)
GA-SAFE-I	GA-Seq-EM	15/10	21/4
GA-SAFE-II	GA-Rand-EM	14/11	21/4

GA-SAFE offers great advantages in terms of solution quality compared to the MATHOD-M algorithm.

D. Effects of the Switch Strategies in SAFE

In this section, we are going to investigate the effects of the one-way and two-way EM switch strategies. To achieve this aim, two algorithms, which are GA-Seq-EM and GA-Rand-EM, are designed to compare with GA-SAFEs. The compared algorithms also use multiple EMs during the optimization process but have differences with the SAFE method in the EM switch mechanism. To enable a fair comparison, the other components, such as EM switch condition (i.e., stagnation) remain the same as GA-SAFE in the compared algorithms. For the GA-Seq-EM, the EM is switched from EM_1 to EM_4 in a sequential scheme without the switchback mechanism. Therefore, it is compared with the GA-SAFE-I to validate the benefits of the one-way switch strategy. For the GA-Rand-EM, the EM is randomly selected from four available EMs when the switch condition is met. Since the GA-Rand-EM allows switchback, it is compared with the GA-SAFE-II to validate the benefits of the two-way switch strategy. The experimental results on 25 tested instances are summarized and shown in Table IV. It can be seen that the one-way switch strategy is quite useful in managing multiple EMs compared to the GA-Seq-EM. The GA-SAFE-I surpasses the GA-Seq-EM on 15 out of the 25 instances in terms of the best result and on 21 instances in terms of the mean result. Moreover, the benefit of the two-way switch strategy is verified on the mean and best results when compared with GA-Rand-EM, showing that GA-SAFE-II wins on most of the tested instances.

E. Effects of the NBE Strategy

In this section, the contribution of the NBE strategy is verified. For the GA-SAFE, the NBE strategy is removed from GA-SAFE to see how well it performs. Also, for the GAs without SAFE, such as GA-Greed-1, the NBE strategy is removed. Subsequently, they form simple versions of GA in which the original EM (i.e., Algorithm 1) is used to evaluate all the assignments (i.e., solution) no matter how much computational time the CVRP solver costs. The results are shown in Table S.VII of the supplementary material. The reported result is the average total cost of the 30 independent trials. Moreover, the results of Wilcoxon's rank sum test are shown in Table S.VIII of the supplementary material. It can be seen that the NBE strategy has a great promotion on the GA variants with relatively HSEM (e.g., GA-Greed-N, GA-VNS, and GA-ACS). They significantly surpass their corresponding competitors without NBE strategy (i.e., GA-Greed-N-w/o-NBE, GA-VNS-w/o-NBE, and GA-ACS-w/o-NBE) at the significance level of 0.05, on all the medium-size instances. However, the benefit of the NBE strategy on a relatively LSEM (e.g., EM_1) is not so remarkable. The GA-Greed-1 performs significantly better than the one without NBE on 12 out of the 25 medium-size instances, while gets a draw on 10 instances. More importantly, both GA-SAFE-I and GA-SAFE-II obtain significantly better results than their versions without the NBE strategy. The results reveal that the NBE strategy benefits the GA-SAFE greatly.

The reason for achieving more improvement on HSEM (e.g., GA-ACS) than LSEM (e.g., GA-Greed-1) by the NBE strategy can be summarized as two parts. On the one hand, the assignment evaluated by the HSEMs is more likely to share some common good structures while the total cost obtained by LSEM is a rough and low-accuracy estimation. As a result, the NBE might not be profitable for GA-Greed-1 due to the accuracy bottleneck. On the other hand, the NBE strategy reuses historical solutions stored in the archive to help the FE process, and the heavy computational cost caused by the HSEM is reduced. As a result, the benefit of the HSEM is exploited, and it leads to a better improvement.

F. Parameter Investigation of GA-SAFE

The parameter stagnation η (i.e., number of consecutive generations without improvement) of the switch condition in the SAFE method is investigated in this section. The results of the average total cost of 30 independent trials are shown in Table S.IX in the supplementary material. For each instance, the best results among GA-SAFEs with different parameter settings are in *boldface*. Generally, it can be seen that the average results obtained by GA-SAFEs with different settings of $\eta = 10$, 20, 30, 40, and 50 are rather similar to each other. It means that the performance of GA-SAFE is not affected by the parameter η seriously. The times that GA-SAFEs with different parameters get the first place among all tested instances are counted. GA-SAFE-I with parameter settings of 10, 20, and 30 for η gets first place on 13, 6, and 3 instances while GA-SAFE-II with parameter settings of 20, 30, and 40 for η gets first place on 6, 7, and 6 instances, respectively. Hence, $\eta = 20$ tends to be a suitable parameter setting for the algorithm. Since no significant differences happen among the parameter settings, the parameter of 20 is chosen as the default value as Section IV-B shows.

G. Discussions

Experimental comparisons verify that the SAFE and NBE indeed help the GAs perform better than the GA variants without them on almost all tested instances, in terms of solution quality within a period in solving CVRP-OC. The GA-SAFEs offer a better performance not only in medium-size instances for CVRP-OC, but also show to be competitive on large-size instances owing to the SAFE method that can automatically adapt the EM in the optimization process and the NBE strategy which makes full use of the high-quality evaluated solutions to enhance search efficiency.

Comparing GA-SAFE-I with GA-SAFE-II, it is seen that on medium-size instances and large-size instances GA-SAFE-II is better because the switchback mechanism to the LSEM in the two-way EM switch strategy offers the population a chance to locate another promising area. Consequently, HSEM can be adopted again to promote the current population.

Moreover, the consumed FEs of different EMs and the execution counts of NBE in the algorithms are recorded and summarized as the total FEs in Table S.X in the supplementary material. It can be seen that the algorithms only using LSEM, such as GA-Greed-1 take much more FEs than that using HSEM, such as GA-ACS in a fixed running time due to the cheaper computational cost of LSEM in each FE. Besides, our proposed GA-SAFEs use the least FEs to evolve and achieve better final results due to the adaptive EM management strategies in the SAFE method. Furthermore, the execution counts of NBE occupy a large proportion of the total FEs in all algorithms, showing that the expensive evaluations by the original EM are replaced and the computational cost is reduced by the NBE strategy.

To further investigate the benefits of the SAFE method on other EC algorithms, we test the SAFE method on the estimation of distribution algorithm (EDA) [59], [60]. The results are shown in Table S.XI and their Wilcoxon's rank sum tests are shown in Table S.XII in the supplementary material. It can be seen that both the EDA-SAFE-I and EDA-SAFE-II are significantly better than the compared algorithms on most instances.

V. CONCLUSION

In this article, the crowdshipping scheduling problem is carried out to validate the performance of the novel SAFE method in flexibly and efficiently solving EOPs. Different from the existing methods that usually use the surrogate model to replace the original objective function, the proposed SAFE method is based on multiple EMs and the fitness function remains unchanged as the original objective function. The CRVP-OC crowdshipping model is built for the case study, followed by the development of the GA-based optimization algorithm with inter-EM management and intra-EM improvement. The SAFE method, which refers to inter-EM management, takes advantage of various EMs throughout the optimization process. Two EM switch strategies (i.e., one-way switch strategy and two-way switch strategy) are proposed. The one-way EM switch strategy is designed to adopt LSEM to find a promising region and to use HSEM to further distinguish the individuals in the population. The two-way switch strategy makes an improvement on the one-way switch strategy that the uncertainty and computationally cheap features of LSEM are made use of to search for another promising area. With the switchback mechanism in the two-way

EM switch strategy, the GA-SAFE-II can find better solutions than GA-SAFE-I on most of the medium-size and large-size instances.

Moreover, the NBE strategy, which refers to intra-EM improvement, is designed to make full use of the high-quality solutions evaluated by the EM. Therefore, two operations, OpExclude and OpJoin, are used to evaluate incoming new solutions by some known and good solutions. Consequently, the NBE strategy reduces the computational cost and improves the performance of the optimization algorithm. An archive is proposed to store the *NBS* for each EM incorporating with archive management operations, including archive insertion, archive reduction, and archive migration.

Extensive experimental tests have been conducted on 25 medium-size and 10 large-size instances. The algorithms GA-SAFE-I and GA-SAFE-II both significantly outperform other GA variants on most of the instances tested, irrespective of whether they are medium size or large size based on the criteria of solution quality. The results show the high effectiveness and high efficiency of the SAFE method and the NBE strategy. From the case study of crowdshipping, the importance of EM management is shown, and the benefit brought by automatic management is testified with experiments. Furthermore, the NBE strategy is an example to design an evaluation method based on the correlation or similarity between the known high-quality solutions and the solutions to be evaluated. The NBE strategy also shows a possible direction for solving EOPs.

In the proposed SAFE method, we only use multiple available EMs, which can be considered as a kind of knowledge, to improve the search ability of the EC algorithms. However, the historical data available in the real-world applications and the evolutionary process has not yet been fully utilized. Therefore, combining the advantages of knowledge and data to further improve the EC algorithms can be promising future work.

References

[1] X.-Y. Zhang, J. Zhang, Y.-J. Gong, Z.-H. Zhan, W.-N. Chen, and Y. Li, "Kuhn–Munkres parallel genetic algorithm for the set cover problem and its application to large-scale wireless sensor networks," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 695–710, Oct. 2016.

[2] Y.-F. Ge *et al.*, "Distributed memetic algorithm for outsourced database fragmentation," *IEEE Trans. Cybern.*, early access, Nov. 4, 2020, doi: 10.1109/TCYB.2020.3027962

[3] T. Huang, Y.-J. Gong, Y.-H. Zhang, Z.-H. Zhan, and J. Zhang, "Automatic planning of multiple itineraries: A Niching genetic evolution approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4225–4240, Oct. 2020.

[4] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.

[5] D. Liang, Z.-H. Zhan, Y. Zhang, and J. Zhang, "An efficient ant colony system approach for new energy vehicle dispatch problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4784–4797, Nov. 2020.

[6] Z.-G. Chen *et al.*, "Multiobjective cloud workflow scheduling: A multiple populations ant colony system approach," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2912–2926, Aug. 2019.

[7] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, 1995, pp. 1942–1948.

[8] X.-F. Liu, Z.-H. Zhan, Y. Gao, J. Zhang, S. Kwong, and J. Zhang, "Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 587–602, Aug. 2019.

[9] Z.-J. Wang, Z.-H. Zhan, S. Kwong, H. Jin, and J. Zhang, "Adaptive granularity learning distributed particle swarm optimization for large-scale optimization," *IEEE Trans. Cybern.*, early access, Mar. 27, 2020, doi: 10.1109/TCYB.2020.2977956.

[10] Z. J. Wang *et al.*, "Automatic niching differential evolution with contour prediction approach for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 114–128, Feb. 2020.

[11] Z.-H. Zhan, Z.-J. Wang, H. Jin, and J. Zhang, "Adaptive distributed differential evolution," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4633–4647, Nov. 2020.

[12] Z. G. Chen, Z. H. Zhan, H. Wang, and J. Zhang, "Distributed individuals for multiple peaks: A novel differential evolution for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 708–719, Aug. 2020.

[13] H. C. Lau, T. Chan, and W. Tsui, "Item-location assignment using fuzzy logic guided genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 765–780, Dec. 2008.

[14] S. Z. Zhou, Z. H. Zhan, Z. G. Chen, S. Kwong, and J. Zhang, "A multiobjective ant colony system algorithm for airline crew rostering problem with fairness and satisfaction," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 765–780, Dec. 2008, doi: 10.1109/TITS.2020.2994779.

[15] X. Zhang, K. J. Du, Z. H. Zhan, S. Kwong, T. L. Gu, and J. Zhang, "Cooperative co-evolutionary bare-bones particle swarm optimization with function independent decomposition for large-scale supply chain network design with uncertainties," *IEEE Trans. Cybern.*, vol. 50, no. 10, pp. 4454–4468, Oct. 2020.

[16] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving deep convolutional neural networks for image classification," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 394–407, Apr. 2020.

[17] M. A. J. Ghasab, S. Khamis, F. Mohammad, and H. J. Fariman, "Feature decision-making ant colony optimization system for an automated recognition of plant species," *Exp. Syst. Appl.*, vol. 42, no. 5, pp. 2361–2370, 2015.

[18] Y. Guo, J. Y. Li, and Z. H. Zhan, "Efficient hyperparameter optimization for convolution neural networks in deep learning: A distributed particle swarm optimization approach," *Cybern. Syst.*, vol. 52, no. 1, pp. 36-57, 2021, doi: 10.1080/01969722.2020.1827797.

[19] J. F. Gonçalves, J. J. de Magalhães Mendes, and M. C. G. Resende, "A hybrid genetic algorithm for the job shop scheduling problem," *Eur. J. Oper. Res.*, vol. 167, no. 1, pp. 77–95, 2005.

[20] Z. J. Wang *et al.*, "Dynamic group learning distributed particle swarm optimization for large-scale optimization and its application in cloud workflow scheduling," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2715–2729, Jun. 2020.

[21] G. Zhang, X. Shao, P. Li, and L. Gao, "An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem," *Comput. Ind. Eng.*, vol. 56, no. 4, pp. 1309–1318, 2009.

[22] S. S. Sebtahmadi, H. B. Azad, S. H. A. Kaboli, M. D. Islam, and S. Mekhilef, "A PSO-DQ current control scheme for performance enhancement of Z-source matrix converter to drive IM fed by abnormal voltage," *IEEE Trans. Power Electron.*, vol. 33, no. 2, pp. 1666–1681, Feb. 2018.

[23] P. Díaz, M. Pérez-Cisneros, E. Cuevas, O. Camarena, F. A. Fausto Martinez, and A. González, "A swarm approach for improving voltage profiles and reduce power loss on electrical distribution networks," *IEEE Access*, vol. 6, pp. 49498–49512, 2018.

[24] E. Cuevas, E. B. Espejo, and A. C. Enríquez, "Clustering representative electricity load data using a particle swarm optimization algorithm," in *Metaheuristics Algorithms in Power Systems*. Heidelberg, Germany: Springer, 2019, pp. 187–210.

[25] Z.-H. Zhan *et al.*, "Cloudde: A heterogeneous differential evolution algorithm and its distributed cloud version," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 3, pp. 704–716, Mar. 2017.

[26] A. Rafieerad *et al.*, "Toward improved mechanical, tribological, corrosion and in-vitro bioactivity properties of mixed oxide nanotubes on Ti–6Al–7Nb implant using multi-objective PSO," *J. Mech. Behav. Biomed. Mater.*, vol. 69, pp. 1–18, May 2017.

[27] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. Philadelphia, PA, USA: SIAM, 2019.

[28] J. Zhang, X. Zhu, Y. Wang, and M. Zhou, "Dual-environmental particle swarm optimizer in noisy and noise-free environments," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2011–2021, Jun. 2019.

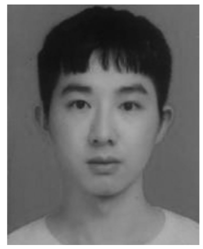
[29] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 442–458, Jun. 2019.

[30] J. Y. Li, Z. H. Zhan, H. Wang, and J. Zhang, "Data-driven evolutionary algorithm with perturbation-based ensemble surrogates," *IEEE Trans. Cybern.*, early access, Aug. 10, 2020, doi: 10.1109/TCYB.2020.3008280.

[31] H. Wang, Y. Jin, and J. O. Jansen, "Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 939–952, Dec. 2016.

[32] W. Luo, R. Yi, B. Yang, and P. Xu, "Surrogate-assisted evolutionary framework for data-driven dynamic optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 2, pp. 137–150, Apr. 2019.

[33] D. Buche, N. N. Schraudolph, and P. Koumoutsakos, "Accelerating evolutionary algorithms with Gaussian process fitness function models," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 35, no. 2, pp. 183–194, May 2005.
[34] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Trans. Evol. Comput.*, vol. 6, no. 5, pp. 481–494, Oct. 2002.
[35] J. Y. Li, Z. H. Zhan, C. Wang, H. Jin, and J. Zhang, "Boosting data-driven evolutionary algorithm with localized data generation," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 923–937, Oct. 2020.
[36] J. Y. Jung, G. Blau, J. F. Pekny, G. V. Reklaitis, and D. Eversdyk, "A simulation based optimization approach to supply chain management under demand uncertainty," *Comput. Chem. Eng.*, vol. 28, no. 10, pp. 2087–2106, 2004.
[37] P. Marbach and J. N. Tsitsiklis, "Simulation-based optimization of Markov reward processes," *IEEE Trans. Autom. Control*, vol. 46, no. 2, pp. 191–209, Feb. 2001.
[38] S. Koziel and S. Ogurtsov, "Model management for cost-efficient surrogate-based optimisation of antennas using variable-fidelity electromagnetic simulations," *IET Microw. Antennas Propag.*, vol. 6, no. 15, pp. 1643–1650, 2012.
[39] H. Wang, Y. Jin, C. Sun, and J. Doherty, "Offline data-driven evolutionary optimization using selective surrogate ensembles," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 203–216, Apr. 2019.
[40] H. Wang, Y. Jin, and J. Doherty, "A generic test suite for evolutionary multifidelity optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 836–850, Dec. 2018.
[41] A. Habib, H. K. Singh, and T. Ray, "A multiple surrogate assisted multi/many-objective multi-fidelity evolutionary algorithm," *Inf. Sci.*, vol. 502, pp. 537–557, Oct. 2019.
[42] Z. Liu, K. S. Bhattacharjee, F.-B. Tian, J. Young, T. Ray, and J. C. Lai, "Kinematic optimization of a flapping foil power generator using a multi-fidelity evolutionary algorithm," *Renew. Energy*, vol. 132, pp. 543–557, Mar. 2019.
[43] C. Archetti, M. Savelsbergh, and M. G. Speranza, "The vehicle routing problem with occasional drivers," *Eur. J. Oper. Res.*, vol. 254, no. 2, pp. 472–480, 2016.
[44] E. Marcucci, M. Le Pira, C. S. Carrocci, V. Gatta, and E. Pieralice, "Connected shared mobility for passengers and freight: Investigating the potential of crowdshipping in urban areas," in *Proc. 5th IEEE Int. Conf. Models Technol. Intell. Transp. Syst.*, 2017, spp. 839–843.
[45] T. V. Le and S. V. Ukkusuri, "Crowd-shipping services for last mile delivery: Analysis from survey data in two countries," 2018. [Online]. Available: arXiv:1810.02856.
[46] K. Gdowska, A. Viana, and J. P. Pedroso,"Stochastic last-mile delivery with crowdshipping," *Transp. Res. Procedia*, vol. 30, pp. 90–100, Mar. 2018.
[47] A. Punel, A. Ermagun, and A. Stathopoulos, "Studying determinants of crowd-shipping use," *Travel Behav. Soc.*, vol. 12, pp. 30–40, Jul. 2018.
[48] T. K. Ralphs, L. Kopman, W. R. Pulleyblank, and L. E. Trotter, "On the capacitated vehicle routing problem," *Math. Program.*, vol. 94, nos. 2–3, pp. 343–359, 2003.
[49] G. Laporte and Y. Nobert, "Exact algorithms for the vehicle routing problem," *North-Holland Mathematics Studies*, vol. 132. New York, NY, USA: Elsevier, 1987, pp. 147–184.
[50] R. Baldacci and A. Mingozzi, "A unified exact method for solving different classes of vehicle routing problems," *Math. Program.*, vol. 120, no. 2, p. 347, 2009.
[51] M. Fischetti, P. Toth, and D. Vigo, "A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs," *Oper. Res.*, vol. 42, no. 5, pp. 846–859, 1994.
[52] E. Choi and D.-W. Tcha, "A column generation approach to the heterogeneous fleet vehicle routing problem," *Comput. Oper. Res.*, vol. 34, no. 7, pp. 2080–2095, 2007
[53] J. Kytöjoki, T. Nuortio, O. Bräysy, and M. Gendreau, "An efficient variable neighborhood search heuristic for very large scale vehicle routing problems," *Comput. Oper. Res.*, vol. 34, no. 9, pp. 2743–2757, 2007.
[54] N. Mladenović and P. Hansen, "Variable neighborhood search," *Comput. Oper. Res.*, vol. 24, no. 11, pp. 1097–1100, 1997.
[55] X. Meng, J. Li, X. Dai, and J. Dou, "Variable neighborhood search for a colored traveling salesman problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1018–1026, Apr. 2018.
[56] G. A. Croes, "A method for solving traveling-salesman problems," *Oper. Res.*, vol. 6, no. 6, pp. 791–812, 1958.
[57] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.
[58] M. Tan and A. Qiu, "Large deformation multiresolution diffeomorphic metric mapping for multiresolution cortical surfaces: A coarse-to-fine approach," *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4061–4074, Sep. 2016.
[59] Z.-G. Chen, Y. Lin, Y.-J. Gong, Z.-H. Zhan, and J. Zhang, "Maximizing lifetime of range-adjustable wireless sensor networks: A neighborhood-based estimation of distribution algorithm," *IEEE Trans. Cybern.*, early access, Apr. 1, 2020, doi: 10.1109/TCYB.2020.2977858.
[60] E. Cuevas, A. Rodríguez, A. Valdivia, D. Zaldívar, and M. Pérez, "A hybrid evolutionary approach based on the invasive weed optimization and estimation distribution algorithms," *Soft Comput.*, vol. 23, no. 24, pp. 13627–13668, 2019.



Sheng-Hao Wu (Student Member, IEEE) received the B.S. degree in computer science and technology from the South China University of Technology, Guangzhou, China, in 2019, where he is currently pursuing the Ph.D. degree in computer science and technology with the School of Computer Science and Engineering.

His research interests mainly include computational intelligence, machine learning, and their applications in real-world problems, and in environments of distributed computing and big data.

Zhi-Hui Zhan (Senior Member, IEEE) received the bachelor's and Ph.D. degrees in computer science from Sun Yat-Sen University, Guangzhou, China, in 2007 and 2013, respectively.

He is currently the Changjiang Scholar Young Professor and the Pearl River Scholar Young Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. His current research interests include evolutionary computation algorithms, swarm intelligence algorithms, and their applications in real-world problems, and in environments of cloud computing and big data.

Prof. Zhan's Doctoral Dissertation was awarded the IEEE Computational Intelligence Society Outstanding Ph.D. Dissertation and the China Computer Federation Outstanding Ph.D. Dissertation. He was a recipient of the Outstanding Youth Science Foundation from the National Natural Science Foundations of China in 2018 and the Wu Wen-Jun Artificial Intelligence Excellent Youth from the Chinese Association of Artificial Intelligence in 2017. He is listed as one of the Most Cited Chinese Researchers in Computer Science. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the *Neurocomputing*.

Jun Zhang (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the City University of Hong Kong, Hong Kong, in 2002.

He is currently a Korea Brain Pool Fellow Professor with Hanyang University, Ansan, South Korea, and a Visiting Scholar with Victoria University, Melbourne, VIC, Australia. His current research interests include computational intelligence, cloud computing, high performance computing, operations research, and power electronic circuits.

Prof. Zhang was a recipient of the Changjiang Chair Professor from the Ministry of Education, China, in 2013, the China National Funds of Distinguished Young Scientists from the National Natural Science Foundation of China in 2011, and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE TRANSACTIONS ON CYBERNETICS.