Multisurrogate-Assisted Ant Colony Optimization for Expensive Optimization Problems With Continuous and Categorical Variables

Jiao Liu, Yong Wang, *Senior Member, IEEE*, Guangyong Sun, and Tong Pang

***Abstract*—As an effective optimization tool for expensive optimization problems (EOPs), surrogate-assisted evolutionary algorithms (SAEAs) have been widely studied in recent years. However, most current SAEAs are designed for continuous/combinatorial EOPs, which are not suitable for mixed-variable EOPs. This article focuses on one kind of mixed-variable EOP: EOPs with continuous and categorical variables (EOPCCVs). A multisurrogate-assisted ant colony optimization algorithm (MiSACO) is proposed to solve EOPCCVs. MiSACO contains two main strategies: 1) multisurrogate-assisted selection and 2) surrogate-assisted local search. In the former, the radial basis function (RBF) and least-squares boosting tree (LSBT) are employed as the surrogate models. Afterward, three selection operators (i.e., RBF-based selection, LSBT-based selection, and random selection) are devised to select three solutions from the offspring solutions generated by ACO, with the aim of coping with different types of EOPCCVs robustly and preventing the algorithm from being misled by inaccurate surrogate models. In the latter, sequence quadratic optimization, coupled with RBF, is utilized to refine the continuous variables of the best solution found so far. By combining these two strategies, MiSACO can solve EOPCCVs with limited function evaluations. Three sets of test problems and two real-world cases are used to verify the effectiveness of MiSACO. The results demonstrate that MiSACO performs well in solving EOPCCVs.**

***Index Terms*—Ant colony optimization (ACO), categorical variables, continuous variables, mixed-variable expensive optimization problems (EOPs), surrogate-assisted evolutionary algorithms (SAEAs).**

I. Introduction

A. Expensive Optimization Problems With Continuous and Categorical Variables

EXPENSIVE optimization problems (EOPs) refer to the optimization problems with time-consuming objective functions and/or constraints. EOPs can be classified into three categories: 1) continuous EOPs which contain only continuous variables; 2) combinatorial EOPs which contain only discrete variables;[1] and 3) mixed-variable EOPs [8], [9], which contain both continuous and discrete variables [10], [11].

Furthermore, mixed-variable EOPs can be divided into different kinds according to the types of discrete variables, such as EOPs with continuous and integer variables [12], EOPs with continuous and binary variables [13], and EOPs with continuous and categorical variables (EOPCCVs) [14]. This article mainly focuses on EOPCCVs.

In general, the mathematical model of an EOPCCV can be expressed as

$$\begin{aligned} \min:\ & f\left(\mathbf{x}^{cn}, \mathbf{x}^{ca}\right) \\ \text{s.t.}\ & L_i^{cn} \le x_i^{cn} \le U_i^{cn} \\ & x_j^{ca} \in \mathbf{v}_j \end{aligned} \tag{1}$$

where $\mathbf{x}^{cn} = (x_1^{cn}, x_2^{cn}, \ldots, x_{n_1}^{cn})$ and $\mathbf{x}^{ca} = (x_1^{ca}, x_2^{ca}, \ldots, x_{n_2}^{ca})$ are the continuous and categorical vectors, respectively; n_1 is the number of continuous variables; n_2 is the number of categorical variables; $f(\mathbf{x}^{cn}, \mathbf{x}^{ca})$ is the objective function; L_i^{cn} and U_i^{cn} are the lower and upper bounds of x_i^{cn}, respectively; $\mathbf{v}_j = \{v_j^1, v_j^2, \ldots, v_j^{l_j}\}$ is the candidate categorical set for x_j^{ca}; and l_j is the size of $\mathbf{v}_j$.

Many real-world applications can be modeled as EOPCCVs [16], [17]. The lightweight and crashworthiness design of the side body of an automobile can be taken as an example [15], [18], as shown in Fig. 1. Usually, the side body of an automobile consists of many parts, such as B-Pillar and side door impact beam [19], [20]. Both the structure and material of each part have a great influence on the mass and crashworthiness. When designing the structure of each part, we need to consider its thickness, which is a continuous variable. In addition, we need to select a kind of material from the candidate material set for each part, which is a categorical variable. Moreover, the evaluation of crashworthiness is based on the finite-element analysis (FEA), which is a time-consuming process. Therefore, it is an EOPCCV.

B. Surrogate-Assisted Evolutionary Algorithms

As a kind of powerful optimization tool, evolutionary algorithms (EAs) have been widely applied to solve science and

Manuscript received December 1, 2020; revised March 2, 2021; accepted March 4, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61976225, and in part by the Beijing Advanced Innovation Center for Intelligent Robots and Systems under Grant 2018IRS06. This article was recommended by Associate Editor K.-C. Tan. *(Corresponding author: Yong Wang.)*

Jiao Liu and Yong Wang are with the School of Automation, Central South University, Changsha 410083, China, and also with the Hunan Xiangjiang Artificial Intelligence Academy, Changsha 410083, China (e-mail: liujiao@csu.edu.cn; ywang@csu.edu.cn).

Guangyong Sun and Tong Pang are with the State Key Laboratory of Advanced Design and Manufacture for Vehicle Body, Hunan University, Changsha 410082, China (e-mail: sgy800@126.com; pangtong2011@163.com).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2021.3064676.

Digital Object Identifier 10.1109/TCYB.2021.3064676

[1]Discrete variables can be integer variables [1], categorical variables [2], [3], binary variables [4], [5], and sequential variables [6], [7].

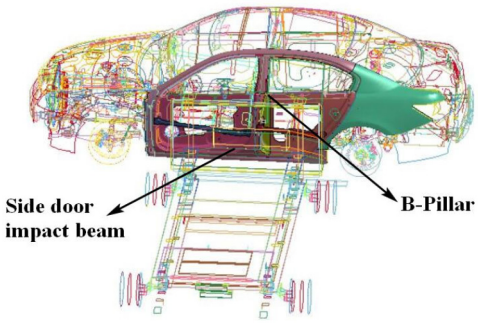


Fig. 1. Lightweight and crashworthiness design of the side body of an automobile [15].

engineering optimization problems [21]–[24]. However, since EAs usually need a large number of function evaluations (FEs) to obtain the optimal solution of an optimization problem, they are not suitable for EOPs. To overcome this barrier, surrogate-assisted EAs (SAEAs), which employ cheap surrogate models to replace a part of time-consuming exact FEs, have been developed [25]–[28]. In the past 15 years, many SAEAs have been proposed to solve EOPs in different fields, such as the mm-wave integrated-circuit optimization [29], structure design of an automobile [30], trauma system design [13], neutral network architecture design [31], antenna design [32], and power system design [33].

Most of current SAEAs focus on continuous EOPs [34]–[37], which utilize surrogate models for continuous functions, such as polynomial regression models [38], support vector regression [14], radial basis functions (RBFs) [39], artificial neural networks [40], and Gaussian processes (GPs) [41]. For example, Liu *et al.* [29] proposed a GP-assisted EA to deal with medium-scale EOPs. Tian *et al.* [42] adopted GP as the surrogate model and proposed a multiobjective infill criterion to deal with high-dimensional EOPs. Wang *et al.* [30] proposed global and local surrogate-assisted differential evolution for expensive constrained optimization problems. Sun *et al.* [43] proposed a surrogate-assisted cooperative swarm optimization algorithm to handle high-dimensional EOPs. Zhang *et al.* [44] combined MOEA/D [45] with GP to deal with expensive multiobjective optimization problems. Chugh *et al.* [46] proposed a surrogate-assisted reference vector guided EA to solve expensive many-objective optimization problems. Since different surrogate models for continuous functions have different strengths for different kinds of problem landscapes, many SAEAs with multiple or ensemble surrogate models for continuous functions have been proposed [47]–[51]. For instance, Lim *et al.* [47] proposed a generalization of surrogate-assisted evolutionary frameworks. Le *et al.* [48] introduced an evolutionary framework with the evolvability learning of surrogates. Lu *et al.* [52] presented an evolutionary optimization framework with hierarchical surrogates. Li *et al.* [53] proposed an ensemble of surrogate-assisted particle swarm optimization algorithms to solve medium-scale EOPs. Guo *et al.* [54] developed a multiobjective EA framework assisted by heterogeneous ensemble surrogate models.

Compared with continuous EOPs, few attempts have been made on combinatorial EOPs [55]. Current studies suggest that surrogate models with tree structures, such as random forest (RF) [56] and least-squares boosting tree (LSBT) [57], [58], are more suitable for dealing with combinatorial EOPs [55]. As a representative, Wang and Jin [59] developed an RF-assisted EA for constrained multiobjective combinatorial optimization in trauma systems. Sun *et al.* [31] incorporated RF into a SAEA to design the architecture of convolutional neural networks. Moreover, some researchers incorporated domain knowledge into SAEAs to solve combinatorial EOPs, thus improving the search ability of the algorithms [60], [61].

Based on our investigation, only several papers work on EOPCCVs [14], [62], [63]. However, the methods proposed in these papers mainly extend surrogate models for continuous functions; thus, their capability of solving EOPCCVs is limited.

C. Motivation and Contributions

For SAEAs, the core problem is how to reasonably use surrogate models to guide the optimization process. As discussed in Section I-B, surrogate models for continuous functions are good at solving continuous EOPs and surrogate models with tree structures perform well on combinatorial EOPs. One may argue that EOPCCVs can be addressed by using surrogate models for continuous functions and surrogate models with tree structures to handle continuous and categorical variables, respectively. However, this way is unreasonable since continuous and categorical variables may interact with each other. Thus, they cannot be optimized separately.

Intuitively, the numbers of continuous and categorical variables have a significant impact on the performance of surrogate models. With respect to an EOPCCV, we consider the following three cases: 1) most of its variables are continuous variables; 2) most of its variables are categorical variables; and 3) the number of continuous variables is similar to that of categorical variables. Obviously, surrogate models for continuous functions and surrogate models with tree structures are good choices for the first and second cases, respectively. However, for the third case, both of these two kinds of surrogate models are necessary. Note that even for the first and second cases, we cannot only use one of these two kinds of surrogate models due to the fact that EOPCCVs contain continuous and categorical variables at the same time. Therefore, in this article, we employ these two kinds of surrogate models simultaneously.

The next issue that arises naturally is how to choose these two kinds of surrogate models for EOPCCVs. In this article, RBF and LSBT are selected as the surrogate model for continuous functions and the surrogate model with a tree structure, respectively. The reasons for our selection are twofold: 1) RBF is a widely used surrogate model for continuous functions. It is simple and easy to train. Moreover, as a kernel-based model, RBF has the potential to deal with categorical variables by redefining the distance between two categorical vectors and 2) as a kind of ensemble surrogate model, LSBT shows excellent generalization ability. In addition, LSBT has the potential to deal with continuous variables by discretizing the decision

space. Therefore, they are expected to complement one another for solving EOPCCVs.

Subsequently, a multisurrogate-assisted ant colony optimization (ACO) algorithm, called MiSACO, is proposed in this article to solve EOPCCVs. To the best of our knowledge, MiSACO is the first attempt to incorporate both the surrogate model for continuous functions and the surrogate model with a tree structure into an EA to solve EOPCCVs. MiSACO introduces two important strategies: 1) multisurrogate-assisted selection and 2) surrogate-assisted local search.

The main contributions of this article can be summarized as follows.

1) The aim of the multisurrogate-assisted selection is to select promising solutions from the offspring solutions generated by ACO_{MV} [10]. In this strategy, we select the best solution from the offspring solutions based on the predicted values provided by RBF (called RBF-based selection) and the best solution from the offspring solutions based on the predicted values provided by LSBT (called LSBT-based selection). In addition, to avoid the population being misled by inaccurate surrogate models, we also randomly select a solution from the offspring solutions (called random selection). As a result, three promising solutions are selected.
2) The surrogate-assisted local search is designed to accelerate the convergence. In this strategy, if the number of evaluated solutions, which have the same categorical variables as the current best solution, is bigger than a threshold, these evaluated solutions are used to construct an RBF for only continuous variables. Based on the constructed RBF, the continuous variables of the current best solution are further optimized by using sequence quadratic programming (SQP), thus improving the quality of the current best solution quickly.
3) Three sets of test problems are used to study the performance of MiSACO. The results suggest that MiSACO has the capability to cope with different types of EOPCCVs. We also apply MiSACO to two practical engineering design problems, that is, the topographical design of stiffened plates against blast loading, and the lightweight and crashworthiness design for the side body of an automobile. The results show that MiSACO can effectively solve them.

The remainder of this article is organized as follows. Section II introduces the related techniques, including the adopted surrogate models and search engine. Section III analyzes the characteristics of RBF and LSBT. The proposed algorithm, MiSACO, is elaborated in Section IV. The experimental studies are executed in Section V. In Section VI, MiSACO is applied to two engineering design problems in the real world. Finally, Section VII concludes this article.

II. Related Techniques

A. RBF

As a commonly used surrogate model, RBF has been widely applied to approximate continuous functions in various science and engineering fields. Based on database $\{(\mathbf{x}_i, y_i)|i = 1, \ldots, N\}$, RBF approximates a continuous function as follows:

$$\hat{f}_{\text{RBF}}(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi(\text{dis}(\mathbf{x}, \mathbf{x}_i)) \tag{2}$$

where $\text{dis}(\mathbf{x}, \mathbf{x}_i) = \|\mathbf{x} - \mathbf{x}_i\|$ represents the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}_i$, and w_i and $\phi(\cdot)$ are the weight coefficient and the basis function, respectively. In general, the Gaussian function [64] is employed as the basis function. When the least-squares loss is taken as the loss function, the weight vector $\mathbf{w} = (w_1, \ldots, w_N)$ can be calculated as follows:

$$\mathbf{w} = \left(\Phi^T \Phi\right)^{-1} \Phi^T \mathbf{y} \tag{3}$$

where $\mathbf{y} = (y_1, \ldots, y_N)$ is the output vector and Φ is the matrix computed as follows:

$$\Phi = \begin{bmatrix} \phi(\text{dis}(\mathbf{x}_1, \mathbf{x}_1)) & \cdots & \phi(\text{dis}(\mathbf{x}_1, \mathbf{x}_N)) \\ \vdots & \ddots & \vdots \\ \phi(\text{dis}(\mathbf{x}_N, \mathbf{x}_1)) & \cdots & \phi(\text{dis}(\mathbf{x}_N, \mathbf{x}_N)) \end{bmatrix}. \tag{4}$$

Note that when approximating functions with both continuous and categorical variables, the distance between two solutions needs to be redefined. Inspired by the Hamming distance, in this article, the distance between the solution to be predicted [i.e., $(\mathbf{x}^{cn}, \mathbf{x}^{ca})$] and the ith solution in the database [i.e., $(\mathbf{x}_i^{cn}, \mathbf{x}_i^{ca})$] is calculated by

$$\text{dis}\left(\left(\mathbf{x}^{cn}, \mathbf{x}^{ca}\right), \left(\mathbf{x}_i^{cn}, \mathbf{x}_i^{ca}\right)\right) = \sqrt{\left\|\mathbf{x}^{cn} - \mathbf{x}_i^{cn}\right\|^2 + \left\|\mathbf{x}^{ca} \oplus \mathbf{x}_i^{ca}\right\|^2} \tag{5}$$

where $\|\cdot\|$ represents the vector norm, $(\mathbf{x}^{cn} - \mathbf{x}_i^{cn})$ represents the difference of two continuous vectors (i.e., $\mathbf{x}^{cn}$ and $\mathbf{x}_i^{cn}$), and $(\mathbf{x}^{ca} \oplus \mathbf{x}_i^{ca})$ represents the vector after the XOR operation of two categorical vectors (i.e., $\mathbf{x}^{ca}$ and $\mathbf{x}_i^{ca}$).

B. LSBT

LSBT is a kind of ensemble learning method based on binary regression trees [57]. The binary regression tree approximates a function by dividing the decision space into several subregions and each of them provides the same predicted value. LSBT can be expressed as the sum of several binary regression trees

$$\hat{f}_{\text{LSBT}}(\mathbf{x}) = \sum_{m=1}^{M} T(\mathbf{x}, \boldsymbol{\Theta}_m) \tag{6}$$

where M is the total number of the binary regression trees, $T(\mathbf{x}, \boldsymbol{\Theta}_m)$ is the mth binary regression tree, and $\boldsymbol{\Theta}_m$ represents the parameter vector, which can be determined iteratively.

Based on database $\{(\mathbf{x}_i, y_i)|i = 1, \ldots, N\}$ and the residual of $\mathbf{x}_i$ in the mth iteration (denoted as $r_{m,i}$), $\boldsymbol{\Theta}_m$ can be obtained by minimizing the following formulation:

$$\sum_{i=1}^{N} \left(r_{m,i} - T(\mathbf{x}_i, \boldsymbol{\Theta}_m)\right)^2 \tag{7}$$

where $r_{m,i} = y_i - \sum_{v=1}^{m-1} T(\mathbf{x}_i, \boldsymbol{\Theta}_v)$.

Note that to cope with continuous variables, when training LSBT, several discrete points are provided for each dimension

Algorithm 1 *ACO_{MV}*

1: Initialize $\mathbb{SA}$;
2: **while** the termination criterion is not satisfied **do**
3: **for** $i = 1 : M$ **do**
4: Construct the ith offspring solution based on the probabilities provided by (9) and (12);
5: Evaluate the ith offspring solution;
6: **end for**
7: Update $\mathbb{SA}$ according to the generated M offspring solutions and elitist selection;
8: **end while**
9: Output the optimal solution

to divide the decision space into several discrete subregions. When predicting the function value of a solution, these discrete points are used to determine which subspace the solution to be predicted is in.

C. ACO_{MV}

The process of ACO_{MV} [10] is described in Algorithm 1. First, we initialize a solution archive (denoted as $\mathbb{SA}$), the purpose of which is to store the continuous and categorical variables of the best K evaluated solutions. The sth ($s \in \{1, \ldots, K\}$) solution in $\mathbb{SA}$ is denoted as: $\mathbf{SA}_s = [\mathbf{SA}_s^{cn}, \mathbf{SA}_s^{ca}] = [(SA_{s,1}^{cn}, SA_{s,2}^{cn}, \ldots, SA_{s,n_1}^{cn}), (SA_{s,1}^{ca}, SA_{s,2}^{ca}, \ldots, SA_{s,n_2}^{ca})]$. A weight (denoted as α_s) is then associated with $\mathbf{SA}_s$, which is calculated as

$$\alpha_s = \frac{1}{qK\sqrt{2\pi}} e^{\frac{-(\text{rank}_s - 1)^2}{2q^2K^2}}, \quad s \in \{1, \ldots, K\} \tag{8}$$

where rank_s represents the rank of $\mathbf{SA}_s$, and q is a parameter called the influence of the best-quality solutions. Note that by utilizing (8), the best solution receives the highest weight, while the weights of the other solutions decrease exponentially with their ranks. Next, ACO_{MV} generates M offspring solutions at each iteration and the elitist selection is used to update $\mathbb{SA}$. The offspring solutions are generated according to α_s. A solution with a big α_s value means a higher probability of sampling around this solution. Since solutions with big α_s values have good objective function values, generating offspring solutions in such a way tends to make the algorithm converge to the promising region. Finally, when the termination criterion is satisfied, the obtained optimal solution is output.

The continuous and categorical variables of an offspring solution are generated in the following ways.

1) When generating the continuous variables of an offspring solution, the continuous vector of the sth solution in $\mathbb{SA}$ is selected based on the following probability:

$$p_s = \frac{\alpha_s}{\sum_{r=1}^{K} \alpha_r}, \quad s \in \{1, \ldots, K\}. \tag{9}$$

We denote the selected continuous vector as $\mathbf{S}^{cn} = (S_1^{cn}, S_2^{cn}, \ldots, S_{n_1}^{cn})$. Then, the ith continuous variable of an offspring solution is generated according to the following Gaussian probability density function:

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{10}$$

where $\mu = S_i^{cn}$. In (10), σ is calculated as

$$\sigma = \xi \sum_{j=1}^{K} \frac{\left| SA_{j,i}^{cn} - S_i^{cn} \right|}{K - 1} \tag{11}$$

where ξ is a parameter called the width of search.

2) For the jth categorical variable of an offspring solution, it is chosen from $\mathbf{v}_j = \{v_j^1, \ldots, v_j^{l_j}\}$ with the following probability:

$$p_{jt} = \frac{\beta_t}{\sum_{h=1}^{l_j} \beta_h}, \quad t \in \{1, \ldots, l_j\} \tag{12}$$

where β_t is the weight associated with the tth value. It is obvious that there are l_j values for the jth categorical variable. Suppose that η_j is the number of values that do not appear in $\mathbb{SA}$, and u_{jt} is the repeated number of the tth value that appears in $\mathbb{SA}$. If $u_{jt} > 1$, suppose that the indexes of the weights corresponding to the tth value in $\mathbb{SA}$ are: $id_1, \ldots, id_{u_{jt}}$. Let $\alpha_{jt} = \max\{\alpha_{id_1}, \ldots, \alpha_{id_{u_{jt}}}\}$. Then, β_t is calculated as

$$\beta_t = \begin{cases} \frac{\alpha_{jt}}{u_{jt}} + \frac{q}{\eta_j}, & \text{if } (\eta_j > 0,\ u_{jt} > 0) \\ \frac{q}{\eta_j}, & \text{if } (\eta_j > 0,\ u_{jt} = 0) \\ \frac{\alpha_{jt}}{u_{jt}}, & \text{if } (\eta_j = 0,\ u_{jt} > 0). \end{cases} \tag{13}$$

III. Characteristics of RBF and LSBT

In this section, the characteristics of RBF and LSBT on approximating EOPCCVs are analyzed. First, three assumptions are given. Afterward, we analyze the predicted error of RBF and LSBT based on these three assumptions. Finally, some considerations behind the analysis are provided.

A. Assumptions

First, we give an assumption to limit the change range of the objective function value according to the distance between two solutions. This assumption is inspired by the bi-Lipschitz continuity, that is, a kind of smoothness condition that has been widely used in the theoretical analysis of the Bayesian optimization [65]. The assumption is provided as follows.

Assumption 1: Based on the distance defined in Section II-A, $f(\mathbf{x}^{cn}, \mathbf{x}^{ca})$ and $\hat{f}_{\text{RBF}}(\mathbf{x}^{cn}, \mathbf{x}^{ca})$ satisfy the following conditions:

$$
\begin{aligned}
&\left|f\left(\mathbf{x}_1^{cn},\mathbf{x}_1^{ca}\right)-f\left(\mathbf{x}_2^{cn},\mathbf{x}_2^{ca}\right)\right| \\
&\quad\geq \frac{1}{L_f}\cdot \mathrm{dis}\left(\left(\mathbf{x}_1^{cn},\mathbf{x}_1^{ca}\right),\left(\mathbf{x}_2^{cn},\mathbf{x}_2^{ca}\right)\right) \\
&\left|f\left(\mathbf{x}_1^{cn},\mathbf{x}_1^{ca}\right)-f\left(\mathbf{x}_2^{cn},\mathbf{x}_2^{ca}\right)\right| \\
&\quad\leq L_f\cdot \mathrm{dis}\left(\left(\mathbf{x}_1^{cn},\mathbf{x}_1^{ca}\right),\left(\mathbf{x}_2^{cn},\mathbf{x}_2^{ca}\right)\right) \\
&\left|\hat{f}_{\mathrm{RBF}}\left(\mathbf{x}_1^{cn},\mathbf{x}_1^{ca}\right)-\hat{f}_{\mathrm{RBF}}\left(\mathbf{x}_2^{cn},\mathbf{x}_2^{ca}\right)\right| \\
&\quad\geq \frac{1}{L_r}\cdot \mathrm{dis}\left(\left(\mathbf{x}_1^{cn},\mathbf{x}_1^{ca}\right),\left(\mathbf{x}_2^{cn},\mathbf{x}_2^{ca}\right)\right) \\
&\left|\hat{f}_{\mathrm{RBF}}\left(\mathbf{x}_1^{cn},\mathbf{x}_1{}^{ca}\right)-\hat{f}_{\mathrm{RBF}}\left(\mathbf{x}_2^{cn},\mathbf{x}_2^{ca}\right)\right| \\
&\quad\leq L_r\cdot \mathrm{dis}\left(\left(\mathbf{x}_1^{cn},\mathbf{x}_1^{ca}\right),\left(\mathbf{x}_2^{cn},\mathbf{x}_2^{ca}\right)\right) \qquad (14)
\end{aligned}
$$

where $(\mathbf{x}_1^{cn},\mathbf{x}_1^{ca})$ and $(\mathbf{x}_2^{cn},\mathbf{x}_2^{ca})$ are two different solutions, and L_f and L_r are two parameters [65].

Second, we consider that for at least one solution, RBF and LSBT can accurately predict its objective function value. Thus, the following assumption is provided.

Assumption 2: For both RBF and LSBT, there exists at least one reference solution $(\mathbf{x}_*^{cn},\mathbf{x}_*^{ca})$ that can make $f(\mathbf{x}_*^{cn},\mathbf{x}_*^{ca})=\hat{f}_{\mathrm{RBF}}(\mathbf{x}_*^{cn},\mathbf{x}_*^{ca})=\hat{f}_{\mathrm{LSBT}}(\mathbf{x}_*^{cn},\mathbf{x}_*^{ca})$.

Finally, to make it easier to analyze the predicted error of LSBT, the following assumption is given.

Assumption 3: When predicting the objective function value, we assume that the solution to be predicted [denoted as $(\mathbf{x}^{cn},\mathbf{x}^{ca})$] is close to $(\mathbf{x}_*^{cn},\mathbf{x}_*^{ca})$; thus, $(\mathbf{x}^{cn},\mathbf{x}^{ca})$ and $(\mathbf{x}_*^{cn},\mathbf{x}_*^{ca})$ are located in the same subregion provided by LSBT.

B. Analysis

For RBF and LSBT, we have the following two propositions.

Proposition 1: The upper bound of RBF's predicted error is described as

$$
\begin{aligned}
&\left|f\left(\mathbf{x}^{cn},\mathbf{x}^{ca}\right)-\hat{f}_{\mathrm{RBF}}\left(\mathbf{x}^{cn},\mathbf{x}^{ca}\right)\right| \\
&\quad\leq \left(L_f+L_r\right)\cdot \mathrm{dis}\left(\left(\mathbf{x}^{cn},\mathbf{x}^{ca}\right),\left(\mathbf{x}_*^{cn},\mathbf{x}_*^{ca}\right)\right). \qquad (15)
\end{aligned}
$$

Proposition 2: The upper and lower bounds of LSBT's predicted error are described as

$$
\left|f\left(\mathbf{x}^{cn},\mathbf{x}^{ca}\right)-\hat{f}_{\mathrm{LSBT}}\left(\mathbf{x}^{cn},\mathbf{x}^{ca}\right)\right|\leq L_f\cdot \mathrm{dis}\left(\left(\mathbf{x}^{cn},\mathbf{x}^{ca}\right),\left(\mathbf{x}_*^{cn},\mathbf{x}_*^{ca}\right)\right) \qquad (16)
$$

and

$$
\left|f\left(\mathbf{x}^{cn},\mathbf{x}^{ca}\right)-\hat{f}_{\mathrm{LSBT}}\left(\mathbf{x}^{cn},\mathbf{x}^{ca}\right)\right|\geq \frac{1}{L_f}\cdot \mathrm{dis}\left(\left(\mathbf{x}^{cn},\mathbf{x}^{ca}\right),\left(\mathbf{x}_*^{cn},\mathbf{x}_*^{ca}\right)\right). \qquad (17)
$$

The proof of Propositions 1 and 2 is given in Section S-I of the supplementary file. Next, based on these two propositions, we discuss the upper and lower bounds of the predicted error.

1) Discussions About the Upper Bound: From (15) and (16), it can be observed that when $\mathrm{dis}((\mathbf{x}^{cn},\mathbf{x}^{ca}),(\mathbf{x}_*^{cn},\mathbf{x}_*^{ca}))\neq 0$, the upper bound of LSBT's predicted error is always smaller than that of RBF's.

2) Discussions About the Lower Bound: It should be noted that the best predicted error provided by RBF can be equal to zero, which means that RBF has the chance to provide an accurate predicted value. In contrast, according to (17), the lower bound of LSBT's predicted error cannot be zero when $\mathrm{dis}((\mathbf{x}^{cn},\mathbf{x}^{ca}),(\mathbf{x}_*^{cn},\mathbf{x}_*^{ca}))\neq 0$, which means that it is hard for LSBT to accurately predict the objective function values of all solutions except the reference solution.

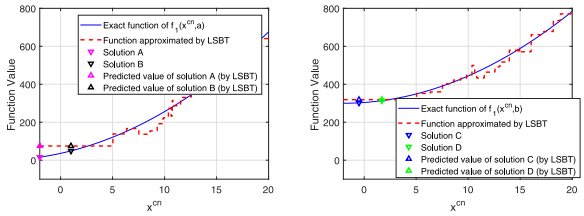


Fig. 2. Approximate $f_1(x^{cn},x^{ca})$ by LSBT. (a) Landscapes of the exact function of $f_1(x^{cn},x^{ca})$ and the function approximated by LSBT when $x^{ca}=a$. (b) Landscapes of the exact function of $f_1(x^{cn},x^{ca})$ and the function approximated by LSBT when $x^{ca}=b$.

In summary, RBF and LSBT have different advantages and disadvantages. When copping with EOPCCVs, RBF has the opportunity to accurately predict the objective function value of each solution. However, the upper bound of its predicted error is larger than that of LSBT. Thus, compared with LSBT, the predicted error of RBF may fluctuate greatly. In contrast, under the condition that $\mathrm{dis}((\mathbf{x}^{cn},\mathbf{x}^{ca}),(\mathbf{x}_*^{cn},\mathbf{x}_*^{ca}))\neq 0$, it is hard for LSBT to accurately predict the objective function, but LSBT has a stable prediction capability.

C. Considerations Based on the Above Analysis

According to the above analysis, we have the following considerations about how to effectively use surrogate models when solving EOPCCVs.

1) Due to the fact that LSBT has a stable prediction capability, when using it to select a solution from the offspring solutions, the one with good quality is very likely to be chosen. However, since it is hard for LSBT to accurately approximate the objective function of an EOPCCV, the most promising solution may be missed. Therefore, only using LSBT to guide the algorithm may not be effective. An example in Fig. 2 is used to illustrate this issue. In Fig. 2, LSBT is used to approximate $f_1(x^{cn},x^{ca})$ with $n_1=1$, $n_2=1$, $x^{cn}\in[-2,20]$, and $x^{ca}\in\{a,b\}$. The aim is to select the best solution from **A**, **B**, **C**, and **D**. Note that **A** has the best original objective function value, and **A** and **B** are better than **C** and **D**. According to the predicted values, the solution with good quality (i.e., **A** or **B**) will be selected. However, since **A** and **B** have the same predicted value, the most promising solution (i.e., **A**) may not be selected.
2) Although RBF has the opportunity to accurately approximate the objective function of each solution, its prediction error may fluctuate greatly. Therefore, only using RBF to guide the search may mislead the algorithm to converge to a wrong optimal solution. An example in Fig. 3 is used to illustrate this issue. In Fig. 3, RBF is used to approximate the following two functions: a) $f_1(x^{cn},x^{ca})$ with $n_1=1$, $n_2=1$, $x^{cn}\in[-2,20]$, and $x^{ca}\in\{a,b\}$ and b) $f_2(x^{cn},\mathbf{x}^{ca})$

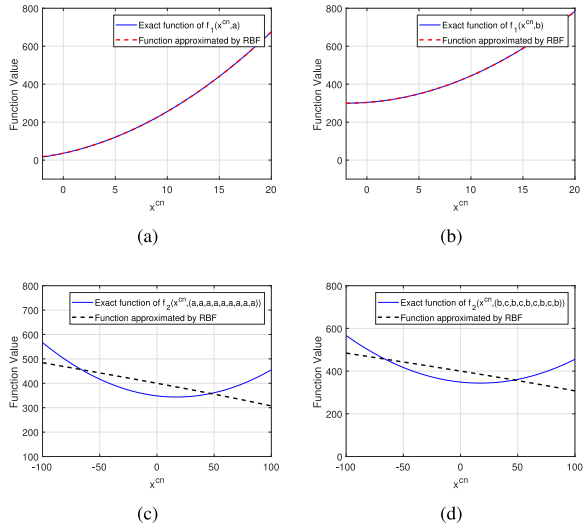


Fig. 3. Approximate $f_1(x^{cn}, x^{ca})$ and $f_2(x^{cn}, \mathbf{x}^{ca})$ by RBF. (a) Landscapes of the exact function of $f_1(x^{cn}, x^{ca})$ and the function approximated by RBF when $x^{ca} = a$. (b) Landscapes of the exact function of $f_1(x^{cn}, x^{ca})$ and the function approximated by RBF $x^{ca} = b$. (c) Landscapes of the exact function of $f_2(x^{cn}, \mathbf{x}^{ca})$ and the function approximated by RBF when $\mathbf{x}^{ca} = (a, a, a, a, a, a, a, a)$. (d) Landscapes of the exact function of $f_2(x^{cn}, \mathbf{x}^{ca})$ and the function approximated by RBF when $\mathbf{x}^{ca} = (b, c, b, c, b, c, b, c)$.

with $n_1 = 1$, $n_2 = 9$, $x^{cn} \in [-100, 100]$, and $x_j^{ca} \in \{a, b, c, d, e\}(j = \{1, \ldots, n_2\})$. The exact functions of $f_1(x^{cn}, x^{ca})$ and $f_2(x^{cn}, \mathbf{x}^{ca})$, and the functions approximated by RBF are exhibited in Fig. 3. For convenience, when approximating $f_2(x^{cn}, \mathbf{x}^{ca})$, we only exhibit the functions when $\mathbf{x}^{ca} = (a, a, a, a, a, a, a, a, a)$ and $\mathbf{x}^{ca} = (b, c, b, c, b, c, b, c, b)$. It can be observed that RBF can exactly approximate $f_1(x^{cn}, x^{ca})$. However, with respect to $f_2(x^{cn}, \mathbf{x}^{ca})$, the prediction error of RBF is large, thus may misleading the optimization process.

Based on the above considerations, we employ both LSBT and RBF to assist EAs to handle EOPCCVs. By doing this, on one hand, LSBT ensures that a solution with good quality can be selected from the offspring solutions generated by EAs; on the other hand, the use of RBF makes EAs have a chance to select the most promising solution, thus improving the efficiency of evolution.

IV. Proposed Method

A. General Framework

The framework of MiSACO is given in Algorithm 2. The symbols in Algorithm 2 are explained as follows.

1) $\mathbb{DB}$: The database containing the information of all the evaluated solutions, that is, the continuous variables, categorical variables, and objective function values of all the evaluated solutions.
2) $\mathbb{SA}$: The solution archive used in ACO_{MV}.
3) *FEs:* The number of FEs.
4) $\mathbb{OP}$: The set containing the offspring solutions generated by ACO_{MV}.
5) $\mathbb{X}_{sel}$: The set containing the solutions selected by the multisurrogate-assisted selection.

Algorithm 2 *MiSACO*

1: $[\mathbb{SA}, \mathbb{DB}, FEs] \leftarrow Initialization$;
2: **while** $FEs < MaxFEs$ **do**
3: $\quad \mathbb{OP} \leftarrow ACO_{MV}(\mathbb{SA})$;
4: $\quad \mathbb{X}_{sel} \leftarrow MSA\_Selection(\mathbb{OP})$;
5: $\quad \mathbb{X}_{ls} \leftarrow SA\_LocalSearch(\mathbb{DB})$;
6: $\quad \mathbb{X} = \mathbb{X}_{sel} \cup \mathbb{X}_{ls}$;
7: $\quad [\mathbb{Y}, FEs] \leftarrow Evaluation(\mathbb{X}, FEs)$;
8: $\quad \mathbb{DB} = \mathbb{DB} \cup \{\mathbb{X}, \mathbb{Y}\}$;
9: $\quad \mathbb{SA} \leftarrow Update(\mathbb{SA}, \mathbb{X}, \mathbb{Y})$;
10: **end while**
11: Output the best solution $\mathbf{x}_{best}$.

6) $\mathbb{X}_{ls}$: The set containing the solution founded by the surrogate-assisted local search.
7) $\mathbb{X}$: The set containing the solutions in $\mathbb{X}_{sel}$ and $\mathbb{X}_{ls}$.
8) $\mathbb{Y}$: The set containing the original expensive objective function values of the solutions in $\mathbb{X}$.

The process of Algorithm 2 can be divided into the following five steps.

1) *Initialization (Line 1):* This step produces K initial solutions via Latin hypercube design and puts them into $\mathbb{SA}$. Subsequently, it evaluates them by the original expensive objective function and initializes $\mathbb{DB}$ and *FEs*.
2) *Generating Offspring Solutions (Line 3):* In this step, M offspring solutions are generated based on ACO_{MV} and reserved into $\mathbb{OP}$.
3) *Multisurrogate-Assisted Selection (Line 4):* This step selects three solutions from $\mathbb{OP}$ according to the multisurrogate-assisted selection, and puts them into $\mathbb{X}_{sel}$.
4) *Surrogate-Assisted Local Search (Line 5):* In this step, the surrogate-assisted local search is used to enhance the quality of the best solution in $\mathbb{DB}$ by further optimizing its continuous variables, and the obtained solution is reserved into $\mathbb{X}_{ls}$.
5) *Updating $\mathbb{DB}$ and $\mathbb{SA}$ (Lines 6–9):* In this step, the solutions in $\mathbb{X}_{sel}$ and $\mathbb{X}_{ls}$ are evaluated by the original expensive objective function. The information of them is kept in $\mathbb{DB}$. Then, based on these solutions, $\mathbb{SA}$ is updated according to the elitist selection.

The unique characteristic of MiSACO lies in its multisurrogate-assisted selection and surrogate-assisted local search. Next, we explain these two strategies, respectively.

B. Multisurrogate-Assisted Selection

The process of the multisurrogate-assisted selection is described in Algorithm 3, in which three selection operators (i.e., the RBF-based selection, the LSBT-based selection, and the random selection) are adopted to select three promising solutions from $\mathbb{OP}$. First, we construct an RBF surrogate model (denoted as $\hat{f}_{\text{RBF}}$) and an LSBT surrogate model (denoted as $\hat{f}_{\text{LSBT}}$) based on $\mathbb{DB}$, and evaluate all the solutions in $\mathbb{OP}$ by using $\hat{f}_{\text{RBF}}$ and $\hat{f}_{\text{LSBT}}$, respectively. Then, the solution with the best $\hat{f}_{\text{RBF}}$ value is selected from $\mathbb{OP}$. We record this solution as $\mathbf{x}_1$, and remove it from $\mathbb{OP}$. Subsequently, the

Algorithm 3 *MSA_Selection*($\mathbb{OP}$)

1: Construct an RBF surrogate model (denoted as $\hat{f}_{RBF}$) and an LSBT surrogate model (denoted as $\hat{f}_{LSBT}$) by utilizing all the solutions in $\mathbb{DB}$ to approximate the objective function of an EOPCCV;
2: Evaluate the solutions in $\mathbb{OP}$ with $\hat{f}_{RBF}$ and select the best one, denoted as $\mathbf{x}_1$;
3: $\mathbb{OP} = \mathbb{OP} \setminus \mathbf{x}_1$;
4: Evaluate the solutions in $\mathbb{OP}$ with $\hat{f}_{LSBT}$ and select the best one, denoted as $\mathbf{x}_2$;
5: $\mathbb{OP} = \mathbb{OP} \setminus \mathbf{x}_2$;
6: Select a solution from $\mathbb{OP}$ randomly, denoted as $\mathbf{x}_3$;
7: $\mathbb{X}_{sel} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$;
8: Output $\mathbb{X}_{sel}$.

solution with the best $\hat{f}_{\text{LSBT}}$ value, denoted as $\mathbf{x}_2$, is selected and removed from $\mathbb{OP}$. Next, a solution, denoted as $\mathbf{x}_3$, is randomly selected from $\mathbb{OP}$. Finally, these three solutions (i.e., $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$) are reserved into $\mathbb{X}_{sel}$.

As mentioned in Section III, when RBF or LSBT is used to approximate the objective function of an EOPCCV, the accuracy cannot be guaranteed. If the accuracy is poor, some solutions with good original objective function values may have bad predicted values. Under this condition, they may be missed. To alleviate this issue, we randomly select a solution from $\mathbb{OP}$ without depending on any surrogate model. An example in Fig. 4 is used to illustrate this issue: $f_3(x^{cn}, x^{ca})$ with $n_1 = 1$, $n_2 = 1$, $x^{cn} \in [-10, 10]$, and $x^{ca} \in \{a, b\}$. It is approximated by RBF and LSBT, respectively. For convenience, we only exhibit the landscapes of the exact function of $f_3(x^{cn}, x^{ca})$ and the function approximated by RBF/LSBT when $x^{ca} = a$. Our aim is to select a better solution from solutions **A** and **B**. Note that the original objective function value of **B** is better than that of **A**. However, both RBF and LSBT provide a better predicted value for **A**, which means **A** will be selected. In contrast, if the random selection is employed, we still have a chance to select **B**, thus improving the accuracy of the surrogate models as shown in Fig. 4(b) and (d) .

C. Surrogate-Assisted Local Search

Considering that the local search strategy can effectively improve the convergence speed [30], [49], in this article, we also design a surrogate-assisted local search strategy to accelerate the convergence. We incorporate RBF into SQP to further optimize the continuous variables of the current best solution in $\mathbb{DB}$ (denoted as $\mathbf{x}_{\text{best}} = [\mathbf{x}^{cn}_{\text{best}}, \mathbf{x}^{ca}_{\text{best}}]$). The implementation of the surrogate-assisted local search is explained as follows.

First, we count the number of the solutions that have the same categorical variables with $\mathbf{x}_{\text{best}}$ (denoted as N_{ls}). If N_{ls} is bigger than a threshold (denoted as $N_{\min}$), these solutions are used to construct an RBF surrogate model (denoted as $\hat{f}_{\text{sub}}$) for only continuous variables. Then, SQP is used to solve the following optimization problem:

$$\begin{aligned} \min:\ & \hat{f}_{\text{sub}}\left(\mathbf{x}^{cn}\right) \\ \text{s.t.}\ & L_i^{cn} \le x_i^{cn} \le U_i^{cn}. \end{aligned} \tag{18}$$

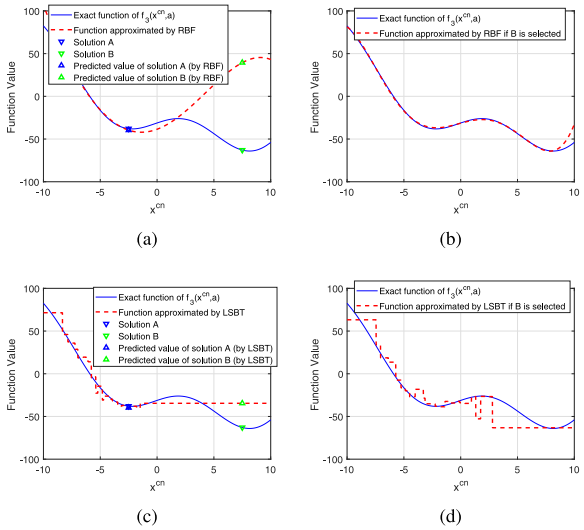


Fig. 4. Approximate $f_3(x^{cn}, x^{ca})$ by RBF and LSBT. (a) Landscapes of the exact function of $f_3(x^{cn}, x^{ca})$ and the function approximated by RBF when $x^{ca} = a$. (b) Landscapes of the exact function of $f_3(x^{cn}, x^{ca})$ and the function approximated by RBF when $x^{ca} = a$ if **B** is selected. (c) Landscapes of the exact function of $f_3(x^{cn}, x^{ca})$ and the function approximated by LSBT when $x^{ca} = a$. (d) Landscapes of the exact function of $f_3(x^{cn}, x^{ca})$ and the function approximated by LSBT when $x^{ca} = a$ if **B** is selected.

Based on the continuous vector obtained by SQP (denoted as $\mathbf{x}^{cn}_{ls}$), a new solution is produced: $\mathbf{x}_{ls} = [\mathbf{x}^{cn}_{ls}, \mathbf{x}^{ca}_{\text{best}}]$. Finally, $\mathbf{x}_{ls}$ is reserved into $\mathbb{X}_{ls}$.

Next, we would like to give two comments on the surrogate-assisted local search.

1) Commonly, it is hard to guarantee the accuracy of RBF if N_{ls} is too small. Therefore, a threshold is adopted to ensure the size of the data points.
2) Since it is almost impossible that many solutions have the same continuous vector, it is very hard to construct an LSBT surrogate model for only categorical variables. Therefore, we do not improve the categorical variables.

V. Experimental Studies

A. Test Problems and Parameters Settings

1) Artificial Test Problems: The first set of test problems contains 30 artificial test problems (i.e., F1–F30). They are originated from five classical continuous functions: 1) Sphere function; 2) Rastrigin function; 3) Alckey function; 4) Elliposoid function; and 5) Griewank function. Their characteristics are listed in Table S-XV of the supplementary files. According to their characteristics, we roughly classify them into three types.

a) *Type 1:* Most of the variables are continuous variables.
b) *Type 2:* Most of the variables are categorical variables.
c) *Type 3:* The number of continuous variables is similar to that of categorical variables.

Obviously, F1–F10 are type-1 artificial test problems, F11–F20 are type-2 artificial test problems, and F21–F30 are type-3 artificial test problems.

2) Capacitated Facility Location Problems: Six capacitated facility location problems (i.e., CFLP1–CFLP6) are constructed in this article. The capacitated facility location

problems can be formulated as follows:

$$\begin{aligned}\min:\ & f(\mathbf{x},\mathbf{y}) = \sum_{i\in I} F_{i,y_i} y_i + \sum_{i\in I}\sum_{j\in J} Q_{i,j} x_{i,j} \\ & \sum_{i\in I} x_{i,j} = D_j, i \in I, j \in J \\ & \sum_{j\in J} x_{i,j} \le C_{y_i} y_i, i \in I, j \in J \\ & y_i \in S, i \in I \\ & x_{i,j} \ge 0, i \in I, j \in J\end{aligned}$$

where $I = \{1, \ldots, m\}$ represents a set of potential facility sites, $J = \{1, \ldots, n\}$ represents a set of customers, $S = \{0, \ldots, s\}$ represents a set of facility types, $C_r (r \in S)$ represents the capacity of the rth type of facility, $F_{i,r}$ represents the cost of operating the rth type of facility at site i, D_j represents the total demand of the jth customer, $Q_{i,j}$ represents the cost of serving a unit of demand for the jth customer from the ith facility, $x_{i,j}$ denotes the jth customer's demand from the ith facility, and y_i represents which type of facility is operated at site i (if $y_i = 0$, no facility will be operated at site i).

3) Dubins Traveling Salesperson Problems: We also construct six Dubins traveling salesperson problems (i.e., DTSP1–DTSP6) in this article. The Dubins travelling salesperson problems are related to the motion planning and task assignment for uninhabited vehicles. They can be formulated as follows:

$$\begin{aligned}\min\ & D(\mathbf{r},\mathbf{x}) = \sum_{i=1}^{n-1} d\left(x_{r_i}, x_{r_{i+1}}\right) + d\left(x_{r_n}, x_{r_1}\right) \\ \text{s.t.}\ & r_i \ne r_j, \quad \text{if } i \ne j \\ & r_i \in \{1, \ldots, n\} \\ & 0 \le x_i \le 2\pi \\ & i \in \{1, \ldots, n\} \\ & j \in \{1, \ldots, n\}\end{aligned}$$

where $\mathbf{r} = (r_1, \ldots, r_n)$ represents the sequence of waypoints needed to pass through, $r_i \in \{1, \ldots, n\}$ represents the ith waypoint, n represents the number of waypoints, $\mathbf{x} = \{x_1, \ldots, x_n\}$ represents the heading of the uninhabited vehicle at the ith waypoint, and $d(\cdot,\cdot)$ represents the shortest Dubins path between two waypoints. For $d(\cdot,\cdot)$, the shortest Dubins path between two waypoints must be one of the following six patterns: {RSL, LSR, RSR, LSL, RLR, LRL}, in which L, R, and S represent turning left with the minimal turning radius, turning right with the minimal turning radius, and moving along a straight line, respectively.

The details of these three sets of test problems are given in Section S-IV of the supplementary file. For each test problem, 20 independent runs were implemented. The parameter settings of MiSACO are listed in Table I. The settings of q and M were consistent with the original paper [10]. For each test problem, 20 independent runs were implemented.

To evaluate the performance of different algorithms, the following two indicators were calculated.

TABLE I
PARAMETER SETTINGS OF MISACO

Parameter	Value
Size of $\mathbb{OP}$: M	100
Influence of the best-quality solutions in ACO_{MV}: q	0.05099
Width of the search in ACO_{MV}: ξ	0.6795
Archive size in ACO_{MV}: K	60
Maximum number of function evolutions: $MaxFEs$	600
Threshold in the surrogate-assisted local search: N_{min}	$5 * n_1$

1) *AOFV:* The average objective function value of the best solutions provided by an algorithm over 20 independent runs.
2) *ASFEs:* The average FEs consumed by an algorithm to successfully obtain the optimal solution of a test problem over 20 independent runs. Note that a run is considered as successful if the following condition is satisfied: $|f(\mathbf{x}_{\text{best}}) - f(\mathbf{x}^*)| \le 1$, where $\mathbf{x}^*$ is the best known solution and $\mathbf{x}_{\text{best}}$ is the best solution provided by an algorithm. For an unsuccessful run, its consumed FEs was set to *MaxFEs*.

AOFV and *ASFEs* measure the convergence accuracy and efficiency of an algorithm, respectively. Since the optimal solutions of CFLP1–CFLP6 and DTSP1–DTSP6 are unknown, for these 12 problems, we did not calculate their *ASFEs* values. In the experimental studies, the Wilcoxon's rank-sum test at a 0.05 significance level was implemented between MiSACO and each of its competitors to test the statistical significance. In the following tables, "+," "−," and "≈" denote that MiSACO performs better than, worse than, and similar to its competitor, respectively.

B. Comparison With ACO_{MV}

In essence, MiSACO is an algorithm that combines surrogate models with ACO_{MV} for solving EOPCCVs. One may be interested in the performance difference between MiSACO and ACO_{MV}. To this end, we compared MiSACO with ACO_{MV}. To clearly exhibit their performance difference, we also calculated a performance metric called acceleration rate based on their *ASFEs* values

$$\text{AR} = \frac{\text{ASFEs}_{\text{ACO}_{\text{MV}}} - \text{ASFEs}_{\text{MiSACO}}}{\text{ASFEs}_{\text{ACO}_{\text{MV}}}} \times 100\% \qquad (19)$$

where $\text{ASFEs}_{\text{ACO}_{\text{MV}}}$ and $\text{ASFEs}_{\text{MiSACO}}$ are the *ASFEs* values of ACO_{MV} and MiSACO, respectively. Note that if any of these two algorithms fails to find any optimal solution over 20 independent runs, the *ASFEs* value will be equal to *MaxFEs*. Under this condition, it is meaningless to calculate the acceleration rate. When this happens, the corresponding *AR* value is denoted as "*NA*." All the results are exhibited in Tables II and III, and Tables S-I and S-II of the supplementary file.

The detailed discussions about the results are given as follows.

1) Results on the Artificial Test Problems:

a) In terms of *AOFV*, it can be observed from Table II that MiSACO can obtain better values than ACO_{MV} on all the 30 artificial test problems. Since F2, F7, F12, F17, F22,

TABLE II
RESULTS OF ACO_{MV} AND MISACO OVER 20 INDEPENDENT RUNS ON THE 30 ARTIFICIAL TEST PROBLEMS. THE WILCOXON'S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL WAS PERFORMED BETWEEN ACO_{MV} AND MISACO

	Problem	ACO_{MV} *AOFV* ± *Std Dev*		MiSACO *AOFV* ± *Std Dev*	ACO_{MV} *ASFEs* ± *Std Dev*		MiSACO *ASFEs* ± *Std Dev*
Type 1	F1	4.12E+00 ± 2.61E+00	+	**6.21E−08 ± 2.24E−08**	600.00 ± 0.00	+	**249.95 ± 36.49**
	F2	6.04E+01 ± 1.22E+01	+	**2.59E+01 ± 1.19E+01**	**600.00 ± 0.00**	≈	**600.00 ± 0.00**
	F3	2.04E+00 ± 6.16E−01	+	**3.04E−01 ± 7.50E−01**	600.00 ± 0.00	+	**375.85 ± 101.67**
	F4	2.63E−01 ± 2.13E−01	+	**4.51E−09 ± 2.25E−09**	457.90 ± 71.55	+	**159.10 ± 25.60**
	F5	1.01E+00 ± 1.26E−01	+	**2.39E−01 ± 2.30E−01**	587.95 ± 30.73	+	**208.90 ± 37.03**
	F6	6.43E+00 ± 6.28E+00	+	**5.74E−08 ± 2.50E−08**	593.60 ± 19.08	+	**269.40 ± 56.18**
	F7	5.53E+01 ± 7.09E+00	+	**2.40E+01 ± 1.22E+01**	**600.00 ± 0.00**	≈	**600.00 ± 0.00**
	F8	1.79E+00 ± 7.11E−01	+	**7.27E−02 ± 2.83E−01**	597.55 ± 8.89	+	**391.60 ± 63.22**
	F9	3.38E−01 ± 2.98E−01	+	**2.59E−03 ± 8.51E−03**	469.25 ± 77.56	+	**231.15 ± 65.29**
	F10	1.08E+00 ± 1.10E−01	+	**2.75E−01 ± 3.01E−01**	595.60 ± 16.49	+	**270.20 ± 123.87**
Type 2	F11	2.39E+00 ± 4.59E+00	+	**9.43E−08 ± 1.06E−07**	533.55 ± 81.34	+	**285.75 ± 73.21**
	F12	5.19E+01 ± 2.00E+01	+	**4.71E+01 ± 1.73E+01**	**600.00 ± 0.00**	≈	**600.00 ± 0.00**
	F13	1.15E+00 ± 9.63E−01	+	**4.19E−01 ± 1.29E+00**	546.75 ± 96.96	+	**307.25 ± 120.94**
	F14	9.10E−03 ± 8.17E−03	+	**2.04E−09 ± 2.03E−09**	354.25 ± 80.54	+	**195.45 ± 39.24**
	F15	7.80E−01 ± 3.02E−01	+	**2.82E−07 ± 3.27E−07**	503.85 ± 86.41	+	**244.50 ± 44.50**
	F16	2.82E+01 ± 4.67E+01	+	**1.27E+00 ± 5.66E+00**	564.75 ± 62.57	+	**409.65 ± 100.36**
	F17	6.36E+01 ± 1.30E+01	+	**4.50E+01 ± 1.61E+01**	**600.00 ± 0.00**	≈	**600.00 ± 0.00**
	F18	1.71E+00 ± 1.43E+00	+	**1.55E+00 ± 1.96E+00**	553.00 ± 70.48	+	**538.25 ± 93.69**
	F19	8.71E−01 ± 6.51E−01	+	**2.71E−01 ± 7.16E−01**	521.60 ± 85.39	+	**356.45 ± 122.27**
	F20	1.65E+00 ± 1.08E+00	+	**1.05E−01 ± 3.23E−01**	564.10 ± 64.50	+	**401.70 ± 109.14**
Type 3	F21	7.08E+00 ± 4.92E+00	+	**7.60E−08 ± 5.64E−08**	599.75 ± 1.12	+	**279.55 ± 36.60**
	F22	6.20E+01 ± 1.33E+01	+	**4.78E+01 ± 1.32E+01**	**600.00 ± 0.00**	≈	**600.00 ± 0.00**
	F23	2.43E+00 ± 8.52E−01	+	**1.52E−01 ± 6.77E−01**	597.35 ± 11.85	+	**363.15 ± 76.32**
	F24	3.09E−01 ± 3.10E−01	+	**9.90E−07 ± 4.42E−06**	444.45 ± 88.68	+	**185.15 ± 37.79**
	F25	1.07E+00 ± 6.24E−02	+	**1.24E−01 ± 2.34E−01**	598.50 ± 6.71	+	**229.45 ± 32.75**
	F26	2.76E+01 ± 7.30E+01	+	**9.70E−08 ± 8.28E−08**	600.00 ± 0.00	+	**330.80 ± 62.07**
	F27	6.46E+01 ± 8.30E+00	+	**4.15E+01 ± 1.78E+01**	**600.00 ± 0.00**	≈	**600.00 ± 0.00**
	F28	1.77E+00 ± 1.07E+00	+	**9.76E−01 ± 1.46E+00**	589.85 ± 18.96	+	**475.15 ± 117.83**
	F29	3.74E−01 ± 3.27E−01	+	**3.74E−02 ± 1.44E−01**	459.60 ± 84.79	+	**273.10 ± 83.76**
	F30	1.16E+00 ± 2.41E−01	+	**4.92E−01 ± 6.18E−01**	600.00 ± 0.00	+	**353.50 ± 118.11**
$+/-/\approx$		30/0/0			24/0/6		

and F27 originate from the Rastrigin function, which is a function with a complex and multimodal landscape, it is very likely to build an inaccurate surrogate model. Therefore, MiSACO cannot find solutions with high accuracies when solving these six artificial test problems. However, MiSACO can still provide smaller *AOFV* values than ACO_{MV} on them. From the Wilcoxon's rank-sum test, MiSACO surpasses ACO_{MV} on all the 30 artificial test problems in terms of *AOFV*.

b) As far as *ASFEs* is concerned, we can observe from Table II that the values provided by MiSACO are better than those resulting from ACO_{MV} on all the 30 artificial test problems except F2, F7, F12, F17, F22, and F27. When solving these six artificial test problems, both MiSACO and ACO_{MV} cannot successfully obtain the optimal solutions in any run. Therefore, their *ASFEs* values are equal to *MaxFEs*. According to the Wilcoxon's rank-sum test, MiSACO beats ACO_{MV} on 24 artificial test problems in terms of *ASFEs*. However, ACO_{MV} cannot outperform MiSACO on any artificial test problem.

c) From Table III, MiSACO converges at least 30% faster than ACO_{MV} toward the optimal solutions on all the 30 artificial test problems except F2, F7, F12, F16, F17, F18, F20, F22, F27, and F28. On average, MiSACO reduces 43.98% *ASFEs* to reach the optimal solutions against ACO_{MV}. Specifically, MiSACO saves 52.49%, 34.64%, and 44.82% *ASFEs* on solving type-1, type-2, and type-3 artificial test problems, respectively.

TABLE III
ACCELERATION RATE OF MISACO AGAINST ACO_{MV} ON THE 30 ARTIFICIAL TEST PROBLEMS

Type 1		Type 2		Type 3	
Problem	*AR*	Problem	*AR*	Problem	*AR*
F1	58.34%	F11	46.44%	F21	53.39%
F2	*NA*	F12	*NA*	F22	*NA*
F3	37.36%	F13	43.80%	F23	39.21%
F4	65.25%	F14	44.83%	F24	58.34%
F5	64.47%	F15	51.47%	F25	61.66%
F6	54.62%	F16	27.46%	F26	44.87%
F7	*NA*	F17	*NA*	F27	*NA*
F8	34.47%	F18	2.67%	F28	19.45%
F9	50.74%	F19	31.66%	F29	40.58%
F10	54.63%	F20	28.79%	F30	41.08%
				Average *AR*	43.98%

2) Results on the Capacitated Facility Location Problems: From Table S-I of the supplementary file, for all the six capacitated facility location problems, MiSACO obtains better *AOFV* values than ACO_{MV}. According to the the Wilcoxon's rank-sum test, MiSACO beats ACO_{MV} on all these six problems.

3) Results on the Dubins Traveling Salesman Problems: From Table S-II of the supplementary file, for all the six Dubins traveling salesman problems, MiSACO provides better *AOFV* values. According to the Wilcoxon's rank-sum test, MiSACO performs better than ACO_{MV} on all these six problems.

From the above discussion, it can be concluded that the proposed multisurrogate-assisted selection and surrogate-assisted local search can significantly enhance the convergence accuracy and efficiency of ACO_{MV}.

C. Comparison With Other State-of-the-Art SAEAs

To further test the performance of MiSACO, we compared it with CAL-SAPSO [49], EGO-Hamming [66], EGO-Gower [63], and BOA-RF [67]. CAL-SAPSO is a SAEA for continuous EOPs. To make it have the capability to deal with categorical variables in EOPCCVs, we encoded each element in the candidate categorical set of each categorical variable into an ordered integer. The rounding operator was used in the optimization process. EGO-Hamming and EGO-Gower are two extended versions of efficient global optimization (EGO) [68]. By redefining the distance between two different categorical vectors, EGO-Hamming and EGO-Gower are able to solve EOPCCVs directly. In these two algorithms, the Hamming distance and Gower distance were employed to measure the difference between two different categorical vectors, respectively. BOA-RF is a variant of the Bayesian optimization [69]. Inspired by the sequential model-based algorithm configuration [67], we employed RF as the surrogate model and used the expected improvement as the acquisition function in BOA-RF.

The results provided by CAL-SAPSO, EGO-Hamming, EGO-Gower, BOA-RF, and MiSACO are recorded in Tables S-III–S-VI of the supplementary file. The detailed discussions are given as follows.

1) Results on the Artificial Test Problems:

a) From Table S-III of the supplementary file, MiSACO can provide smaller *AOFV* values than CAL-SAPSO on all the 30 artificial test problems. Moreover, each *AOFV* value of MiSACO is at least one order smaller than the corresponding *AOFV* value of CAL-SAPSO on all the 30 artificial test problems except F2, F7, F12, F17, F22, and F27. From Table S-IV of the supplementary file, MiSACO provides smaller *ASFEs* values than CAL-SAPSO on 24 artificial test problems (F1, F3–F6, F8–F11, F13–F16, F18–F21, F23–F26, and F28–F30). Thus, MiSACO can find the optimal solutions faster on these 24 artificial test problems. According to the Wilcoxon's rank-sum test, MiSACO surpasses CAL-SAPSO on 29 artificial test problems in terms of *AOFV* and 24 artificial test problems in terms of *ASFEs*, respectively.

b) Compared with EGO-Hamming, MiSACO provides better *AOFV* values and better *ASFEs* values on 29 artificial test problems (F1–F11 and F13–F30) and 24 artificial test problems (F1, F3–F6, F8–F11, F13–F16, F18–F21, F23–F26, and F28–F30), respectively. According to the Wilcoxon's rank-sum test, MiSACO has an edge over EGO-Hamming on 27 artificial test problems in terms of *AOFV* and 24 artificial test problems in terms of *ASFEs*, respectively. However, EGO-Hamming cannot beat MiSACO on any artificial test problem in terms of any performance indicator.

c) Compared with MiSACO, EGO-Gower provides worse *AOFV* values on 27 artificial test problems and better *AOFV* values on only three artificial test problems (F12, F17, and F22). With respect to *ASFEs*, EGO-Gower is worse than MiSACO on 24 artificial test problems (F1, F3–F6, F8–F11, F13–F16, F18–F21, F23–F26, and F28–F30), and cannot provide any better value on any artificial test problem. According to the Wilcoxon's rank-sum test, MiSACO beats EGO-Gower on 26 artificial test problems in terms of *AOFV* and 24 artificial test problems in terms of *ASFEs*, respectively.

d) BOA-RF provides worse *AOFV* and *ASFEs* values than MiSACO on 30 and 24 (i.e., F1, F3–F6, F8–F11, F13–F16, F18–F21, F23–F26, and F27–F30) artificial test problems, respectively. Meanwhile, with respect to both *AOFV* and *ASFEs*, MiSACO does not provide any worse value than BOA-RF on any artificial test problem. According to the Wilcoxon's rank-sum test, MiSACO beats BOA-RF on 30 artificial test problems in terms of *AOFV* and 24 artificial test problems in terms of *ASFEs*, respectively. Moreover, MiSACO does not lose on any artificial test problem.

2) Results on the Capacitated Facility Location Problems: From Table S-V of the supplementary file, MiSACO obtains better *AOFV* values than CAL-SAPSO, EGO-Hamming, and BOA-RF on all the six capacitated facility location problems. Moreover, MiSACO is better than EGO-Gower on five problems. According to the Wilcoxon's rank-sum test, MiSACO beats CAL-SAPSO, EGO-Hamming, EGO-Gower, and BOA-RF on six, six, three, and five problems, respectively.

3) Results on the Dubins Traveling Salesman Problems: From Table S-VI of the supplementary file, MiSACO is better than CAL-SAPSO, EGO-Hamming, EGO-Gower, and BOA-RF on all the six Dubins traveling salesman problems in terms of *AOFV*. According to the Wilcoxon's rank-sum test, MiSACO beats the four competitors on all the six problems.

The above results demonstrate that, overall, the performance of MiSACO is better than that of the four state-of-the-art competitors in terms of both *AOFV* and *ASFEs*. The superiority of MiSACO against the four competitors can be attributed to the fact that these four competitors mainly extend surrogate models for continuous functions, thus having limited capabilities to cope with EOPCCVs.

VI. Real-World Applications

In this section, MiSACO was applied to solve two EOPCCVs in the real world, that is: 1) the topographical design of stiffened plates against blast loading and 2) the lightweight and crashworthiness design for the side body of an automobile.

A. Topographical Design of Stiffened Plates Against Blast Loading

Presently, the explosion caused by accidents and terrorist attacks has attracted much attention. In order to protect the personnel and facilities from the explosion damage, the research on blast-resistant structures is of great significance. As a kind of structure that can effectively deal with blast loading, stiffened plates have been widely studied [71], [72]. Recently,

Liu *et al.* [70] designed the topographical structures of two new kinds of stiffened plates. Based on their research work, we tried to assign different materials for different stiffeners, thus further improving the structural resistance of the stiffened plates against blast loading.

As shown in Fig. S-1(a) of the supplementary file, one kind of the stiffened plate in [70] is considered in this article. The front plate of the stiffened plate is a square shape with $L \times L = 250$ mm $\times 250$ mm, and its thickness is 1 mm. The height of each stiffener is $H = 10$ mm. Fig. S-1(b) in the supplementary file depicts the variable distribution of this structure. We can observe that this structure is 1/8 symmetric; hence, the thicknesses and materials of the 13 stiffeners are considered as the design variables, that is, $x_1^{\text{thick}}, \ldots, x_{13}^{\text{thick}}$ and $x_1^{\text{mat}}, \ldots, x_{13}^{\text{mat}}$. Same as [70], the maximum deflection of the center point of the plate (denoted as D) is employed to assess the structural resistance, and the mass of the plate (denoted as $MASS$) is constrained within a certain value (denoted as M^*). Thus, this design problem can be described as follows:

$$\begin{aligned} \min:\ & D\left(\mathbf{x}^{\text{thick}}, \mathbf{x}^{\text{mat}}\right) \\ \text{s.t.}\ & MASS\left(\mathbf{x}^{\text{thick}}, \mathbf{x}^{\text{mat}}\right) \le M^* \\ & \mathbf{x}^{\text{thick}} = \left(x_1^{\text{thick}}, \ldots, x_{13}^{\text{thick}}\right) \\ & \mathbf{x}^{\text{mat}} = \left(x_1^{\text{mat}}, \ldots, x_{13}^{\text{mat}}\right) \\ & x_1^{\text{thick}}, \ldots, x_{13}^{\text{thick}} \in [0, 2] \\ & x_1^{\text{mat}}, \ldots, x_{13}^{\text{mat}} \in \{\text{MAT1}, \ldots, \text{MAT5}\} \end{aligned} \tag{20}$$

where MAT1, ..., MAT5 represent the five categories of steel: 1) mild steel; 2) IF300/420; 3) DP350/600; 4) IF260/400; and 5) DP500/800. The value of M^* was set to 0.09705 kg.

MiSACO was used to optimize the structure of the stiffened plate, and *MaxFEs* was set to 800. For comparison, we also employed EGO-Gower and GA to optimize the structure. Since the value of *MASS* can be calculated directly according to the density of materials and the thicknesses of stiffeners, the constraint in (20) is not an expensive function. Thus, in these three algorithms, the feasibility rule [73][2] was used to deal with the constraint. Table IV summarizes the values of D and *MASS* of the structures reported in [70] and presented by the three algorithms. Moreover, the topographical structures of the four corresponding stiffened plates are shown in Fig. S-2 of the supplementary file.

From Table IV, compared with the structure reported in [70], the structure provided by MiSACO improves the D value by 19.09%. Meanwhile, these two structures have similar structure mass. This indicates that assigning different materials for different stiffeners can effectively improve the structural resistance. Compared with the structures provided by EGO-Gower and GA, the structure resulting from MiSACO has the best D value. Moreover, Fig. S-3 in the supplementary file plots the convergence curves derived from EGO-Gower, GA, and MiSACO. It can be observed from Fig. S-3 that MiSACO can provide the best convergence performance.

[2]The feasibility rule compares two solutions as follows: 1) between two infeasible solutions, the one with smaller degree of constraint violation is preferred; 2) if one solution is infeasible and the other is feasible, the feasible one is preferred; and 3) between two feasible solutions, the one with a better objective function value is preferred.

TABLE IV
RESULTS ABOUT THE STRUCTURES OF THE STIFFENED PLATES REPORTED IN [70] AND OBTAINED BY EGO-GOWER, GA, AND MISACO

Status	Reported in [70]	EGO-Gower	GA	MiSACO
D	8.020 mm	6.828 mm	7.880 mm	6.489 mm
$MASS$	0.124 kg	0.124 kg	0.123 kg	0.124 kg

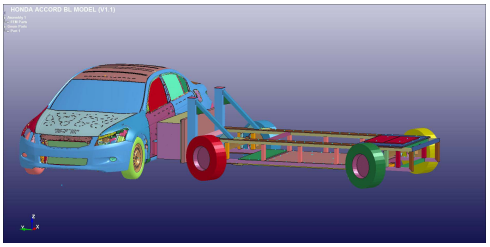


Fig. 5. Side crash FEA model considered in this article. In this model, Honda Accord was employed as the baseline.

B. Lightweight and Crashworthiness Design for the Side Body of an Automobile

In the field of automotive engineering, it is desirable to design an automobile body with low mass and high crashworthiness, thus reducing the fuel consumption and improving the safety of the automobile. In this article, we focus on the design of the side body of an automobile.

The side crash FEA model is shown in Fig. 5. The FEA model was established according to Honda Accord, and the details about this model can be found from the technical report provided by Singh [15]. We selected five thin-walled plate parts from the side body of the automobile, and tried to redesign their thicknesses and materials. These five thin-walled plate parts are shown in Fig. 6. As it is very time consuming to perform a side crash simulation by using the FEA model described in Fig. 5, in this article, we used a simplified FEA model, as shown in Fig. S-4 in the supplementary file. It contains B-pillar and a part of the side door of the original automobile. On our computer, this simplified FEA model took about 50 min to execute a run.

In this article, the following three indicators were used to assess the structure.

1) *Maximum Invasion at the Middle of B-Pillar:* Reducing the maximum invasion at the middle of B-pillar can improve the safety of passengers in the automobile. We denote the maximum invasion at the middle of B-pillar as *FI*.
2) *Maximum Invasion Velocity at the Middle of B-Pillar:* Commonly, a small maximum invasion velocity at the middle of B-pillar can reduce the probability of passenger injury. In this article, the maximum invasion velocity at the middle of B-pillar is denoted as *FV*.
3) *Mass of the Side Body:* This indicator is to evaluate the lightweight level of the automobile. We denote this indicator as *MASS*.

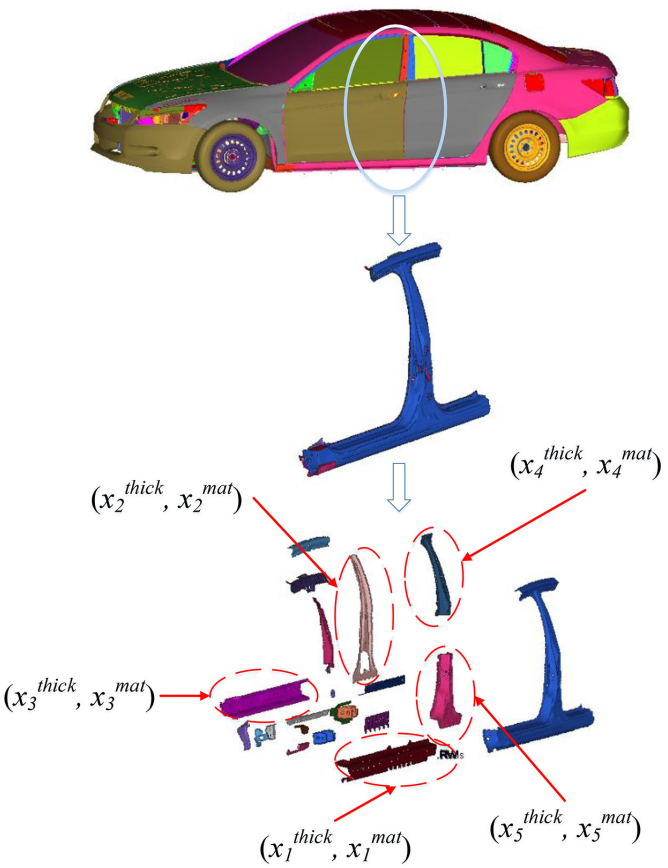


Fig. 6. Five thin-walled plate parts that need to be optimized.

According to the simplified FEA model, the values of *FI*, *FV*, and *MASS* of the original design are 240.04 mm, 9674.7 mm/s, and 0.0252*t*, respectively.[3] Our aim is to reduce the *FI* value without increasing the *FV* and *MASS* values. Overall, this design can be described as follows:

$$\begin{aligned} \min:\ & FI\left(\mathbf{x}^{\text{thick}}, \mathbf{x}^{\text{mat}}\right) \\ \text{s.t.}\ & FV\left(\mathbf{x}^{\text{thick}}, \mathbf{x}^{\text{mat}}\right) \le 9674.7 \\ & MASS\left(\mathbf{x}^{\text{thick}}, \mathbf{x}^{\text{mat}}\right) \le 0.0252 \\ & \mathbf{x}^{\text{thick}} = \left(x_1^{\text{thick}}, \ldots, x_5^{\text{thick}}\right) \\ & \mathbf{x}^{\text{mat}} = \left(x_1^{\text{mat}}, \ldots, x_5^{\text{mat}}\right) \\ & x_1^{\text{thick}}, \ldots, x_5^{\text{thick}} \in [0.2, 2] \\ & x_1^{\text{mat}}, \ldots, x_5^{\text{mat}} \in \{\text{MAT1}, \ldots, \text{MAT6}\} \end{aligned} \tag{21}$$

where $x_1^{\text{thick}}, \ldots, x_5^{\text{thick}}$ are the thickness variables of the five thin-walled plate parts, and $x_1^{\text{mat}}, \ldots, x_5^{\text{mat}}$ are the material variables of the five thin-walled plate parts. There are six kinds of materials for each thin-walled plate part: MAT1, ..., MAT6, which represent six kinds of steel: 1) DP350/600; 2) DP500/800; 3) HSLA350/450; 4) IF140/270; 5) IF260/410; and 6) IF300/420.

[3]In our experiment, LS-DYNA was employed as the simulation solver, and its version was "ls971s R4.2." The whole simulation process was executed in "Windows 7 64-bit Ultimate."

TABLE V
RESULTS OF THE ORIGINAL DESIGN AND THE DESIGN PROVIDED BY MISACO

Status	Original Design	MiSACO
FI	240.04 *mm*	177.26 *mm*
FV	9674.7 *mm/s*	9659.7 *mm/s*
MASS	0.0252 *t*	0.0251 *t*

We consumed 500 FEs to optimize the side body by using MiSACO. The entire optimization process took about $(50*500)/(60*24) \approx 17.36$ days. The feasibility rule [73] was used to deal with the constraints in (21). For $FI(\mathbf{x}^{\text{thick}}, \mathbf{x}^{\text{mat}})$ and $FV(\mathbf{x}^{\text{thick}}, \mathbf{x}^{\text{mat}})$, the surrogate models were constructed independently. Since $\text{MASS}(\mathbf{x}^{\text{thick}}, \mathbf{x}^{\text{mat}})$ can be calculated directly, we did not establish any surrogate model for it. The values of *FI*, *FV*, and *MASS* of the original design and MiSACO are listed in Table V. Compared with the original design, MiSACO improves the *FI* value by 26.16%. At the same time, the *FV* and *MASS* values of the original design are similar to those provided by MiSACO. It means that the crashworthiness of the side body of the automobile can be greatly enhanced without adding the mass of the structure. Thus, the safety of the automobile can be significantly improved.

The above experiments reveal that MiSACO could be an effective tool to solve EOPCCVs in the real world.

VII. CONCLUSION

Many real-world applications can be modeled as EOPCCVs. However, few attempts have been made on solving EOPCCVs. In this article, we proposed a multisurrogate-assisted ACO, called MiSACO, to solve EOPCCVs. MiSACO contained two main strategies: 1) the multisurrogate-assisted selection and 2) the surrogate-assisted local search. In the former, three selection operators (i.e., the RBF-based selection, LSBT-based selection, and random selection) were employed. The aim of them is to help MiSACO to deal with different types of EOPCCVs robustly and prevent MiSACO from being misled by inaccurate surrogate models. In the latter, we focused on improving the quality of the current best solution by further optimizing its continuous variables. This strategy was implemented by constructing an RBF for only continuous variables and optimizing this RBF by using SQP. From the comparative studies on the three sets of test problems, the effectiveness of MiSACO was verified. We also applied MiSACO to solve two EOPCCVs in the real world: 1) the topographical design of stiffened plates against blast loading and 2) the lightweight and crashworthiness design for the side body of an automobile. The results showed that MiSACO can effectively solve them.

In the future, we will apply MiSACO to solve large-scale EOPCCVs (e.g., increase the number of categorical variables and/or the size of candidate categorical sets). When the scale of an EOPCCV becomes larger, it is not easy to construct an accurate surrogate model. Some techniques such as dimension reduction methods will be further incorporated into MiSACO to deal with large-scale EOPCCVs. In addition, in the surrogate-assisted local search, if the best solution does not

change for several generations, it will be optimized repeatedly; thus, the computational resources may be wasted. In the future, we will try to address this limitation.

REFERENCES

[1] L. Costa and P. Oliveira, "Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems," *Comput. Chem. Eng.*, vol. 25, nos. 2–3, pp. 257–266, 2001.

[2] M. A. Abramson, C. Audet, J. W. Chrissis, and J. G. Walston, "Mesh adaptive direct search algorithms for mixed variable optimization," *Optim. Lett.*, vol. 3, no. 1, pp. 35–47, 2009.

[3] C. Audet and J. E. Dennis, Jr., "Pattern search algorithms for mixed variable programming," *SIAM J. Optim.*, vol. 11, no. 3, pp. 573–594, 2001.

[4] R. Li *et al.*, "Mixed integer evolution strategies for parameter optimization," *Evol. Comput.*, vol. 21, no. 1, pp. 29–64, 2013.

[5] J. Lampinen and I. Zelinka, "Mixed integer-discrete-continuous optimization by differential evolution," in *Proc. 5th Int. Conf. Soft Comput.*, 1999, pp. 71–76.

[6] J. Wang and L. Wang, "Decoding methods for the flow shop scheduling with peak power consumption constraints," *Int. J. Prod. Res.*, vol. 57, no. 10, pp. 3200–3218, 2019.

[7] D. Lei, M. Li, and L. Wang, "A two-phase meta-heuristic for multiobjective flexible job shop scheduling problem with total energy consumption threshold," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1097–1109, Mar. 2019.

[8] F. Wang, Y. Li, A. Zhou, and K. Tang, "An estimation of distribution algorithm for mixed-variable newsvendor problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 479–493, Jun. 2020.

[9] W.-L. Liu, Y.-J. Gong, W.-N. Chen, Z. Liu, H. Wang, and J. Zhang, "Coordinated charging scheduling of electric vehicles: A mixed-variable differential evolution approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 12, pp. 5094–5109, Dec. 2020.

[10] T. Liao, K. Socha, M. A. M. de Oca, T. Stutzle, and M. Dorigo, "Ant colony optimization for mixed-variable optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 503–518, Aug. 2014.

[11] Y. Lin, Y. Liu, W.-N. Chen, and J. Zhang, "A hybrid differential evolution algorithm for mixed-variable optimization problems," *Inf. Sci.*, vol. 466, pp. 170–188, Oct. 2018.

[12] J. Müller, C. A. Shoemaker, and R. Piché, "SO-MI: A surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems," *Comput. Oper. Res.*, vol. 40, no. 5, pp. 1383–1400, 2013.

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

[14] M. Herrera, A. Guglielmetti, M. Xiao, and R. F. Coelho, "Metamodel-assisted optimization based on multiple kernel regression for mixed variables," *Structural Multidiscipl. Optim.*, vol. 49, no. 6, pp. 979–991, 2014.

[15] H. Singh, "Mass reduction for light-duty vehicles for model years 2017–2025," U.S. Dept. Transp., Washington, DC, USA, Rep. DTNH22-13-D-00298, 2012.

[16] J. Fang, G. Sun, N. Qiu, G. P. Steven, and Q. Li, "Topology optimization of multicell tubes under out-of-plane crushing using a modified artificial bee colony algorithm," *J. Mech. Design*, vol. 139, no. 7, 2017, Art. no. 071403.

[17] G. Sun, H. Zhang, J. Fang, G. Li, and Q. Li, "A new multi-objective discrete robust optimization algorithm for engineering design," *Appl. Math. Model.*, vol. 53, pp. 602–621, Jan. 2018.

[18] F. Pan, P. Zhu, and Y. Zhang, "Metamodel-based lightweight design of B-pillar with TWB structure via support vector regression," *Comput. Structures*, vol. 88, nos. 1–2, pp. 36–44, 2010.

[19] F. Xiong, D. Wang, S. Chen, Q. Gao, and S. Tian, "Multi-objective lightweight and crashworthiness optimization for the side structure of an automobile body," *Structural Multidiscipl. Optim.*, vol. 58, no. 4, pp. 1823–1843, 2018.

[20] H. Safari, H. Nahvi, and M. Esfahanian, "Improving automotive crash-worthiness using advanced high strength steels," *Int. J. Crashworthiness*, vol. 23, no. 6, pp. 645–659, 2018.

[21] Q. Yang *et al.*, "Adaptive multimodal continuous ant colony optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 191–205, Apr. 2017.

[22] Z.-M. Huang, W.-N. Chen, Q. Li, X.-N. Luo, H.-Q. Yuan, and J. Zhang, "Ant colony evacuation planner: An ant colony system with incremental flow assignment for multipath crowd evacuation," *IEEE Trans. Cybern.*, early access, Sep. 10, 2020. [Online]. Available: https://doi.org/10.1109/TCYB.2020.3013271

[23] Z. 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.

[24] J. Y. Li, Z. H. Zhan, R. D. Liu, C. Wang, S. Kwong, and J. Zhang, "Generation-level parallelism for evolutionary computation: A pipeline-based parallel particle swarm optimization," *IEEE Trans. Cybern.*, early access, Nov. 4, 2020. [Online]. Available: https://10.1109/TCYB.2020.3028070

[25] F. Wei, W.-N. Chen, Q. Yang, J. Deng, X. Luo, H. Jin, and J. Zhang, "A classifier-assisted level-based learning swarm optimizer for expensive optimization," *IEEE Trans. Evol. Comput.*, early access, Apr. 19, 2020. [Online]. Available: https://doi.org/10.1109/TEVC.2020.3017865

[26] X. Xiang, Y. Tian, J. Xiao, and X. Zhang, "A clustering-based surrogate-assisted multiobjective evolutionary algorithm for shelter location problem under uncertainty of road networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7544–7555, Dec. 2020.

[27] 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.

[28] S. H. Wu, Z. H. Zhan, and J. Zhang, "SAFE: Scale-adaptive fitness evaluation method for expensive optimization problems," *IEEE Trans. Evol. Comput.*, early access, Jan. 14, 2021. [Online]. Available: https://10.1109/TEVC.2021.3051608

[29] B. Liu, Q. Zhang, and G. G. Gielen, "A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 180–192, Apr. 2014.

[30] Y. Wang, D. Q. Yin, S. Yang, and G. Sun, "Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1642–1656, May 2019.

[31] Y. Sun, H. Wang, B. Xue, Y. Jin, G. G. Yen, and M. Zhang, "Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 350–364, Apr. 2020.

[32] S. Soltani and R. D. Murch, "A compact planar printed MIMO antenna design," *IEEE Trans. Antennas Propag.*, vol. 63, no. 3, pp. 1140–1149, Mar. 2015.

[33] Y. Del Valle, G. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. Harley, "Particle swarm optimization: Basic concepts, variants and applications in power systems," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 171–195, Apr. 2008.

[34] X. Wang, G. G. Wang, B. Song, P. Wang, and Y. Wang, "A novel evolutionary sampling assisted optimization method for high dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 815–827, Oct. 2019.

[35] X. Sun, D. Gong, Y. Jin, and S. Chen, "A new surrogate-assisted interactive genetic algorithm with weighted semisupervised learning," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 685–698, Apr. 2013.

[36] N. Namura, K. Shimoyama, and S. Obayashi, "Expected improvement of penalty-based boundary intersection for expensive multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 6, pp. 898–913, Dec. 2017.

[37] A. Habib, H. K. Singh, T. Chugh, T. Ray, and K. Miettinen, "A multiple surrogate assisted decomposition based evolutionary algorithm for expensive multi/many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 1000–1014, Dec. 2019.

[38] Y. Lian and M. S. Liou, "Multiobjective optimization using coupled response surface model and evolutionary algorithm," *Eur. Radiol.*, vol. 43, no. 6, pp. 1316–1325, 2005.

[39] Y. S. Ong, P. B. Nair, and A. J. Keane, "Evolutionary optimization of computationally expensive problems via surrogate modeling," *AIAA J.*, vol. 41, no. 4, pp. 687–696, 2003.

[40] 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.

[41] B. Liu, H. Yang, and M. J. Lancaster, "Global optimization of microwave filters based on a surrogate model-assisted evolutionary algorithm," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 6, pp. 1–10, Jun. 2017.

[42] J. Tian, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Multiobjective infill criterion driven Gaussian process-assisted particle swarm optimization of high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 459–472, Jun. 2019.

[43] C. Sun, Y. Jin, R. Cheng, J. Ding, and J. Zeng, "Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 644–660, Aug. 2017.

[44] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with Gaussian process model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456–474, Jun. 2010.

[45] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Jun. 2010.

[46] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 129–142, Feb. 2018.

[47] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 329–355, Jun. 2010.

[48] M. N. Le, Y. S. Ong, S. Menzel, Y. Jin, and B. Sendhoff, "Evolution by adapting surrogates," *Evol. Comput.*, vol. 21, no. 2, pp. 313–340, 2013.

[49] H. Wang, Y. Jin, and J. Doherty, "Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2664–2677, Sep. 2017.

[50] 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. [Online]. Available: https://10.1109/TCYB.2020.3008280

[51] 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.

[52] X. Lu, T. Sun, and K. Tang, "Evolutionary optimization with hierarchical surrogates," *Swarm Evol. Comput.*, vol. 47, pp. 21–32, Jun. 2019.

[53] F. Li, X. Cai, and L. Gao, "Ensemble of surrogates assisted particle swarm optimization of medium scale expensive problems," *Appl. Soft Comput.*, vol. 74, pp. 291–305, Jan. 2019.

[54] D. Guo, Y. Jin, J. Ding, and T. Chai, "Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1012–1025, Mar. 2019.

[55] T. Bartz-Beielstein and M. Zaefferer, "Model-based methods for continuous and discrete global optimization," *Appl. Soft Comput.*, vol. 55, pp. 154–167, Jun. 2017.

[56] A. Liaw and M. Wiener, "Classification and regression by random forest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[57] C. D. Sutton, "Classification and regression trees, bagging, and boosting," *Handbook Stat.*, vol. 24, pp. 303–329, 2005.

[58] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[59] H. Wang and Y. Jin, "A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 536–549, Feb. 2020.

[60] S. Nguyen, M. Zhang, and K. C. Tan, "Surrogate-assisted genetic programming with simplified models for automated design of dispatching rules," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2951–2965, Sep. 2017.

[61] B. Yuan, B. Li, T. Weise, and X. Yao, "A new memetic algorithm with fitness approximation for the defect-tolerant logic mapping in crossbar-based nanoarchitectures," *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 846–859, Dec. 2014.

[62] J. Pelamatti, L. Brevault, M. Balesdent, E.-G. Talbi, and Y. Guerin, "Efficient global optimization of constrained mixed variable problems," *J. Global Optim.*, vol. 73, no. 3, pp. 583–613, 2019.

[63] M. Halstrup, "Black-box optimization of mixed discrete-continuous optimization problems," Ph.D. dissertation, Dept. Comput. Sci., Technische Universität Dortmund, Dortmund, Germany, 2016.

[64] C. Sun, Y. Jin, J. Zeng, and Y. Yu, "A two-layer surrogate-assisted particle swarm optimization algorithm," *Soft Comput.*, vol. 19, no. 6, pp. 1461–1475, 2015.

[65] C. Li, S. Gupta, S. Rana, V. Nguyen, S. Venkatesh, and A. Shilton, "High dimensional Bayesian optimization using dropout," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 2096–2102.

[66] M. Zaefferer, "Surrogate models for discrete optimization problems," Ph.D. dissertation, Appl. Sci., Technische Universität Dortmund, Dortmund, Germany, 2018.

[67] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *International Conference on Learning and Intelligent Optimization*, C. A. Coello Coello, Ed. Heidelberg, Germany: Springer, Jan. 2011, pp. 507–523.
F. Hutter, H. H. Hoos, and K. Leyton-Brown. "Sequential model-based optimization for general algorithm configuration", In International Conference on Learning and Intelligent Optimization.

[68] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Global Optim.*, vol. 13, no. 4, pp. 455–492, 1998.

[69] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016.

[70] T. Liu, G. Sun, J. Fang, J. Zhang, and Q. Li, "Topographical design of stiffener layout for plates against blast loading using a modified algorithm," *Structural Multidiscipl. Optim.*, vol. 59, no. 2, pp. 335–350, 2019.

[71] H. Liu, B. Li, Z. Yang, and J. Hong, "Topology optimization of stiffened plate/shell structures based on adaptive morphogenesis algorithm," *J. Manuf. Syst.*, vol. 43, pp. 375–384, Apr. 2017.

[72] C. Zheng, X. Kong, W. Wu, and F. Liu, "The elastic-plastic dynamic response of stiffened plates under confined blast load," *Int. J. Impact Eng.*, vol. 95, pp. 141–153, Sep. 2016.

[73] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Meth. Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 311–338, 2000.

Jiao Liu received the B.S. degree in process equipment and control engineering and the M.S. degree in power engineering and engineering thermophysics from the Taiyuan University of Technology, Taiyuan, China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree in control science and engineering with Central South University, Changsha, China.

His current research interests include evolutionary computation, mixed-integer programming, and lightweight design of automobiles.

Yong Wang (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from the Central South University, Changsha, China, in 2011.

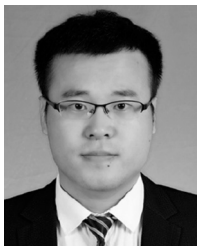
He is a Professor with the School of Automation, Central South University. His current research interests include the theory, algorithm design, and interdisciplinary applications of computational intelligence.

Prof. Wang was a recipient of the Cheung Kong Young Scholar by the Ministry of Education, China, in 2018, and a Web of Science Highly Cited Researcher in Computer Science in 2017 and 2018. He is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the *Swarm and Evolutionary Computation*.

Guangyong Sun received the Ph.D. degree in mechanical engineering from Hunan University, Changsha, China, in 2011.

He is a Professor with the State Key Laboratory of Advanced Design and Manufacture for Vehicle Body, Hunan University. His current research interests include lightweight design, structural optimization, and automotive safety.

Prof. Sun was a Web of Science Highly Cited Researcher in Engineering in 2020. He is an Associate Editor of *Thin-Walled Structures* and *Applied Composite Materials*.



Tong Pang received the Ph.D. degree in mechanical engineering from Hunan University, Changsha, China, in 2019.

He is currently a Research Associate with the College of Mechanical and Vehicle Engineering, Hunan University. His current research interests include lightweight design, structural optimization, and automotive crashworthiness.