

Research Article

A Globally Optimal Robust Design Method for Complex Systems

Yue Chen ^{1,2}, Jian Shi ^{1,2} and Xiao-jian Yi ^{1,3,4}

¹Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China

²School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China

³Department of Overall Technology, China North Vehicle Research Institute, Beijing 100072, China

⁴School of Mechatronic Engineering, Beijing Institute of Technology, Beijing 100081, China

Correspondence should be addressed to Jian Shi; jshi@iss.ac.cn and Xiao-jian Yi; yixiaojianbit@sina.cn

Received 19 December 2019; Revised 30 March 2020; Accepted 8 April 2020; Published 11 May 2020

Academic Editor: Carlos Aguilar-Ibanez

Copyright © 2020 Yue Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The uncertainty of the engineering system increases with the growing complexity of the engineering system; therefore, the tolerance to the uncertainty is essential. In the design phase, the output performance should reach the design criterion, even under large variations of design parameters. The tolerance to design parameter variations may be measured by the size of a solution space in which the output performance is guaranteed to deliver the required performance. In order to decouple dimensions, a maximum solution hyperbox, expressed by intervals with respect to each design parameter, is sought. The proposed approach combines the metaheuristic algorithm with the DIRECT algorithm where the former is used to seek the maximum size of hyperbox, and the latter is used as a checking technique that guarantees the obtained hyperbox is indeed a solution hyperbox. There are three advantages of the proposed approach. First, it is a global search and has a considerable high possibility to produce the globally maximum solution hyperbox. Second, it can be used for both analytically known and black-box performance functions. Third, it guarantees that any point selected within the obtained hyperbox satisfies the performance criterion as long as the performance function is continuous. Furthermore, the proposed approach is illustrated by numerical examples and real examples of complex systems. Results show that the proposed approach outperforms the GHZ and CES-IA methods in the literature.

1. Introduction

With the rapid development of technology in recent years, the systems applied in various engineering areas such as electronics, communication, and networking have become more and more complex. The increasing complexity of the system provides a new source for the uncertainty. Uncertainty arises because some design parameters cannot yet be specified exactly or they may be changed over the course of development [1, 2]. However, the higher requirements on systems place great importance on analyzing the uncertainty.

Traditional optimization techniques seek an optimum in the design space. Typically, without considering the uncertainty, the optimum design is frequently pushed to the constraint boundary of the design. Consequently, this type of optimum design may be nonrobust and sensitive to parameter variabilities. Some authors even believe that optimization is actually just the opposite of robustness [3].

As reliability is nevertheless mandatory, engineers have to look for robust designs in order to avoid unexpected deviations from the nominal performance [4]. To this end, more advanced methods have been developed to include uncertainties of the parameters and robustness criteria in the optimization. These include

- (i) Methods for selecting a design that is insensitive to the variations without eliminating possible variations of design parameters, i.e., *robust design optimization* (RDO), see [5–7]
- (ii) Methods for identifying optimum designs which are characterized by a low probability of failure, i.e., *reliability-based design optimization* (RBDO), see [1, 2, 8]
- (iii) Methods for computing the effect of variability of input parameters on variability of objective function value, i.e., *sensitivity analysis* (SA), see [9]

Unfortunately, RDO, RBDO, and SA suffer from certain disadvantages which exclude them from some important applications. RBDO and RDO deal with models in the case where the variability of design parameters is given. If, however, the variability of the design parameters is not known completely, it has to be estimated, which is not always possible. When applying SA, information on how to improve a nonrobust solution is limited, what design parameter needs to be adjusted and what value it should assume is unknown.

In this paper, we aim to infer the maximum uncertainty or variability of design parameters that we can tolerate without violating the required performance. That is, we focus on finding a maximum box-shaped solution space rather than a single optimum solution. The box-shaped solution space, representing a hyperbox, can be expressed by intervals for each design parameter and has the following benefits:

- (i) The performance function delivers the required performance as long as the design parameters lie in the box-shaped solution space
- (ii) For a design to be good, the choice of the value of one parameter within its assigned interval does not depend on the values of other parameters as long as they are within their respective intervals
- (iii) The intervals may be used to assess the robustness and sensitivity of the uncertain design parameters by the widths of the associated intervals.
- (iv) The intervals are independent of each other and may be combined with intervals of other disciplines, enabling distributed design in separate disciplines

Two categories of approaches that solve similar problems could be identified in the literature. The first category is based on data-mining and machine-learning techniques [10–13]. The stochastic approach proposed in [10] combines query and online learning, which probes a box-shaped candidate solution space by stochastic sampling and then readjusts its boundaries in order to remove designs with insufficient performance and explore more design space that has not been probed before. The quality of the results and the efficiency of this stochastic approach are studied in detail in [11]. This stochastic approach, however, produces a box-shaped solution space that may contain some bad designs and may not have the best size because the Monte Carlo sampling is used to estimate the locations of good and bad designs. The second category is based on an analytical technique, called interval arithmetic [14]. The algorithm presented in [15], for example, applies interval arithmetic within an iterative optimization scheme to compute the feasibility of candidate boxes. Interval arithmetic, however, limits the applicability of the algorithm, as the accuracy of the results depends on the problem and cannot be assessed for general cases. Another drawback of the algorithm in [15] is that the performance function needs to be analytically known and cannot be treated as a black box [16].

The maximum box-shaped solution space must both fulfill the “quantitative” viewpoint (i.e., its volume is maximum) and

“feasible” viewpoint (i.e., it is indeed a solution space). To address these problems, we propose an innovative approach which combines the metaheuristic algorithm with the DIRECT algorithm. More specifically, we use the metaheuristic algorithm to search the maximum hyperbox and use the DIRECT algorithm to guarantee that all designs in the obtained hyperbox satisfy the performance criterion (i.e., the obtained hyperbox is indeed a solution hyperbox). Actually, in the proposed approach, metaheuristic algorithms do not rely on mathematical properties to be applied, and DIRECT algorithm only requires evaluations of the performance function. Therefore, our approach in essence treats the performance function as a black box and can be applied to various types of practical engineering problems for complex systems, such as reliability allocation, availability allocation, and maintenance analysis. As a real example of the complex system, the reliability allocation problem of the power-shift steering transmission control system (PSSTCS) of a heavy vehicle is analyzed by the proposed approach in this paper.

The proposed approach is similar to the traditional RDO, as the variations of design parameters are considered. However, it differs from the traditional method in which the optimization is to seek the intervals of a permissible design range rather than a single design, and interval boundaries are used as degrees of freedom rather than design parameters.

The paper is organized as follows. In Section 2, motivated by a problem from engineering practice, the optimization problem is formulated for identifying the maximum hyperbox which guarantees the performance requirement. Section 3 introduces the DIRECT algorithm. Section 4 presents in detail the proposed approach. Section 5 discusses the numerical performance of the proposed approach. In Section 6, engineering cases are investigated. Section 7 presents some concluding remarks.

2. Motivation and Problem Statement

2.1. Example Problem. We consider the voltage divider shown in Figure 1 in [15]. The terminal voltage V_{ab} is expressed as $V_{ab} = V_0 R / (R + R_a)$ and the resistance $R_{in} = R + R_a$.

Good designs fulfilling the design goals should satisfy

$$\begin{aligned} 4.5V &\leq V_{ab} \leq 5.5V, \\ 80\Omega &\leq R_{in} \leq 120\Omega. \end{aligned} \quad (1)$$

Figure 2 shows the complete solution space (gray region). However, the classical optimum that minimizes R_{in} and maximizes V_{ab} is not robust. The immediate neighborhood of the optimum includes designs that violate condition (1) and thus fails to meet at least one design goal.

The classical optimum would not be a suitable choice because of the lack of robustness. Using the complete solution space instead would also be impractical, as the tolerable range for R_a would depend on the choice of R . For decoupling the design parameters, hyperboxes contained in the complete solution space, expressed by intervals for each design parameter, are desirable.

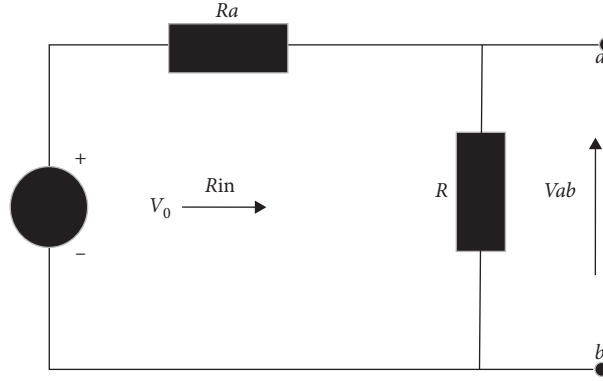


FIGURE 1: Voltage divider example.

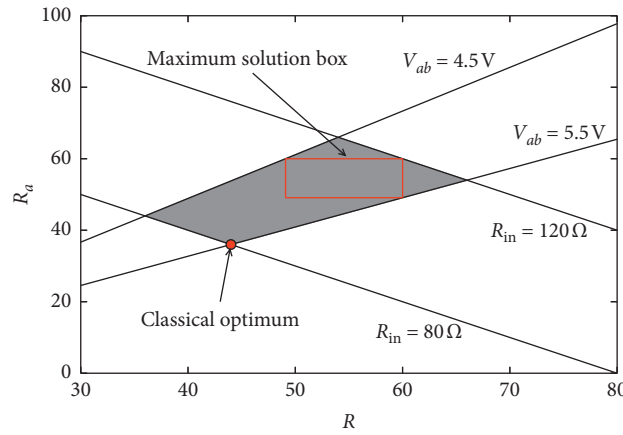


FIGURE 2: The complete solution space (gray region) and the solution box with the largest volume (region delimited by red lines).

The robustness of uncertain design parameters [17] is defined as the widths of the associated intervals whereby the performance criterion (1) is met. Maximum robustness is provided if the intervals are as large as possible. The choice of intervals is not unique, and the size of the interval for R_a depends on the choice of the interval for R . The approach proposed here is to maximize the volume of the solution hyperbox, and the resulting unique intervals are shown in Figure 2.

2.2. Problem Statement. Let $\mathbf{x} = (x_1, x_2, \dots, x_m)$ represent the *design parameters* or *design points* or *designs*, where m is the number of dimensions. The *output performance* at \mathbf{x} is given by

$$y = f(\mathbf{x}), \quad (2)$$

with f being the *performance function*. If the analytical form of f is not known, it is considered as a black box, and y may have to be computed numerically. In typical optimization problems, the performance is optimized, that is, a design \mathbf{x} is sought such that y obtains an extreme value. In the approach presented here, the performance has to be *sufficient* by satisfying the *performance criterion*:

$$f(\mathbf{x}) \geq f^c, \quad (3)$$

with the threshold value f^c . The performance criterion is the requirement of performance. For example, if the performance function is the reliability, the performance criterion is that the reliability should be above the given threshold (e.g., 0.99).

Design points satisfying (3) are called *good*, otherwise are called *bad*. The *design space* $D(\mathbf{x}^l, \mathbf{x}^u)$ is assumed to be continuous and defined by the lower and upper bounds of the design parameters, given by \mathbf{x}^l and \mathbf{x}^u , that is, $D(\mathbf{x}^l, \mathbf{x}^u) = [\mathbf{x}^l, \mathbf{x}^u] = \{\mathbf{x}: x_i^l \leq x_i \leq x_i^u, i = 1, \dots, m\}$. The *complete solution space* is the set of all good design points in the design space. The shape of the complete solution space depends on the output performance and the performance criterion given by (2) and (3), respectively. In general, boundaries of the range of one parameter depend on the values of other parameters. In order to obtain interval boundaries of each parameter that are independent of other parameters, only subspaces that are hyperboxes are considered. The hyperbox $D = D(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})$ can be expressed by

$$\begin{aligned} D(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}) &\stackrel{\text{def}}{=} [\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}] \\ &= \{\mathbf{x}: x_i^{\text{low}} \leq x_i \leq x_i^{\text{up}}, i = 1, \dots, m\}, \end{aligned} \quad (4)$$

that is, the *lower bound* and *upper bound* of the hyperbox $D(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})$ are \mathbf{x}^{low} and \mathbf{x}^{up} , respectively. Moreover, let $\mathbf{x}^{\text{bound}}$ denote the *bounds* of the hyperbox, namely,

$$\mathbf{x}^{\text{bounddef}}(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}) = (x_1^{\text{low}}, \dots, x_m^{\text{low}}, x_1^{\text{up}}, \dots, x_m^{\text{up}}). \quad (5)$$

The volume of the hyperbox $D(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})$ is thus given by

$$v(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}) \stackrel{\text{def}}{=} \prod_{i=1}^m (x_i^{\text{up}} - x_i^{\text{low}}). \quad (6)$$

A hyperbox which only includes good design points is called a *solution hyperbox*. Simply, the hyperbox which meets the condition $f(\mathbf{x}) \geq f^c$ for all $\mathbf{x} \in D(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})$ is the solution hyperbox. Furthermore, $f_{\min}(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})$ is defined as the global minimization of the performance function on the hyperbox, that is, $f_{\min}(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}) \stackrel{\text{def}}{=} \min_{\mathbf{x} \in D(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})} f(\mathbf{x})$.

With these preparations at hand, the problem of searching for a maximum solution hyperbox can be formulated as the following constrained optimization problem:

$$\begin{aligned} & \max_{\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}} v(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}) \\ & \text{subject to } \min_{\mathbf{x} \in D(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})} f(\mathbf{x}) \geq f^c \\ & \mathbf{x}^l \leq \mathbf{x}^{\text{low}} \leq \mathbf{x}^{\text{up}} \leq \mathbf{x}^u. \end{aligned} \quad (7)$$

If $f(\mathbf{x})$ is analytically known and $\min_{\mathbf{x} \in [\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}]} f(\mathbf{x})$ (i.e., $f_{\min}(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})$) can be explicitly expressed by \mathbf{x}^{low} and \mathbf{x}^{up} , the optimization problem (7) can be solved by traditional optimization algorithms. Unfortunately, in most of the engineering problems, the following two cases are more common. First, the performance function $f(\mathbf{x})$ is not analytically known; therefore, it is produced by numerical simulations. Second, $\min_{\mathbf{x} \in [\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}]} f(\mathbf{x})$ cannot be explicitly expressed by \mathbf{x}^{low} and \mathbf{x}^{up} . To cope with these two cases, this paper aims to present a new approach based on performance function samples (i.e., the performance function can be analytically known or black box). There are two key difficulties for solving this optimization problem. First is how to maximize the size of the hyperbox. Second is how to guarantee the obtained hyperbox is a solution hyperbox.

Remark 1. If the performance criterion is $f^l \leq f(\mathbf{x}) \leq f^u$, the optimization problem formulation (7) sets $f(\mathbf{x}) \stackrel{\text{def}}{=} \min\{f(\mathbf{x}) - f^l, -f(\mathbf{x}) + f^u\}$ and $f^c \stackrel{\text{def}}{=} 0$.

Remark 2. Typically, engineering problems have multiple performance functions $f_i(\mathbf{x}) \geq f_i^c, i = 1, \dots, q$. In the optimization problem formulation (7), $\min_{\mathbf{x} \in D} f(\mathbf{x}) \geq f^c$ is replaced by $\min_{\mathbf{x} \in D} f_i(\mathbf{x}) \geq f_i^c, i = 1, \dots, q$.

3. The DIRECT Algorithm

We consider the global optimization problem of the form

$$\min_{\mathbf{x} \in D(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})} f(\mathbf{x}). \quad (8)$$

As mentioned in Section 2, $D(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}) = [\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}] = \{\mathbf{x}: x_i^{\text{low}} \leq x_i \leq x_i^{\text{up}}, i = 1, \dots, m\}$ is an m -dimensional hyperbox (hyper-rectangle), and $f(\mathbf{x})$ is a performance function.

In the literature, there are numerous methods for dealing with problem (8) (see, e.g., [18–27]). DIRECT (DIviding

RECTangles) is one of the most known partitioning-based algorithms that balance local and global search in an attempt to find the global minimizer efficiently [24–27]. The main procedure of the DIRECT algorithm involves partitioning potentially optimal (the most promising) hyper-rectangles and evaluating the performance function at the centers of these hyper-rectangles [28–31].

DIRECT's performance is independent of the scaling of D and the problem is typically scaled as follows:

$$\begin{aligned} \mathcal{T}(\mathbf{x}) &= A(\mathbf{x} - \mathbf{x}^{\text{low}}), \\ A &= \text{diag}\left(\frac{1}{x_1^{\text{up}} - x_1^{\text{low}}}, \dots, \frac{1}{x_m^{\text{up}} - x_m^{\text{low}}}\right). \end{aligned} \quad (9)$$

Thus, the bound-constrained problem (8) can be reduced to the following problem:

$$\min_{\mathbf{x} \in \tilde{D}} f(A^{-1}\mathbf{x} + \mathbf{x}^{\text{low}}), \quad (10)$$

over the unit hypercube $\tilde{D} = [0, 1]^m$. That is, the search space is transformed to the m -dimensional unit hypercube.

3.1. Selection Scheme. To select the potentially optimal hyper-rectangles, DIRECT assesses the goodness based on the lower bound estimates for the performance function over each hyper-rectangle. The requirement of potential optimality is stated formally in Definition 1.

Definition 1 (potentially optimal hyper-rectangle). Suppose that we have a partition of the unit hypercube into n hyper-rectangles. Let \mathbf{c}_i denote the center point of the i th hyper-rectangle, d_i denote the distance from the center point to the vertices, $\varepsilon > 0$ be a positive constant, and f_{\min} be the current best function value. A hyper-rectangle j is said to be potentially optimal if there exist some $\tilde{K} > 0$ such that

$$f(A^{-1}\mathbf{c}_j + \mathbf{x}^{\text{low}}) - \tilde{K}d_j \leq f(A^{-1}\mathbf{c}_i + \mathbf{x}^{\text{low}}) - \tilde{K}d_i, \quad (11)$$

for all $i = 1, \dots, n$,

$$f(A^{-1}\mathbf{c}_j + \mathbf{x}^{\text{low}}) - \tilde{K}d_j \leq f_{\min} - \varepsilon|f_{\min}|. \quad (12)$$

The hyper-rectangle j is potentially optimal if the lower Lipschitz bound for the performance function computed by the left-hand side of (11) is the smallest one with some positive constant \tilde{K} among all hyper-rectangles. In (12), the parameter ε is used to protect from an excessive refinement of the local minimum [24, 31].

DIRECT identifies potentially optimal hyper-rectangles in at least two different ways: using modified Graham's scan algorithm [32] or the rule described by Lemma 2.3 in [33]. Usually this does not impose significant difference; thus, in this paper, we use the modified Graham's scan algorithm.

3.2. Division and Sampling Scheme. We start by sampling the points $\mathbf{c} \pm \delta \mathbf{e}_i, i = 1, \dots, m$, where \mathbf{c} is the center point of the hypercube, δ is one-third the side length of the hypercube, and \mathbf{e}_i is the i th unit vector (i.e., a vector with a one in the i th

- (1) Identify the set I of dimensions with the maximum side length. Let δ equal one-third of this maximum side length.
- (2) Sample the function at the points $\mathbf{c} \pm \delta \mathbf{e}_i$ for all $i \in I$, where \mathbf{c} is the center of rectangle and \mathbf{e}_i is the i th unit vector.
- (3) Divide the rectangle containing \mathbf{c} into thirds along the dimensions in I , starting with the dimension with the lowest value of $\omega_i = \min\{f(A^{-1}(\mathbf{c} + \delta \mathbf{e}_i) + \mathbf{x}^{\text{low}}), f(A^{-1}(\mathbf{c} - \delta \mathbf{e}_i) + \mathbf{x}^{\text{low}})\}$, and continuing to the dimension with the next smallest ω_i until we have split on all dimensions $i \in I$.

ALGORITHM 1: Procedure for dividing rectangles.

- (0) Parameter setting: set the maximum number of iterations K_{\max} .
- (1) Normalize the search space to be the unit hypercube. Let \mathbf{c}_1 be the center point of this hypercube and evaluate $f(A^{-1}\mathbf{c}_1 + \mathbf{x}^{\text{low}})$. Set $f_{\min} = f(A^{-1}\mathbf{c}_1 + \mathbf{x}^{\text{low}})$, $n = 1$, and $k = 1$.
- (2) Identify the set S of potentially optimal rectangles.
- (3) Select any rectangle $j \in S$.
- (4) Use Algorithm 1 to determine where to sample within rectangle j and how to divide the rectangle into subrectangles. Update f_{\min} , set $n = n + \Delta n$, where Δn is the number of new points sampled.
- (5) Set $S = S - \{j\}$; If $S \neq \emptyset$ go to Step 3.
- (6) Set $k = k + 1$. If $k = K_{\max}$, stop and output the minimum value f_{\min} ; Otherwise, go to Step 2.

ALGORITHM 2: The DIRECT algorithm.

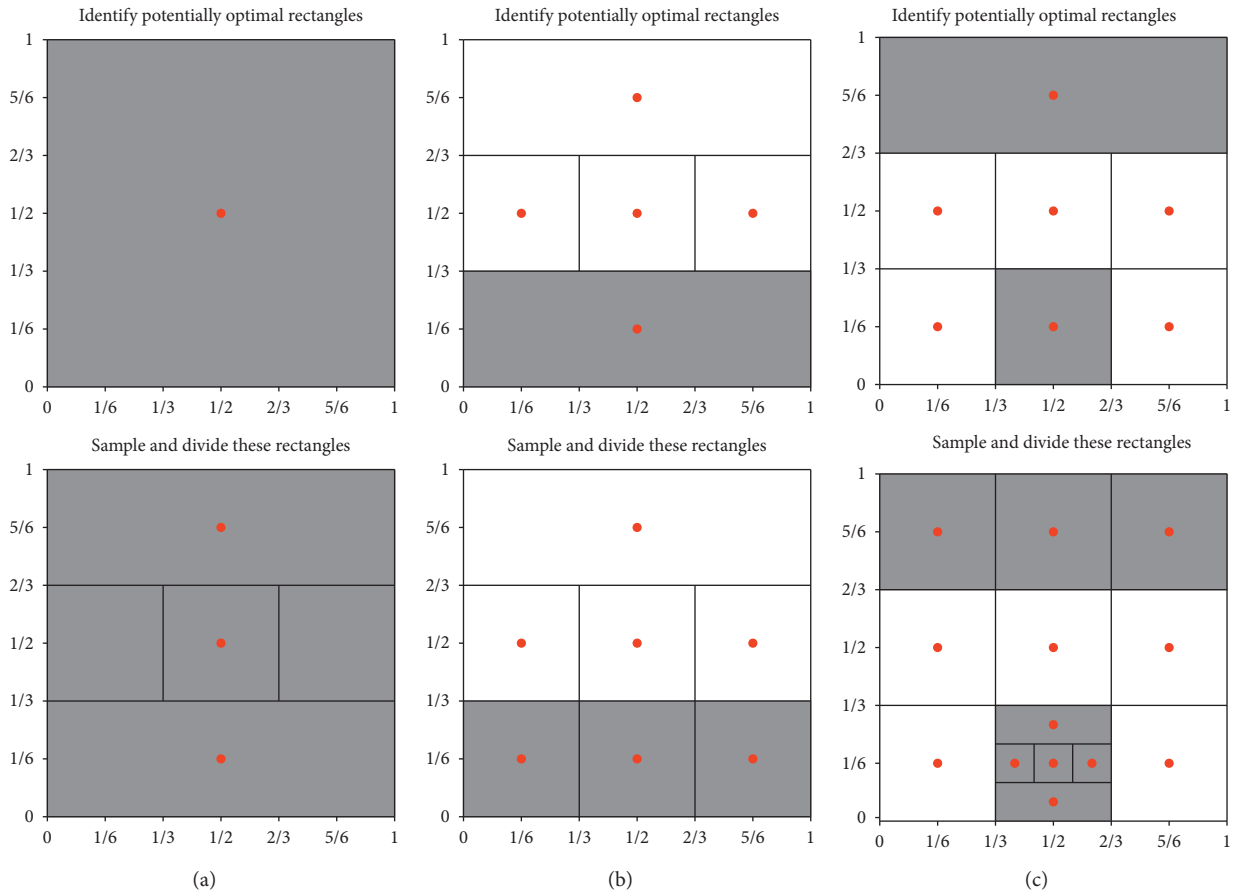


FIGURE 3: Three iterations of the DIRECT algorithm on Branin's function: gray colored hyper-rectangles highlight potentially optimal ones. They are selected in the first phase of the current iteration and divided in the second phase of the current iteration. Red points are sample points where the performance function is evaluated. (a) Iteration 1, (b) Iteration 2, and (c) Iteration 3.

position and zeros elsewhere). The strategy in [31] is used to divide the hypercube.

More precisely, we adopt the following rule for subdividing a hypercube. Let

$$\omega_i = \min\{f(A^{-1}(\mathbf{c} + \delta\mathbf{e}_i) + \mathbf{x}^{\text{low}}), f(A^{-1}(\mathbf{c} - \delta\mathbf{e}_i) + \mathbf{x}^{\text{low}})\} \quad (13)$$

be the best of the function values sampled along dimension i . Then, we start by splitting along the dimension with the smallest ω value. Once this is done, we split the rectangle containing \mathbf{c} into thirds along the dimension with the next smallest ω value. Continue in this way until we have split on all long dimensions. By dividing only along the long dimensions, we ensure that the rectangles shrink on every dimension. A formal description of the rectangle division procedure is given in Algorithm 1.

3.3. The DIRECT Algorithm Flowchart. We now have all the ingredients for the DIRECT algorithm. We initialize the search by sampling at the center of the unit hypercube. Each iteration begins with identifying the set of potentially optimal hyper-rectangles as described in Section 3.1. These hyper-rectangles are then sampled and subdivided as described in Section 3.2. The process continues until a pre-specified iteration limit is reached. A formal statement of the DIRECT algorithm is given in Algorithm 2.

In order to better illustrate how the DIRECT algorithm works, Figure 3 shows the first three iterations of the algorithm on the two-dimensional Branin function in [34]. For each iteration, the first row shows the set of potentially optimal rectangles (gray colored), and the second row shows how these rectangles are sampled (red points) and subdivided.

3.4. Convergence. The DIRECT algorithm is guaranteed to converge to the globally optimal function value if the performance function is continuous or at least continuous in the neighborhood of a global optimum [31]. This follows the fact that, as the number of iterations goes to infinity, the set of points sampled by DIRECT forms a dense subset of the unit hypercube. That is, given any point \mathbf{x} in the unit hypercube and any $\eta > 0$, DIRECT will eventually sample a point within the neighborhood of \mathbf{x} with radius η . It is proved that DIRECT with a large enough iteration number can converge with a high accuracy [31, 35–37].

4. The Proposed Approach

As metaheuristic algorithms are general and do not rely on mathematical properties for application and the DIRECT algorithm converges to the global optimum with a high accuracy, an innovative approach which combines metaheuristic algorithms with the DIRECT algorithm is proposed in this paper to tackle the two challenges mentioned in Section 2.2. First, the metaheuristic algorithm is used for seeking the largest hyperbox. Second, the DIRECT algorithm is used as a checking technique to

ensure all designs in the obtained hyperbox satisfy the performance criterion.

More specifically, in this paper, two methods are proposed: one combines simulated annealing with the DIRECT algorithm and the other combines the distributed covariance matrix adaptation evolution strategy with the DIRECT algorithm. They will be referred to as the simulated annealing DIRECT algorithm (SA-DIRECT) and the distributed covariance matrix adaptation evolution strategy DIRECT algorithm (d CMA-ES-DIRECT), respectively. Of course, one can use other metaheuristic algorithms, but these two metaheuristic algorithms are derivative-free and insensitive to initial solution and global search.

4.1. Simulated Annealing DIRECT Algorithm (SA-DIRECT). Simulated annealing (SA) proposed by Kirkpatrick et al. [38] is based on the similarity between the solid annealing process and solving global optimization problems. It is a generic probabilistic meta-algorithm for the global optimization problem. SA has been applied to a wide range of problems especially in those cases where traditional optimization techniques have shown poor performances or simply have failed [39].

Simulated annealing DIRECT (SA-DIRECT) algorithm is illustrated in Algorithm 3. Initially, the SA-DIRECT sets the design space and builds an initial solution for problem (7). More specifically, we generate a solution $(\mathbf{x}_0^{\text{low}}, \mathbf{x}_0^{\text{up}})$, where $\mathbf{x}_0^{\text{low}}$ and \mathbf{x}_0^{up} are stochastically generated from the range $[\mathbf{x}^l, \mathbf{x}^u]$. Then, we evaluate $\min_{\mathbf{x} \in D(\mathbf{x}_0^{\text{low}}, \mathbf{x}_0^{\text{up}})} f(\mathbf{x})$ by the DIRECT algorithm in order to check whether this generated solution meets constraints. If the constraint conditions, i.e., $\mathbf{x}^l \leq \mathbf{x}_0^{\text{low}} \leq \mathbf{x}_0^{\text{up}} \leq \mathbf{x}^u$ and $\min_{\mathbf{x} \in D(\mathbf{x}_0^{\text{low}}, \mathbf{x}_0^{\text{up}})} f(\mathbf{x}) \geq f^c$, are met, we calculate the associated value of the fitness function (i.e., the volume expressed by equation (6)). Otherwise, we discard this generated solution and generate a new initial solution.

Next, during the process of optimization, in each iteration, a neighbor solution to the current solution $(\mathbf{x}_i^{\text{low}}, \mathbf{x}_i^{\text{up}})$ is generated according to a predefined neighborhood structure, and $\min_{\mathbf{x} \in D(\mathbf{x}_i^{\text{low}}, \mathbf{x}_i^{\text{up}})} f(\mathbf{x})$ is evaluated by the DIRECT algorithm. If the condition $\min_{\mathbf{x} \in D(\mathbf{x}_i^{\text{low}}, \mathbf{x}_i^{\text{up}})} f(\mathbf{x}) \geq f^c$ is not satisfied, this neighbor solution is rejected. Otherwise, the volume $v(\mathbf{x}_i^{\text{low}}, \mathbf{x}_i^{\text{up}})$ is calculated by equation (6). The improving move (the volume of the neighbor is larger than that of the current solution) is always accepted, whilst worse neighbor is accepted with a certain probability determined by the Boltzmann probability [40], $\exp(\Delta v/T) > r$, where Δv is the difference between the volume of the current solution and the generated neighbor, and r is randomly generated from the range $[0, 1]$. Moreover, T is a parameter (called the temperature) which periodically lowers during the search process according to the temperature updating rule. The temperature updating rule (as adopted in [41]) is $T = \beta^t T_0$, where T_0 is the initial temperature, β is the cooling ratio, and t is the number of times the temperature has been lowered. The halting criterion of SA is “reaching the maximum number of times the temperature could be lowered.”

Finally, the SA-DIRECT verifies whether the best solution meets the constraints by the DIRECT algorithm with a

- (0) Parameter setting:
Set the initial temperature T_0 , the cooling ratio β , the maximum number of the inner iterations l_{\max} and the maximum number of the outer iterations t_{\max} .
- (1) Initialization:
- (1.1) Set the lower and upper bounds of design space, \mathbf{x}^l and \mathbf{x}^u , and the threshold level f^c .
 - (1.2) Generate randomly $\mathbf{x}_0^{\text{low}}$ and \mathbf{x}_0^{up} from the range $[\mathbf{x}^l, \mathbf{x}^u]$.
 - (1.3) Evaluate $f_{\min} = \min_{\mathbf{x} \in D(\mathbf{x}_0^{\text{low}}, \mathbf{x}_0^{\text{up}})} f(\mathbf{x})$ by Algorithm 2 with $K_{\max} = \lfloor 50 \ln 2 m \rfloor$.
 - (1.4) If all the constraints are satisfied, i.e., $\mathbf{x}^l \leq \mathbf{x}_0^{\text{low}} \leq \mathbf{x}_0^{\text{up}} \leq \mathbf{x}^u$ and $f_{\min} \geq f^c$, then calculate the volume by equation (6) and store it in v_0 , i.e., $v_0 = \nu(\mathbf{x}_0^{\text{low}}, \mathbf{x}_0^{\text{up}})$; Otherwise, go to Step 1.2.
 - (1.5) Set $\mathbf{x}^{\text{low}} = \mathbf{x}_c^{\text{low}} = \mathbf{x}_0^{\text{low}}$, $\mathbf{x}^{\text{up}} = \mathbf{x}_c^{\text{up}} = \mathbf{x}_0^{\text{up}}$, $v_b = v_c = v_0$, and $T = T_0$.
- (2) For $t = 1, \dots, t_{\max}$ **do**:
- (2.1) Set $l = 1$.
 - (2.2) While $l \leq l_{\max}$ **do**:
 - (2.2.1) Generate a new solution $(\mathbf{x}_l^{\text{low}}, \mathbf{x}_l^{\text{up}})$ in the neighborhood of $(\mathbf{x}_c^{\text{low}}, \mathbf{x}_c^{\text{up}})$, i.e., $\mathbf{x}_l^{\text{low}} = \mathbf{x}_c^{\text{low}} + (\mathbf{x}^u - \mathbf{x}^l) \odot (2\mathbf{r}^l - 1)^3$, $\mathbf{x}_l^{\text{up}} = \mathbf{x}_c^{\text{up}} + (\mathbf{x}^u - \mathbf{x}^l) \odot (2\mathbf{r}^l - 1)^3$, where \mathbf{r}^l and \mathbf{r}^u are randomly and uniformly generated from range $[0, 1]^m$, and \odot denotes componentwise multiplication.
 - (2.2.2) Evaluate $f_{\min} = \min_{\mathbf{x} \in D(\mathbf{x}_l^{\text{low}}, \mathbf{x}_l^{\text{up}})} f(\mathbf{x})$ by Algorithm 2 with the parameter setting $K_{\max} = \lfloor 100 \ln 2 m \rfloor$.
 - (2.2.3) If constraints $\mathbf{x}^l \leq \mathbf{x}_l^{\text{low}} \leq \mathbf{x}_l^{\text{up}} \leq \mathbf{x}^u$ and $f_{\min} \geq f^c$ are all met, then evaluate the volume v_l by equation (6), i.e., $v_l = \nu(\mathbf{x}_l^{\text{low}}, \mathbf{x}_l^{\text{up}})$; Otherwise go to Step 2.2.1.
 - (2.2.4) Evaluate $\Delta v = v_l - v_c$, if $\Delta v \geq 0$, then set $\mathbf{x}_c^{\text{low}} = \mathbf{x}_l^{\text{low}}$ and $\mathbf{x}_c^{\text{up}} = \mathbf{x}_l^{\text{up}}$; If $v_l - v_b > 0$, then set $\mathbf{x}^{\text{low}} = \mathbf{x}_l^{\text{low}}$, $\mathbf{x}^{\text{up}} = \mathbf{x}_l^{\text{up}}$.
 - (2.2.5) If $\Delta v < 0$, then generate a random number r in $[0, 1]$ and evaluate $P = \exp(\Delta v/T)$; If $P > r$, then set $\mathbf{x}_c^{\text{low}} = \mathbf{x}_l^{\text{low}}$, $\mathbf{x}_c^{\text{up}} = \mathbf{x}_l^{\text{up}}$, $v_c = v_l$.
 - (2.2.6) Set $l = l + 1$.
 - (2.3) Set $T = \beta T$.
- (3) Verification:
- (3.1) Evaluate $f_{\min} = \min_{\mathbf{x} \in D(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})} f(\mathbf{x})$ by Algorithm 2 with the parameter setting $K_{\max} = \lfloor 1000 \ln 2 m \rfloor$.
 - (3.2) If $f_{\min} \geq f^c$, output the best solution $(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})$; Otherwise go to Step 1.

ALGORITHM 3: The proposed simulated annealing DIRECT (SA-DIRECT) algorithm.

relatively large maximum number of iterations. If the constraints are not met, the SA-DIRECT goes back to Step 1. Otherwise, the SA-DIRECT outputs the solution.

In this paper, the maximum number of the iterations at each temperature $l_{\max} = 200$ and the maximum number of times the temperature could be lowered $t_{\max} = 1000$.

4.2. Distributed Covariance Matrix Adaptation Evolution Strategy DIRECT Algorithm (dCMA-ES-DIRECT). The evolution strategy (ES) is developed as a powerful tool for numerical optimization tasks [42]. Covariant matrix adaptation evolution strategy (CMA-ES) acts as an improved robust form of evolution strategy [43]. The main feature of the CMA-ES is the ability of being invariant to landscape transformations and scaling modulation. The CMA-ES is also invariant to applications of rotation, reflection, and translation, besides maintaining order and monotonicity [43]. It offers no discrepancy in behavior towards varied nature of functions and can be easily generalized. Complexity of algorithm is largely reduced with update schemes of CMA-ES, and thus it offers an extremely prospective mode of maximization in fitting function landscapes [44, 45]. The implementation details of CMA-ES are given in the appendix.

The CMA-ES is powerful and performs well. However, better results can be obtained by distributed covariant matrix adaptation evolution strategy (dCMA-ES) with multiple subpopulations and proper migration strategy [46].

Particularly, the population model of dCMA-ES divides the large population into multiple small demes. These demes evolve independently of each other for a certain number of generations, and then a number of individuals are migrated from one deme to another. The dCMA-ES preserves diversity in the population through multiple demes, while increasing the selection pressure through periodic migration [47].

The migration operator includes four parameters: (a) migration topology that defines the topology of the connections between demes, (b) the migration rate (the fraction of the population that migrates) that controls how many individuals migrate, (c) migration interval that affects the frequency of migrations, and (d) migration policy that selects emigrants and replaces existing individuals with incoming migrants [48].

In this paper, the ring topology is selected as the migration scheme [47]. In other words, individuals are transferred between directionally adjacent demes. The migration policy is that the best individuals (the larger volume value) are selected as migrants to replace the worst individuals at the receiving demes. The migration interval is set to 20 generations to permit the demes to partially converge prior to migration. The migration rate is set to 5 percent to provide a reasonable selection pressure. The final remaining experimental parameters directly related to the dCMA-ES are the deme size and deme count.

The distributed covariant matrix adaptation evolution strategy DIRECT (dCMA-ES-DIRECT) algorithm is illustrated

- (0) **Parameter setting:**
Set the default strategy parameters according to Appendix A.
- (1) **Initialization:**
- (1.1) Set the lower and upper bounds of the design space, \mathbf{x}^l and \mathbf{x}^u , the best volume $v_b = 0$, the maximum number of the outer iterations g_{\max} , and the threshold level f^c .
 - (1.2) Set the deme size d_s , the deme count d_c , the migration rate mig_r , the migration interval mig_i .
 - (1.3) Set the initial covariance matrix C_{0j} , the step-size σ_{0j} , the evolution paths $\mathbf{p}_{\sigma j}$ and $\mathbf{p}_{c j}$, $j = 1, \dots, d_c$ (details in the appendix).
 - (1.4) Generate the initial mean \mathbf{m}_{0j} uniformly and randomly from the range $[(\mathbf{x}^l, \mathbf{x}^l), (\mathbf{x}^u, \mathbf{x}^u)]$, $j = 1, \dots, d_c$.
- (2) For $g = 0, \dots, g_{\max}$ **do:**
- (2.1) For $j = 1, \dots, d_c$ **do:**
 - (2.1.1) For $i = 1, \dots, d_s$ **do:**
 - (2.1.1.1) Set $k = (j - 1) \times d_s + i$, then generate a random number $\mathbf{x}_k^{\text{bound}}$ from normal distribution $N(\mathbf{m}_{gj}, \sigma_{gj}^2 C_{gj})$. Based on equation (5), $\mathbf{x}_k^{\text{low}} = (\mathbf{x}_{k1}^{\text{bound}}, \dots, \mathbf{x}_{km}^{\text{bound}})$ and $\mathbf{x}_k^{\text{up}} = (\mathbf{x}_{k,m+1}^{\text{bound}}, \dots, \mathbf{x}_{k,2m}^{\text{bound}})$.
 - (2.1.1.2) Evaluate $f_{\min,k} = \min_{\mathbf{x} \in D(\mathbf{x}_k^{\text{low}}, \mathbf{x}_k^{\text{up}})} f(\mathbf{x})$ by Algorithm 2 with parameter setting $K_{\max} = \lfloor 100 \ln 2m \rfloor$.
 - (2.1.1.3) If all constraints are satisfied, i.e., $\mathbf{x}^l \leq \mathbf{x}_k^{\text{low}} \leq \mathbf{x}_k^{\text{up}} \leq \mathbf{x}^u$, and $f_{\min,k} \geq f^c$, then calculate the volume by equation (6), i.e., $v_k = v(\mathbf{x}_k^{\text{low}}, \mathbf{x}_k^{\text{up}})$; Otherwise go to Step 2.1.1.1.
 - (2.1.2) Set $k_1 = (j - 1) \times d_s + 1$ and $k_2 = j \times d_s$. Sort d_s individuals $\mathbf{x}_{k_1}^{\text{bound}}, \dots, \mathbf{x}_{k_2}^{\text{bound}}$ based on the value of the volume from largest to smallest. If $v_{s(1)} > v_b$, set $v_b = v_{s(1)}$, $\mathbf{x}^{\text{low}} = \mathbf{x}_{s(1)}^{\text{low}}$ and $\mathbf{x}^{\text{up}} = \mathbf{x}_{s(1)}^{\text{up}}$, where $s(i)$ denotes the index of i th ranked individual.
 - (2.2) **Migrate:**
 - (2.2.1) If $(g + 1)$ is divisible by mig_i **do:**
 - (2.2.1.1) For $j = 1, \dots, d_c$ **do:**
 - (2.2.1.1.1) Set $k_1 = (j - 1) \times d_s + 1$ and $k_2 = j \times d_s$. If $j = d_c$, then set $k_3 = 1$ and $k_4 = d_s$; Otherwise set $k_3 = j \times d_s + 1$ and $k_4 = (j + 1) \times d_s$.
 - (2.2.1.1.2) Select $\lfloor \text{mig}_r \times d_s \rfloor$ best individual out of $\mathbf{x}_{k_1}^{\text{bound}}, \dots, \mathbf{x}_{k_2}^{\text{bound}}$ to replace $\lfloor \text{mig}_r \times d_s \rfloor$ worst individual out of $\mathbf{x}_{k_3}^{\text{bound}}, \dots, \mathbf{x}_{k_4}^{\text{bound}}$.
 - (2.3) For $j = 1, \dots, d_c$ **do:**
 - (2.3.1) **Recombination:**
 - (2.3.1.1) Update $\mathbf{m}_{g+1,j}$, according to equation (A.3) in the appendix.
 - (2.3.2) **Step-size control**
 - (2.3.2.1) Update $\mathbf{p}_{\sigma,j}$, according to equation (A.4) in the appendix.
 - (2.3.2.2) Update $\sigma_{g+1,j}$, according to equation (A.5) in the appendix.
 - (2.3.3) **Covariance matrix adaptation**
 - (2.3.3.1) Update $\mathbf{p}_{c,j}$, according to equation (A.6) in the appendix.
 - (2.3.3.2) Update $C_{g+1,j}$, according to equation (A.7) in the appendix.
- (3) **Verification:**
- (3.1) Evaluate $f_{\min} = \min_{\mathbf{x} \in D(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})} f(\mathbf{x})$ by Algorithm 2 with the parameter setting $K_{\max} = \lfloor 1000 \ln 2m \rfloor$.
 - (3.2) If $f_{\min} \geq f^c$, output the individual $(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})$; Otherwise go to Step 1.

ALGORITHM 4: The proposed distributed covariant matrix adaptation evolution strategy DIRECT ($d\text{CMA-ES-DIRECT}$) algorithm.

in Algorithm 4. The $d\text{CMA-ES-DIRECT}$ begins with generating the initial mean, covariance matrix, and step size. Subsequently, the new individual is randomly generated by sampling from a multivariate normal distribution and is checked whether the constraints are satisfied by the DIRECT algorithm. If the constraints are not satisfied, resampling is performed until the generated individual becomes good. The fitness, i.e., the volume of the hyperbox in this paper, is calculated. The next step involves storing the individual with the maximum volume. Then, the migration operator is performed. Next, the update schemes by equations (A.2)–(A.6) in the appendix are applied to equip the individuals for the next generation.

Finally, the $d\text{CMA-ES-DIRECT}$ verifies whether the best individual meets the constraints by the DIRECT algorithm with a relatively large maximum number of iterations. If the constraints are not met, the $d\text{CMA-ES-DIRECT}$ goes back to Step 1. Otherwise, the $d\text{CMA-ES-DIRECT}$ outputs the best individual $(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})$.

4.3. Characteristics of the Proposed Approach. As these two metaheuristic algorithms are insensitive to initial solution, derivative-free and global search [49], the DIRECT algorithm with a large enough iteration number converges to the global optimum with a high accuracy [31]; the proposed methods which combine metaheuristic algorithms with the DIRECT algorithm have the following good characteristics:

- (1) The quality of the hyperbox obtained finally is not affected by the initial solution, except that the computational time may increase with improper starting designs.
- (2) The proposed methods have the advantage of not getting trapped in local optimum and have great possibility to reach the globally maximum solution hyperbox.
- (3) Because of the discrete nature of the performance function evaluations in the DIRECT algorithm, the

proposed methods can be used for both analytically known and black-box performance functions.

- (4) The proposed methods guarantee that any point selected within the hyperbox obtained finally satisfies the performance criterion as long as the performance function is continuous.

In the majority of practical engineering problems, the continuity of the performance function is easily satisfied, no matter it is analytically known or a black box. Therefore, the proposed approach has strong applicability and is valuable to practical engineering applications.

From Algorithms 3 and 4, we see that the DIRECT algorithm with a relatively small maximum number of iterations (i.e., $K_{\max} = \lfloor 100 \ln 2m \rfloor$) is adopted in the iterative optimization phase, while the DIRECT algorithm with a relatively large maximum number of iterations (i.e., $K_{\max} = \lfloor 1000 \ln 2m \rfloor$) is adopted in the verification phase. This setting reduces the computation time while ensuring that any point selected within the obtained hyperbox satisfies the performance criterion.

Due to the stochastic nature of metaheuristic algorithms, it is important to show the robustness of the proposed approaches. Therefore, as adopted in [15], the proposed approach runs 20 times, and the best solution among the 20 runs is used in practice. In addition, the boxplot containing the minimum, the maximum, the median, and the first and third quartiles of the 20 runs is shown.

5. Numerical Examples and Comparisons

A stochastic approach (we denote this method by ‘‘GHZ’’ hereafter) that computes the solution hyperbox has been discussed in [11]. An approach (we denote this method by ‘‘CES-IA’’ hereafter) based on the use of cellular evolutionary strategies (CES) and interval arithmetic (IA) has been proposed in [15].

In order to compare the proposed SA-DIRECT and d CMA-ES-DIRECT methods with the existing GHZ and CES-IA methods, the following two numerical examples are considered:

- (i) Example 1 studies multiple performance functions. Since each performance function is monotone, the exact solution of the optimization problem (7) exists and can be considered globally optimal.
- (ii) Example 2 studies a performance function which has multiple local minima. Unfortunately, the exact solution of the optimization problem (7) does not exist.

In addition, for comparison purpose, the solutions of the d CMA-ES-IA method which combines the distributed covariance matrix adaptation evolution strategy (d CMA-ES) with interval arithmetic (IA) is also shown.

5.1. Example 1: Multiple Performance Functions. Example 1 comes from [11], including multiple performance functions, where $\mathbf{x}^l = (0, 0)$ and $\mathbf{x}^u = (4, 4)$ are the lower and upper bounds of the design space, respectively, and

$$\begin{pmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ f_3(\mathbf{x}) \\ f_4(\mathbf{x}) \\ f_5(\mathbf{x}) \\ f_6(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} -\frac{1}{8}x_1 - \frac{1}{4}x_2 \\ -\frac{4}{17}x_1 - \frac{2}{17}x_2 \\ \frac{1}{2}x_1 - \frac{1}{2}x_2 \\ \frac{1}{2}x_1 + \frac{1}{3}x_2 \\ \frac{1}{3}x_1 + \frac{2}{3}x_2 \\ -x_1 + \frac{3}{2}x_2 \end{pmatrix}, \quad (14)$$

$$\begin{pmatrix} f_1^c \\ f_2^c \\ f_3^c \\ f_4^c \\ f_5^c \\ f_6^c \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \end{pmatrix}.$$

Since $f_i(\mathbf{x})$ is monotone, as the definition $f_{\min,i}(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}) \stackrel{\text{def}}{=} \min_{D(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})} f_i(\mathbf{x})$, we have

$$\begin{pmatrix} f_{\min,1}(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}) \\ f_{\min,2}(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}) \\ f_{\min,3}(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}) \\ f_{\min,4}(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}) \\ f_{\min,5}(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}) \\ f_{\min,6}(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}) \end{pmatrix} = \begin{pmatrix} -\frac{1}{8}x_1^{\text{up}} - \frac{1}{4}x_2^{\text{up}} \\ -\frac{4}{17}x_1^{\text{up}} - \frac{2}{17}x_2^{\text{up}} \\ \frac{1}{2}x_1^{\text{low}} - \frac{1}{2}x_2^{\text{up}} \\ \frac{1}{2}x_1^{\text{low}} + \frac{1}{3}x_2^{\text{low}} \\ \frac{1}{3}x_1^{\text{low}} + \frac{2}{3}x_2^{\text{low}} \\ -x_1^{\text{up}} + \frac{3}{2}x_2^{\text{low}} \end{pmatrix}. \quad (15)$$

That is, $f_{\min,i}(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})$ has an explicit expression, for $i = 1, \dots, 6$.

Therefore, the optimization problem (7) can be restated as follows:

$$\begin{aligned} \max_{\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}} \quad & \prod_{i=1}^2 (x_i^{\text{up}} - x_i^{\text{low}}) \\ \text{subject to} \quad & f_{\min,i}(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}}) \geq f_i^c, \quad i = 1, \dots, 6. \end{aligned} \quad (16)$$

Obviously, (16) is an optimization problem with inequality constraints. The exact solution can be found by the Lagrange multiplier method [50] and has the value tabulated in the second column of Table 1.

The result in [11] obtained by the GHZ method is listed in the third column of Table 1. The CES-IA method adopts the same parameter setting as in [15] and the best solution among its 20 runs is shown in the fourth column of Table 1. The best solution among 20 runs of the d CMA-ES-IA method is listed in the fifth column of Table 1. The row entitled ‘‘Volume’’ contains the volume of the obtained hyperbox, and the row entitled ‘‘Error’’ contains the relative error between the volume and the exact volume.

In the proposed SA-DIRECT method, the initial temperature T_0 is set to 100, 150, and 200, and the cooling ratio β is set to 0.94 and 0.98. In the proposed d CMA-ES-DIRECT method, two sets of deme size and count values are chosen: 8 and 1 and 8 and 3, respectively. The best solutions among their respective 20 runs of the SA-DIRECT and CMA-ES-DIRECT methods are presented in Table 1. A visualization of results of these methods in Table 1 is shown in Figure 4. Moreover, we magnify the region A in Figure 4 and show it in Figure 5.

According to Table 1, the volumes of the obtained hyperboxes by the CES-IA, d CMA-ES-IA, SA-DIRECT, and d CMA-ES-DIRECT methods are approximately the same as the exact value, whereas the volume of the obtained hyperbox by the GHZ method has a somewhat large error.

Moreover, we can see that the SA-DIRECT method is insensitive to parameter variations of simulated annealing.

From Figure 4, we observe that the obtained hyperboxes by the SA-DIRECT and d CMA-ES-DIRECT methods are in close proximity to the exact solution. From Figure 5, it can be seen that the green lines exceed the gray region, which means that the hyperbox obtained by the GHZ method contains some design points which are not within the complete solution space, and therefore is not a solution hyperbox. However, the hyperboxes obtained by the CES-IA, d CMA-ES-IA, SA-DIRECT, and d CMA-ES-DIRECT methods all locate within the complete solution space and therefore are solution hyperboxes.

Consequently, the hyperboxes obtained by the CES-IA, d CMA-ES-IA, SA-DIRECT, and d CMA-ES-DIRECT methods not only are close to the exact hyperbox but also guarantee that any point selected within those hyperboxes satisfies the performance criterion.

Furthermore, we sketch the boxplots using the volume of the hyperbox of 20 runs, which are presented in Figure 6. From Figure 6, we observe that, in each case of parameter setting, the volumes of the obtained hyperboxes are all above 2.2000 after eliminating outliers, and the range of the volumes is relatively small; therefore, the two proposed approaches are robust.

5.2. Numerical Example 2: Multiple Local Minima. The maximization of the solution hyperbox is considered in case of the Michalewics function:

$$f(\mathbf{x}) = - \sum_{i=1}^2 \sin(x_i) \left(\sin\left(\frac{ix_i^2}{\pi}\right) \right)^{20}. \quad (17)$$

Stripinis et al. [51] have pointed out that the Michalewics function has two local minima and one global minimum. In this example, the lower and upper bounds of design space are $\mathbf{x}^l = (1, 1)$ and $\mathbf{x}^u = (\pi, \pi)$, respectively, and the threshold value is $f^c = -1.5$.

Unfortunately, the bound-constrained optimization problem $\min_{D(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})} f(\mathbf{x})$ has no explicit expressions; therefore, we cannot obtain the exact solution for the optimization problem (7) in this example. However, $\min_{D(\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})} f(\mathbf{x})$ can be solved by the FMINCON function of MATLAB. For comparison purpose, an approach (we denote this method by ‘‘SA-FMINCON’’ hereafter) which combines simulated annealing and FMINCON function is presented to solve problem (7).

The best solutions of their respective 20 runs of the GHZ, CES-IA, d CMA-ES-IA, SA-DIRECT, d CMA-ES-DIRECT, and SA-FMINCON methods are shown in Table 2. A visualization of these results is shown in Figure 7. According to Table 2 and Figure 7, the hyperbox obtained by the SA-FMINCON method is the maximum and the hyperboxes obtained by the SA-DIRECT and d CMA-ES-DIRECT methods are much larger than those by the GHZ, CES-IA, and d CMA-ES-IA methods. Besides, we see that the SA-DIRECT method is relatively insensitive to the initial parameters of simulated annealing.

TABLE 1: The exact, GHZ, CES-IA, *d*CMA-ES-IA, SA-DIRECT, and *d*CMA-ES-DIRECT solutions of Example 1.

	Exact	GHZ	CES-IA	<i>d</i> CMA-ES-IA	SA-DIRECT				<i>d</i> CMA-ES-DIRECT	
					$T_0 = 100$ $\beta = 0.94$	$T_0 = 150$ $\beta = 0.94$	$T_0 = 150$ $\beta = 0.98$	$T_0 = 200$ $\beta = 0.98$	$d_s = 8$ $d_c = 1$	$d_s = 8$ $d_c = 3$
x_1^{low}	1.1800	1.1740	1.1804	1.1826	1.1805	1.1876	1.1744	1.1957	1.1790	1.1799
x_1^{up}	2.8500	2.7350	2.8196	2.8392	2.8432	2.8298	2.8547	2.8111	2.8473	2.8453
x_2^{low}	1.2300	1.2350	1.2298	1.2262	1.2294	1.2206	1.2384	1.2081	1.2316	1.2302
x_2^{up}	2.5800	2.6330	2.5902	2.5802	2.5770	2.5850	2.5727	2.5943	2.5763	2.5773
Volume	2.2545	2.1823	2.2396	2.2432	2.2408	2.2406	2.2419	2.2398	2.2436	2.2436
Error (%)	—	3.20	0.66	0.50	0.60	0.62	0.56	0.65	0.48	0.48

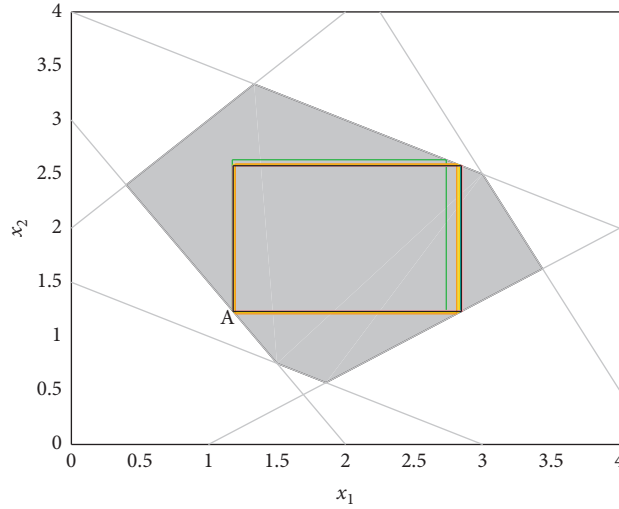


FIGURE 4: Example 1: the gray region is the complete solution space. The regions are obtained by the following methods: the exact (red), GHZ (green), CES-IA (yellow), *d*CMA-ES-IA (cyan), SA-DIRECT with $T_0 = 100$ and $\beta = 0.94$ (blue), SA-DIRECT with $T_0 = 150$ and $\beta = 0.94$ (gold), SA-DIRECT with $T_0 = 150$ and $\beta = 0.98$ (pink), SA-DIRECT with $T_0 = 200$ and $\beta = 0.98$ (orange), *d*CMA-ES-DIRECT with $d_s = 8$ and $d_c = 1$ (magenta), and *d*CMA-ES-DIRECT with $d_s = 8$ and $d_c = 3$ (black).

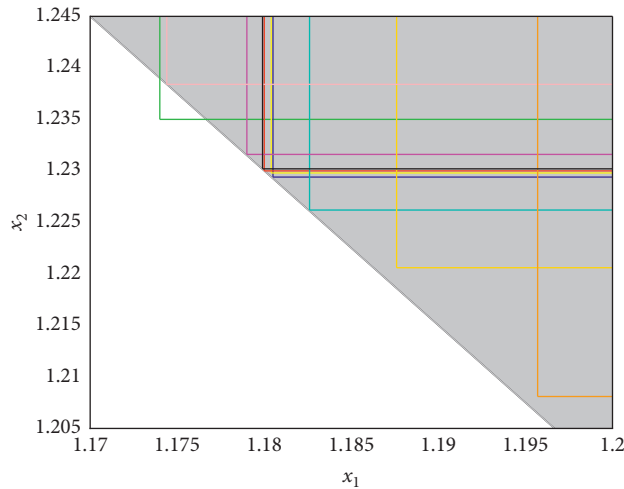


FIGURE 5: Magnification of region A in Figure 4.

The values of the performance function $f(\mathbf{x})$ on the obtained hyperboxes by the GHZ, CES-IA, *d*CMA-ES-IA, SA-DIRECT, *d*CMA-ES-DIRECT, and SA-FMINCON methods are shown in Figure 8. We can see that the values of

the performance function $f(\mathbf{x})$ on the hyperboxes by the CES-IA, *d*CMA-ES-IA, SA-DIRECT, and *d*CMA-ES-DIRECT methods are all above the performance criterion while some values of the performance function $f(\mathbf{x})$ on those by

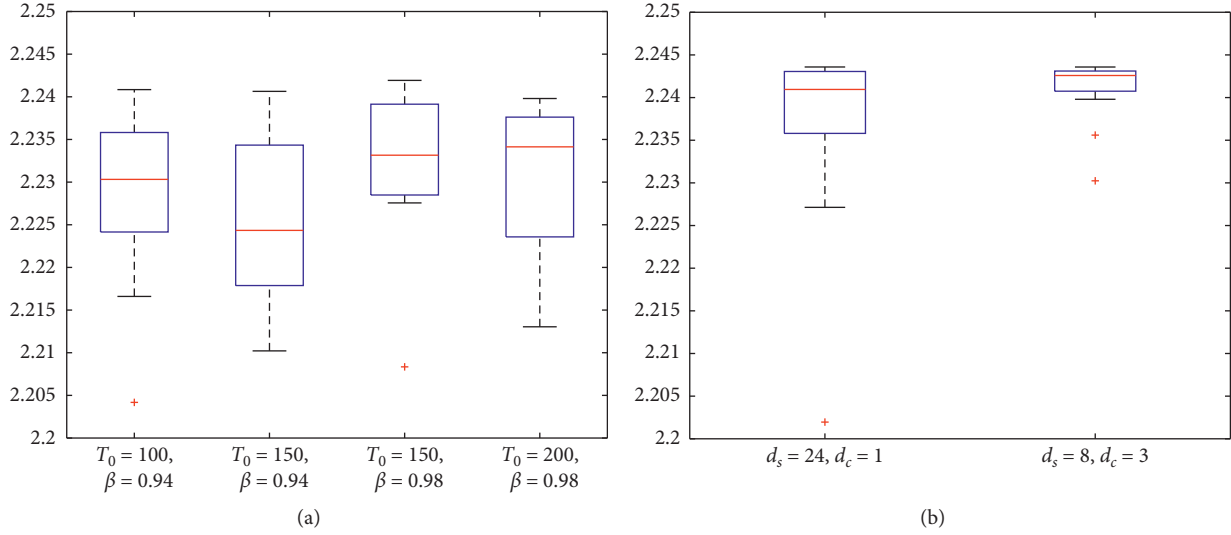


FIGURE 6: Example 1: the boxplots of the volume of the obtained hyperbox. (a) SA-DIRECT method and (b) d CMA-ES-DIRECT method.

TABLE 2: The best GHZ, CES-IA, d CMA-ES-IA, SA-DIRECT, d CMA-ES-DIRECT, and SA-FMINCON solutions of Example 2.

	GHZ	CES-IA	d CMA-ES-IA	SA-DIRECT				d CMA-ES-DIRECT		SA-FMINCON
				$T_0 = 100$ $\beta = 0.94$	$T_0 = 150$ $\beta = 0.94$	$T_0 = 150$ $\beta = 0.98$	$T_0 = 200$ $\beta = 0.98$	$d_s = 8$ $d_c = 1$	$d_s = 8$ $d_c = 3$	
x_1^{low}	1.5095	1.2388	1.0001	1.0002	1.0007	1.0007	1.0007	1.0000	1.0000	1.0009
x_1^{up}	2.6406	2.2659	1.9619	3.1403	3.1406	3.1414	3.1410	3.1416	3.1415	3.1412
x_2^{low}	1.6520	1.8628	1.8756	1.6622	1.6625	1.6633	1.6625	1.6620	1.6620	1.0037
x_2^{up}	3.1415	2.5521	2.5673	3.1410	3.1414	3.1414	3.1408	3.1416	3.1416	3.1415
Volume	1.6849	0.7080	0.7426	3.1649	3.1648	3.1647	3.1640	3.1687	3.1685	4.5755

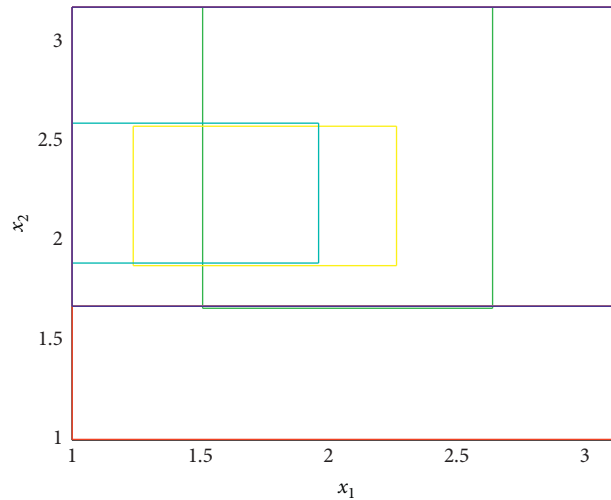


FIGURE 7: Example 2: the regions obtained by the following methods: the GHZ (green), CES-IA (yellow), d CMA-ES-IA (cyan), SA-DIRECT with $T_0 = 100$ and $\beta = 0.94$ (blue), d CMA-ES-DIRECT with $d_s = 8$ and $d_c = 1$ (magenta), and SA-FMINCON (red).

the GHZ and SA-FMINCON methods are below the performance criterion. This indicates that the obtained hyperboxes by the CES-IA, d CMA-ES-IA, SA-DIRECT, and d CMA-ES-DIRECT methods only include good design

points while those by the GHZ and SA-FMINCON methods include some bad design points.

The boxplots of the volume of the hyperbox of 20 runs are given in Figure 9. From Figure 9, we see that, in each case

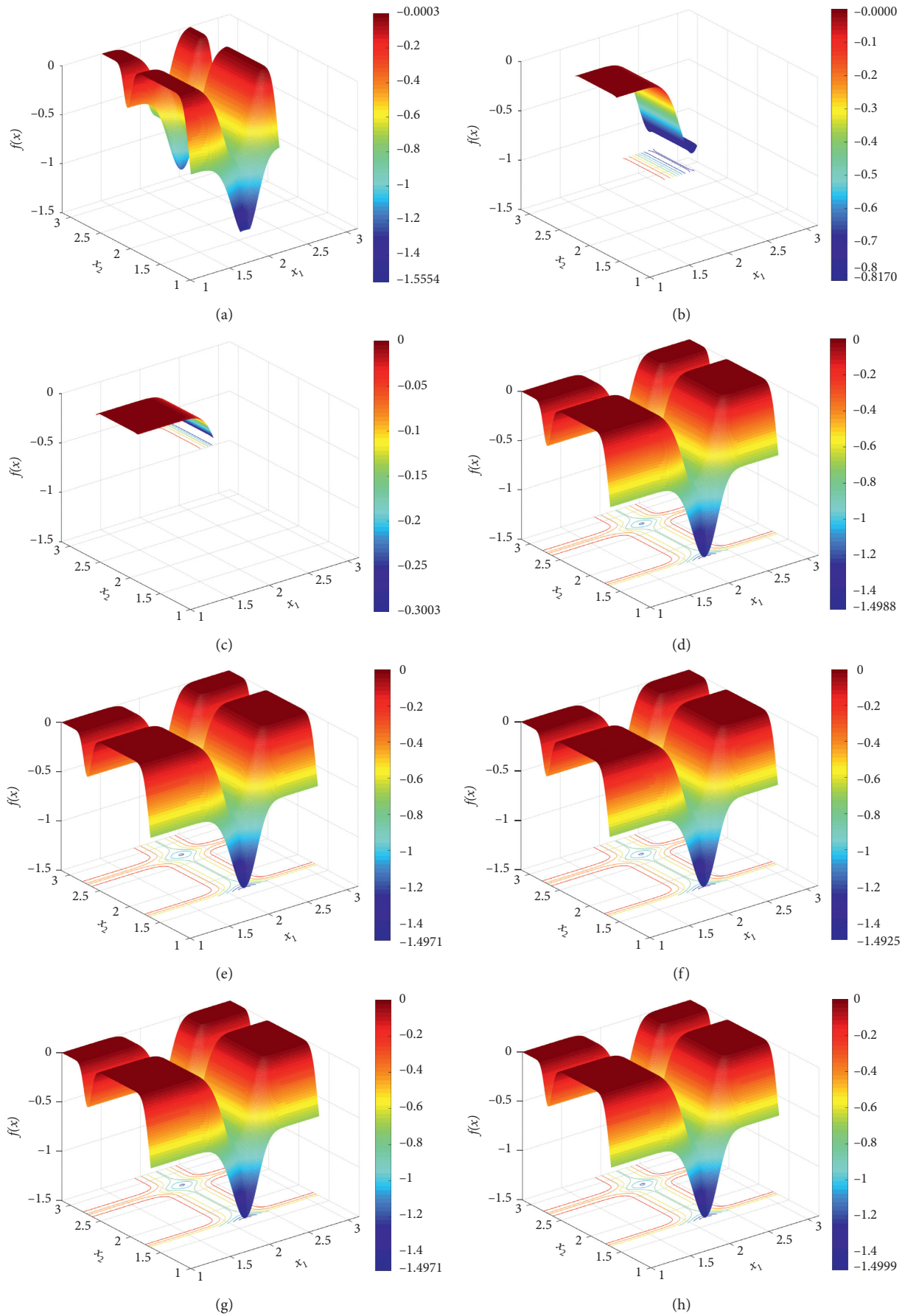


FIGURE 8: Continued.

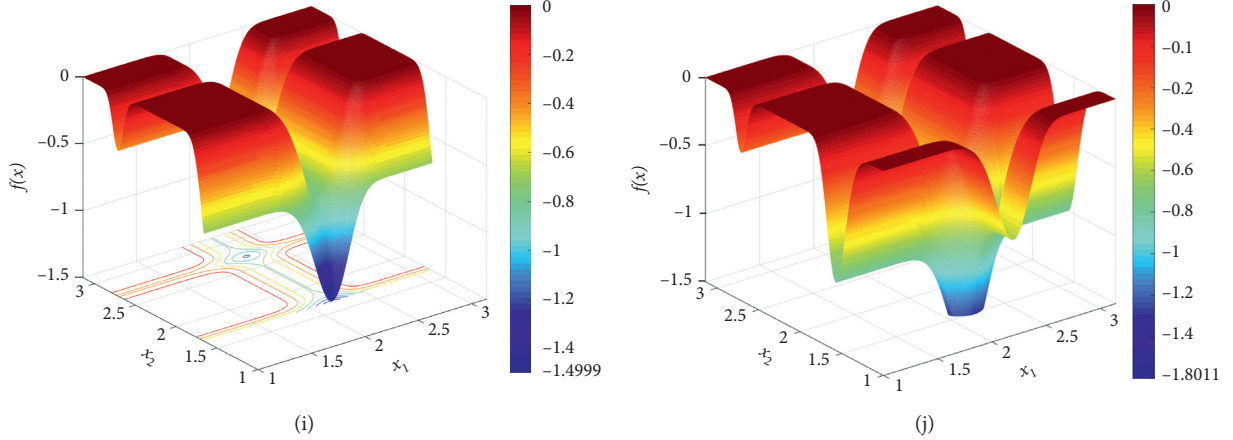


FIGURE 8: The values of the performance function $f(\mathbf{x})$ on the hyperboxes obtained by the following methods: (a) GHZ, (b) CES-IA, (c) d CMA-ES-IA, (d) SA-DIRECT with $T_0=100, \beta=0.94$, (e) SA-DIRECT with $T_0=150, \beta=0.94$, (f) SA-DIRECT with $T_0=150, \beta=0.98$, (g) SA-DIRECT with $T_0=200, \beta=0.98$, (h) d CMA-ES-DIRECT with $d_s=8, d_c=1$, (i) d CMA-ES-DIRECT with $d_s=8, d_c=3$, and (j) SA-fmincon.

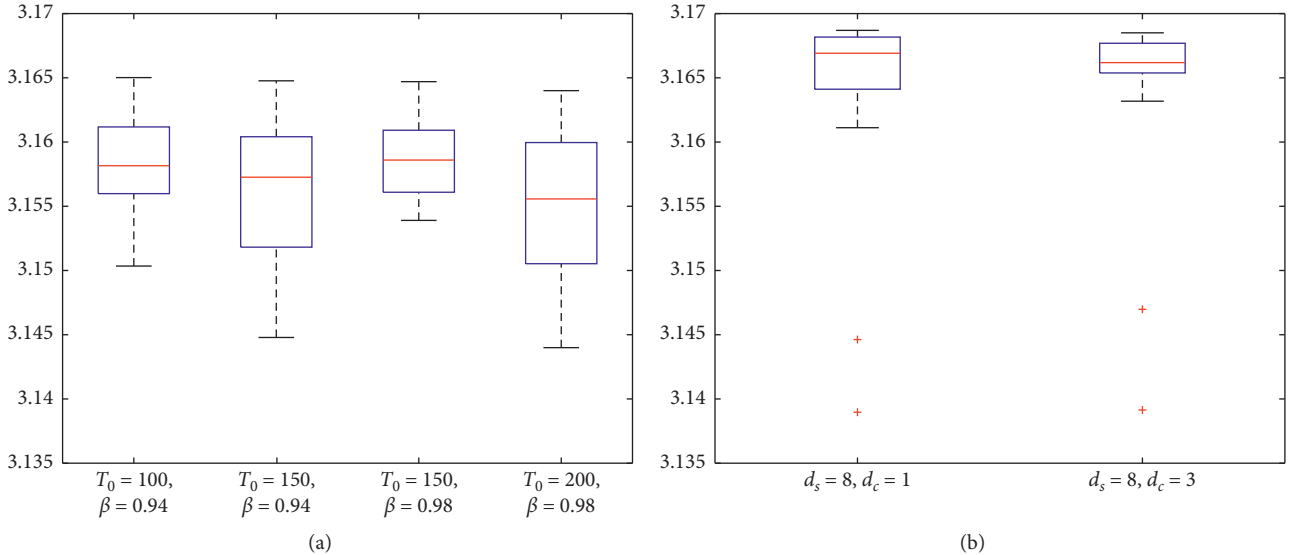


FIGURE 9: Example 2: the boxplots of the volume of the obtained hyperbox. (a) SA-DIRECT method and (b) d CMA-ES-DIRECT method.

of parameter setting, the volumes of the obtained hyperboxes are all above 3.1400 after eliminating outliers, and the range of the volumes is relatively small; therefore, the robustness of the proposed approaches are verified.

5.3. Discussion. The CES-IA, d CMA-ES-IA, and SA-FMINCON methods require an analytically known performance function, while GHZ, SA-DIRECT, and d CMA-ES-DIRECT methods can be used for both analytically known and black-box performance functions.

The hyperboxes obtained by the CES-IA and d CMA-ES-IA methods in Example 1 are very close to those obtained by the proposed SA-DIRECT and d CMA-ES-DIRECT methods while the hyperboxes obtained by the CES-IA and d CMA-ES-IA methods in Example 2 are much smaller than those obtained by the proposed SA-DIRECT and d CMA-ES-

DIRECT methods. This is mainly caused by the dependency problem of the interval arithmetic. In the CES-IA and d CMA-ES-IA methods, the range of the performance function inside the generated hyperbox is calculated by the interval arithmetic. In Example 1, each design parameter appears only once in the performance function $f_i(\mathbf{x})$; thus, the interval arithmetic can determine the range of the performance function very accurately. However, in Example 2, the performance function $f(\mathbf{x})$ is unable to be described by the output interval directly, instead the Taylor interval extension [52] is used. In this case, the design parameters occur several times in the calculation, and each occurrence is taken independently. Consequently, the range of the performance function inside the generated hyperbox is overestimated and the actually solution hyperbox is discarded. Rocco et al. [15] have pointed out that the main drawback of the interval arithmetic adopted in the CES-IA method is that

results can be overestimated due to the dependency problem, and overestimation cannot be estimated and could cause that no solution hyperbox is found.

Numerical examples show that the hyperboxes obtained by the GHZ method may include some bad design points. This is mainly because the GHZ method uses Monte Carlo sampling to estimate locations of good and bad designs. Monte Carlo sampling cannot guarantee that each design point in the obtained hyperbox is good for even a large number of samples, especially when the performance function has multiple local minima. In addition, numerical examples show that the hyperboxes by the GHZ method are smaller than those by the proposed SA-DIRECT and d CMA-ES-DIRECT methods. This is because the GHZ method removes bad sample points sequentially in only one order. The order, however, has an effect on the result. Zimmermann and von Hoessle [10] have concluded that the GHZ method is not necessarily globally optimal, even not locally optimal.

In Example 2, although the hyperbox obtained by the SA-FMINCON method is larger than those by the proposed SA-DIRECT and d CMA-ES-DIRECT methods, it includes some bad design points, and therefore is not a solution hyperbox. This demonstrates that it is improper to use the FMINCON function of MATLAB to solve the optimization problem (7) especially when the performance function has multiple local minima.

From the “quantitative” viewpoint, the solution hyperboxes obtained by the proposed SA-DIRECT and d CMA-ES-DIRECT methods are nearly the same as the exact result in Example 1, and the solution hyperboxes obtained by the SA-DIRECT and d CMA-ES-DIRECT methods are much larger than those obtained by the CES-IA and GHZ methods in Example 2 without exact result. From the “feasible” viewpoint, no matter Example 1 or Example 2, any design point selected within the solution hyperboxes obtained by the SA-DIRECT and d CMA-ES-DIRECT methods satisfies the performance criterion. Therefore, numerical examples prove that the proposed global search is effective.

Moreover, from Figures 6 and 9, we observe that the SA-DIRECT and d CMA-ES-DIRECT methods are robust. As the SA-DIRECT method with $T_0 = 150$ and $\beta = 0.98$ has a relatively large maximum value and a relatively small interquartile range, we therefore set $T_0 = 150$ and $\beta = 0.98$ in the remainder of paper. Similarly, the d -CMA-ES-DIRECT method with $d_s = 3 + \lfloor 3 + \ln 2m \rfloor$ (i.e., $d_s = 8$ in Examples 1 and 2) and $d_c = 3$ has a relatively large maximum value and a relatively small interquartile range; we therefore set $d_s = 3 + \lfloor 3 + \ln 2m \rfloor$ and $d_c = 3$ in the rest of the paper.

6. Case Studies

The first case is the life-support system in a space capsule which has been studied by Rocco et al. [15] and is reinvestigated by the proposed approach. The second case is the power-shift steering transmission control system (PSSTCS) which is a typical mechatronics complex control system with a price of approximately 500,000 USD. Therefore, it is

importantly significant to improve the robustness of the PSSTCS.

6.1. Life-Support System in a Space Capsule

6.1.1. Life-Support System in a Space Capsule: Reliability Constraint. The complex system (life-support system in a space capsule) in [15], as shown in Figure 10, consists of four components. The reliability of the system is given by

$$R_s = f(\mathbf{R}) = 1 - R_3 (\bar{R}_1 \bar{R}_4)^2 - \bar{R}_3 [1 - R_2 (1 - \bar{R}_1 \bar{R}_4)]^2, \quad (18)$$

where $\mathbf{R} = (R_1, R_2, R_3, R_4)$ and $\bar{R}_i = 1 - R_i$ for $i = 1, \dots, 4$.

Rocco et al. [15] have applied the CES-IA method to find a largest symmetric hyperbox using a specified point as symmetry center, whereby any point selected within this hyperbox satisfies the performance criterion. Particularly, the center point is $\mathbf{R}^c = (0.90, 0.90, 0.90, 0.90)$, the lower and upper bounds of the design space are $\mathbf{R}^l = \mathbf{0}$ and $\mathbf{R}^u = \mathbf{1}$, respectively, and the performance criterion is $R_s \geq 0.99$. The best solution among the 20 runs of the CES-IA method shown in [15] is given in the fourth column of Table 3 in terms of the intervals of design parameters.

For comparison purpose, the optimization problem considered in this paper is the same as that considered in [15], and it is stated similarly to the optimization problem (7):

$$\begin{aligned} \max_{\mathbf{R}^{\text{up}}} \quad & \prod_{i=1}^4 2(R_i^{\text{up}} - R_i^c), \\ \text{subject to} \quad & \min_{\mathbf{R} \in [\mathbf{R}^{\text{low}}, \mathbf{R}^{\text{up}}]} f(\mathbf{R}) \geq 0.99, \\ & \mathbf{R}^c \leq \mathbf{R}^{\text{up}} \leq \mathbf{R}^u, \end{aligned} \quad (19)$$

where $\mathbf{R}^{\text{low}} = 2\mathbf{R}^c - \mathbf{R}^{\text{up}}$.

The best solutions among the 20 runs of the GHZ, SA-DIRECT, and d CMA-ES-DIRECT methods are shown in the third, fifth, and sixth columns of Table 3, respectively. To show how close to the global solution are the results obtained by the proposed methods, the second column of Table 3 presents the exact solution obtained by the Lagrange multiplier method. To show whether the obtained hyperboxes satisfy the performance criterion, the intervals of the performance function R_s on the hyperboxes obtained by the exact, GHZ, CES-IA, SA-DIRECT, and d CMA-ES-DIRECT methods are, respectively, calculated and listed in Table 4.

Based on Tables 3 and 4, the SA-DIRECT and d CMA-ES-DIRECT methods outperform the GHZ and CES-IA methods in which the volume errors are smaller and the minimum values of the system reliability on their obtained hyperboxes are above the reliability criterion (0.99). Although the hyperbox obtained by the GHZ method is maximum, from the third column of Table 4, we can see that it includes some design points whose corresponding reliabilities are below the reliability criterion. Although the hyperbox obtained by the CES-IA method only includes good design points, its volume is smaller than those by the SA-DIRECT and d CMA-ES-DIRECT methods.

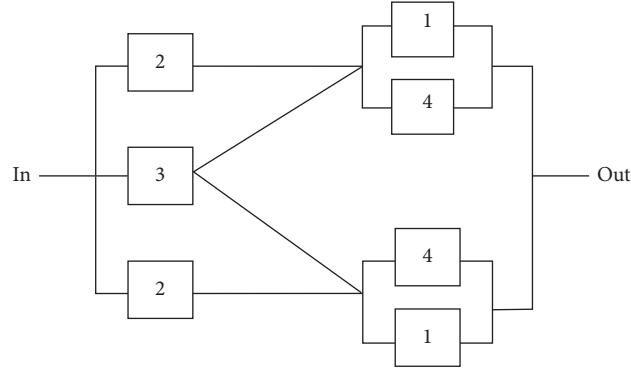


FIGURE 10: Example of the reliability system.

Besides, the average volumes are, respectively, 1.2111×10^{-3} and 1.2115×10^{-3} in 20 runs of the SA-DIRECT and *d*CMA-ES-DIRECT methods, and the standard deviations are, respectively, 1.2184×10^{-6} and 1.2745×10^{-6} . This implies that the proposed methods are robust.

6.1.2. Life-Support System in a Space Capsule: Additional Cost Constraint. Normally, the design problem also seeks to minimize the cost function C_s :

$$C_s = 2 \sum_{i=1}^4 K_i R_i^{\alpha_i}. \quad (20)$$

In [15], $K_1 = 100$, $K_2 = 100$, $K_3 = 100$, $K_4 = 150$, and $\alpha_i = 1$ for all i . Using the above values and the previous SA-DIRECT ranges of R_i , the corresponding range of C_s is [724.8847, 895.1153]. Using the exact results of R_i , the corresponding range of C_s is [724.8300, 895.1000].

Rocco et al. [15] studied the problem to derive the ranges for each R_i subject to $R_s \geq 0.99$ and $750 \leq C_s \leq 850$. We solve the same problem by the proposed methods for comparison. The best solution shown in [15] by the CES-IA method is given in the fourth column of Table 5.

Since the cost function C_s is monotone, the exact solution can be obtained by the Lagrange multiplier method and is shown in the second column of Table 5. The best solutions among the 20 runs of the GHZ, SA-DIRECT, and *d*CMA-ES-DIRECT methods are also shown in the third, fifth, and sixth columns of Table 5, respectively. The intervals of R_s and C_s on the obtained hyperbox, denoted by $R_{s,\text{bounded}}$ and $C_{s,\text{bounded}}$, respectively, are shown in Table 6. From Tables 5 and 6, we can see that the proposed methods are better than the GHZ and CES-IA methods because the volume errors are smaller and the performance criterion is satisfied in the obtained hyperboxes.

The relative errors between the exact solution and the best solutions of the proposed SA-DIRECT and *d*CMA-ES-DIRECT methods are both 0.02. This indicates that the results are globally optimal.

6.2. The Power-Shift Steering Transmission Control System (PSSTCS). The power-shift steering transmission control system (PSSTCS) is a key complex system with

multicharacteristics of heavy vehicle to achieve the control of the steering, speed changing, fan driving, and lubricating. The PSSTCS is composed of a hydraulic oil supply system, an integration pump-motor system, a fan control system, an electronic control system, and a hydraulic control system. The hydraulic oil supply system consists of a fill oil and constant pressure system of pressure oil tank, a pump group, and a fill oil system of transmission control and fan control. The function constitutes and the structure principle drawing of PSSTCS are shown in Figures 11 and 12, respectively. The PSSTCS is a nonmonotonic coherent system; therefore, its reliability function is not analytically known and is evaluated by a graphical inductive analysis method based on the principles of the decision tree, i.e., the goal-oriented (GO) reliability assessment method, as illustrated in the appendix.

As shown in Table 7, the number of design parameters is 86. The optimization problem (7) is given as follows:

$$\begin{aligned} & \max_{\mathbf{R}^{\text{up}}, \mathbf{R}^{\text{low}}} \sum_{i=1}^{86} \log(R_i^{\text{up}} - R_i^{\text{low}}) \\ & \text{subject to } \min_{\mathbf{R} \in [\mathbf{R}^{\text{low}}, \mathbf{R}^{\text{up}}]} f(\mathbf{R}) \geq f^c \\ & \mathbf{R}^{\text{l}} \leq \mathbf{R}^{\text{low}} \leq \mathbf{R}^{\text{up}} \leq \mathbf{R}^{\text{u}}, \end{aligned} \quad (21)$$

where $\mathbf{R}^{\text{l}} = \mathbf{0.9990}$, $\mathbf{R}^{\text{u}} = \mathbf{1.0000}$, $f^c = 0.9900$, and f is the reliability function and is evaluated by the goal-oriented (GO) reliability assessment method. Note that the volume here is replaced by the log-volume (logarithmic transformation of the volume) in order to calculate conveniently.

Since the reliability function of the PSSTCS is not analytically known, the CES-IA method is not suitable for this complex system. The best solutions among the 20 runs of the GHZ, SA-DIRECT, and *d*CMA-ES-DIRECT methods are shown in the third, fourth, and fifth columns of Table 7, respectively. The log-volumes of the GHZ, SA-DIRECT, and *d*CMA-ES-DIRECT hyperboxes are listed in Table 8. We see that the log-volumes of the SA-DIRECT and *d*CMA-ES-DIRECT hyperboxes are all much larger than that of the GHZ hyperbox.

To show whether the obtained hyperboxes satisfy the performance criterion, N sample design points are randomly chosen from each of the obtained hyperbox by using the Latin hypercube sampling; then, the rate of good sample design points (the performance criterion satisfaction probability for the hyperbox) is calculated as follows:

TABLE 3: Comparison I: the exact, GHZ, CES-IA, SA-DIRECT, and d CMA-ES-DIRECT solutions.

	Exact	GHZ	CES-IA	SA-DIRECT	d CMA-ES-DIRECT
R_1	[0.8000, 1.0000]	[0.8058, 0.9942]	[0.8010, 0.9980]	[0.8000, 1.0000]	[0.8000, 1.0000]
R_2	[0.8245, 0.9755]	[0.8068, 0.9932]	[0.8220, 0.9770]	[0.8241, 0.9759]	[0.8246, 0.9754]
R_3	[0.8000, 1.0000]	[0.8065, 0.9935]	[0.8010, 0.9980]	[0.8001, 0.9999]	[0.8000, 1.0000]
R_4	[0.8000, 1.0000]	[0.8059, 0.9941]	[0.8010, 0.9980]	[0.8000, 1.0000]	[0.8000, 1.0000]
Volume	$1.2134e-03$	$1.2354e-03$	$1.1786e-03$	$1.2114e-03$	$1.2120e-03$
Error (%)	—	1.81	2.90	0.16	0.12

TABLE 4: The intervals of R_s on the exact, GHZ, CES-IA, SA-DIRECT, and d CMA-ES-DIRECT hyperboxes, denoted by $R_{s,\text{bounded}}$.

	Exact	GHZ	CES-IA	SA-DIRECT	d CMA-ES-DIRECT
$R_{s,\text{bounded}}$	[0.9900, 1.0000]	[0.9892, 1.0000]	[0.9900, 0.9999]	[0.9900, 1.0000]	[0.9900, 1.0000]

TABLE 5: Comparison II: the exact, GHZ, CES-IA, SA-DIRECT, and d CMA-ES-DIRECT solutions.

	Exact	GHZ	CES-IA	SA-DIRECT	d CMA-ES-DIRECT
R_1	[0.8500, 0.9500]	[0.8549, 0.9451]	[0.8482, 0.9517]	[0.8502, 0.9498]	[0.8502, 0.9498]
R_2	[0.8500, 0.9500]	[0.8553, 0.9447]	[0.8519, 0.9480]	[0.8498, 0.9502]	[0.8498, 0.9502]
R_3	[0.8500, 0.9500]	[0.8538, 0.9462]	[0.8522, 0.9477]	[0.8503, 0.9497]	[0.8503, 0.9497]
R_4	[0.8667, 0.9333]	[0.8551, 0.9449]	[0.8652, 0.9347]	[0.8664, 0.9336]	[0.8664, 0.9336]
Volume	$6.6667e-05$	$6.6977e-05$	$6.6016e-05$	$6.6656e-05$	$6.6661e-05$
Error (%)	—	0.46	0.97	0.02	0.02

TABLE 6: The intervals for R_s and C_s , denoted by $R_{s,\text{bounded}}$ and $C_{s,\text{bounded}}$, respectively, based on the exact, GHZ, CES-IA, SA-DIRECT, and d CMA-ES-DIRECT hyperboxes.

	Exact	GHZ	CES-IA	SA-DIRECT	d CMA-ES-DIRECT
$R_{s,\text{bounded}}$	[0.9955, 0.9998]	[0.9958, 0.9998]	[0.9956, 0.9998]	[0.9954, 0.9999]	[0.9958, 0.9999]
$C_{s,\text{bounded}}$	[770.0100, 849.9900]	[769.3231, 850.6769]	[770.0810, 849.9189]	[770.0000, 849.9999]	[770.0000, 849.9999]

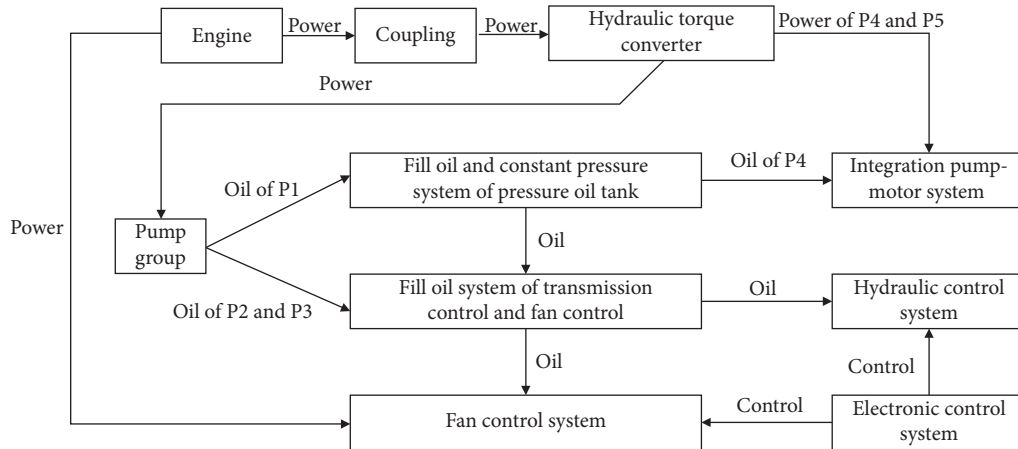


FIGURE 11: Function constitutes of PSSTCS.

$$\text{rate} = \frac{1}{N} \sum_{i=1}^N \#(f(\mathbf{R}_i) \geq f^c), \quad (22)$$

where $\#(\cdot)$ is the indicator function, i.e., $\#(f(\mathbf{R}_i) \geq f^c) = \begin{cases} 1, & f(\mathbf{R}_i) \geq f^c \\ 0, & f(\mathbf{R}_i) < f^c \end{cases}$.

The performance criterion satisfaction probabilities of the obtained hyperboxes by the GHZ, SA-DIRECT, and

d CMA-ES-DIRECT methods are shown under different sample sizes in Figure 13. It can be observed that the performance criterion satisfaction probabilities for the SA-DIRECT and d CMA-ES-DIRECT hyperboxes are all 1 while those for the GHZ hyperbox are below 1. This implies that the obtained hyperboxes by the SA-DIRECT and d CMA-ES-DIRECT methods are solution hyperboxes. Therefore, our approach is superior to the GHZ method.

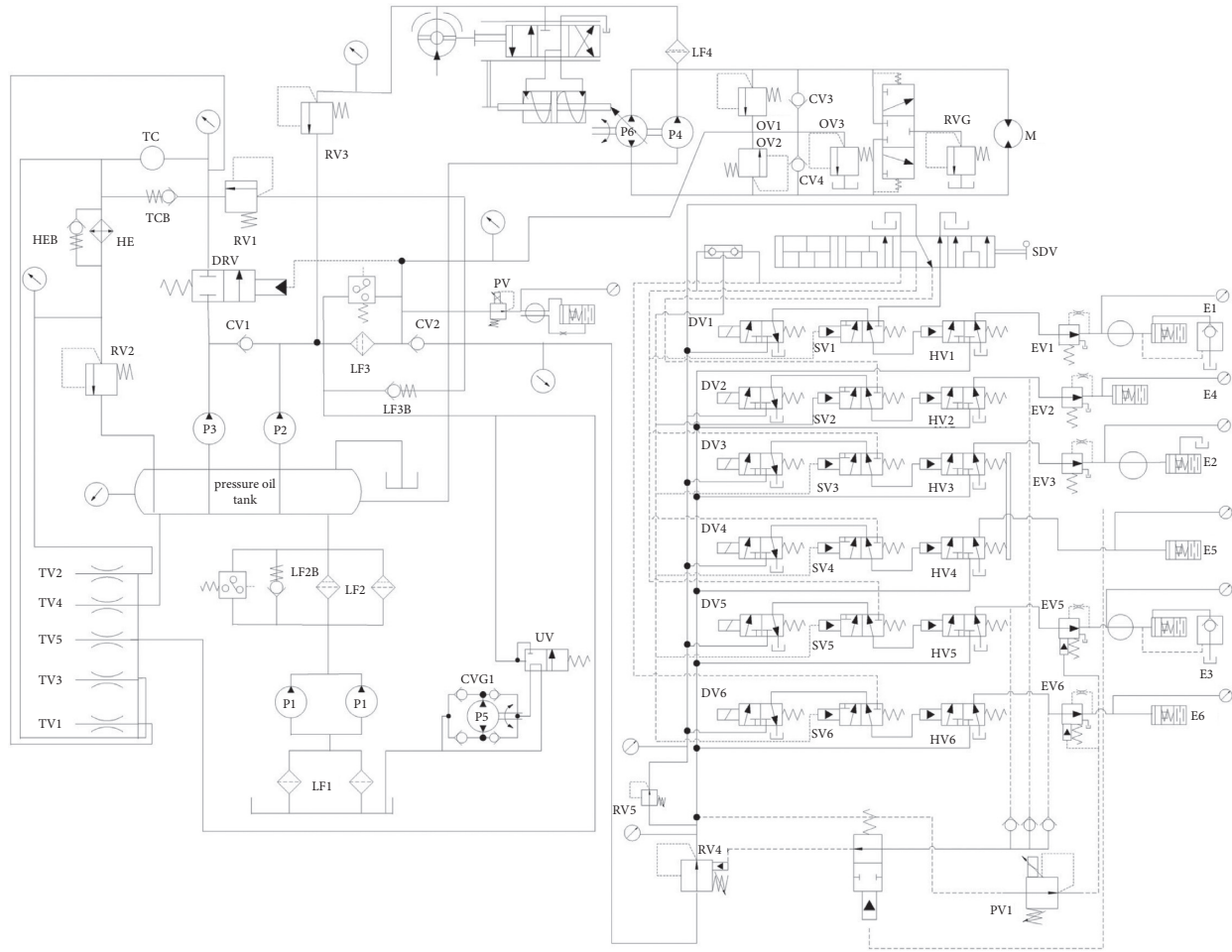


FIGURE 12: Structure principle drawing of PSSTCS.

TABLE 7: The GHZ, SA-DIRECT, and *d*CMA-ES-DIRECT solutions.

No. (variable)	Unit	GHZ	SA-DIRECT	<i>d</i> CMA-ES-DIRECT
1	Oil pan	[0.9996, 0.9999]	[0.9996, 1.0000]	[0.9997, 1.0000]
2	Filter LF1	[0.9996, 0.9998]	[0.9992, 1.0000]	[0.9994, 1.0000]
3	Filter LF1	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
4	Pump P1	[0.9995, 0.9998]	[0.9991, 1.0000]	[0.9993, 1.0000]
5	Pump P1	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
6	Power of P1, P2, and P3	[0.9996, 0.9998]	[0.9996, 1.0000]	[0.9995, 1.0000]
7	Filter LF2	[0.9995, 0.9997]	[0.9993, 1.0000]	[0.9993, 1.0000]
8	Filter LF2	[0.9995, 0.9998]	[0.9995, 1.0000]	[0.9994, 1.0000]
9	Bypass-valve LF2B	[0.9995, 0.9997]	[0.9992, 1.0000]	[0.9993, 1.0000]
10	Pressure oil tank	[0.9996, 0.9999]	[0.9993, 1.0000]	[0.9997, 1.0000]
11	Pump P3	[0.9990, 1.0000]	[0.9990, 1.0000]	[0.9990, 1.0000]
12	Pump P2	[0.9990, 1.0000]	[0.9990, 1.0000]	[0.9990, 1.0000]
13	One-way valve CV1	[0.9990, 1.0000]	[0.9990, 1.0000]	[0.9990, 1.0000]
14	Filter LF3	[0.9996, 0.9999]	[0.9997, 1.0000]	[0.9997, 1.0000]
15	One-way valve CV2	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
16	Bypass-valve LF3B	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
17	Constant pressure valve RV1	[0.9996, 0.9999]	[0.9996, 1.0000]	[0.9996, 1.0000]
18	Power P4 and P5	[0.9997, 0.9999]	[0.9999, 1.0000]	[0.9998, 1.0000]
19	Pump P4	[0.9996, 0.9999]	[0.9998, 1.0000]	[0.9997, 1.0000]
20	Filter LF4	[0.9996, 0.9999]	[0.9999, 1.0000]	[0.9997, 1.0000]
21	Constant pressure valve RV2	[0.9996, 0.9999]	[0.9996, 1.0000]	[0.9997, 1.0000]
22	Steering wheel signal	[0.9996, 0.9999]	[0.9996, 1.0000]	[0.9995, 1.0000]
23	Closed-loop feedback link	[0.9990, 1.0000]	[0.9990, 1.0000]	[0.9990, 1.0000]
24	Valve body of SDV1	[0.9996, 0.9998]	[0.9997, 1.0000]	[0.9997, 1.0000]
25	Piston valve block	[0.9996, 0.9999]	[0.9998, 1.0000]	[0.9996, 1.0000]
26	Hydraulic cylinder DLU	[0.9996, 0.9999]	[0.9997, 1.0000]	[0.9995, 1.0000]
27	Swash plate pump of P5	[0.9996, 0.9999]	[0.9995, 1.0000]	[0.9997, 1.0000]

TABLE 7: Continued.

No. (variable)	Unit	GHZ	SA-DIRECT	dCMA-ES-DIRECT
28	Two-way variable displacement pump P5	[0.9990, 1.0000]	[0.9990, 1.0000]	[0.9990, 1.0000]
29	Overflow valve OV1	[0.9996, 0.9999]	[0.9995, 1.0000]	[0.9996, 1.0000]
30	One-way valve CV4	[0.9996, 0.9999]	[0.9998, 1.0000]	[0.9995, 1.0000]
31	Overflow valve OV2	[0.9990, 1.0000]	[0.9990, 1.0000]	[0.9990, 1.0000]
32	One-way valve CV3	[0.9990, 1.0000]	[0.9990, 1.0000]	[0.9990, 1.0000]
33	Group valve RVG	[0.9997, 0.9999]	[0.9998, 1.0000]	[0.9996, 1.0000]
34	Reversing motor M	[0.9996, 0.9999]	[0.9998, 1.0000]	[0.9997, 1.0000]
35	Power of electronic control system	[0.9990, 1.0000]	[0.9990, 1.0000]	[0.9990, 1.0000]
36	Control panel	[0.9990, 1.0000]	[0.9990, 1.0000]	[0.9990, 1.0000]
37	Panel switch	[0.9990, 1.0000]	[0.9990, 1.0000]	[0.9990, 1.0000]
38	Handle signal	[0.9990, 1.0000]	[0.9990, 1.0000]	[0.9990, 1.0000]
39	State signal sensor	[0.9990, 1.0000]	[0.9990, 1.0000]	[0.9990, 1.0000]
40	Switch D1	[0.9996, 0.9999]	[0.9994, 1.0000]	[0.9996, 1.0000]
41	Switch D2	[0.9996, 0.9999]	[0.9993, 1.0000]	[0.9997, 1.0000]
42	Pilot valve PV2	[0.9997, 0.9999]	[0.9998, 1.0000]	[0.9998, 1.0000]
43	Liquid viscous clutch cylinder of left fan	[0.9996, 0.9999]	[0.9995, 1.0000]	[0.9996, 1.0000]
44	Active friction plate of left pan	[0.9996, 0.9999]	[0.9996, 1.0000]	[0.9997, 1.0000]
45	Friction plate of left pan	[0.9997, 0.9999]	[0.9993, 1.0000]	[0.9997, 1.0000]
46	Power of liquid viscous clutch	[0.9996, 0.9999]	[0.9995, 1.0000]	[0.9997, 1.0000]
47	Liquid viscous clutch cylinder of right fan	[0.9996, 0.9999]	[0.9999, 1.0000]	[0.9996, 1.0000]
48	Active friction plate of right fan	[0.9996, 0.9999]	[0.9998, 1.0000]	[0.9997, 1.0000]
49	Friction plate of right fan	[0.9996, 0.9999]	[0.9997, 1.0000]	[0.9996, 1.0000]
50	Overflow valve OV3	[0.9996, 0.9999]	[0.9999, 1.0000]	[0.9997, 1.0000]
51	Throttle valve TV1	[0.9995, 0.9998]	[0.9992, 1.0000]	[0.9993, 1.0000]
52	Throttle valve TV2	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
53	Overflow valve RV3	[0.9996, 0.9999]	[0.9997, 1.0000]	[0.9997, 1.0000]
54	Overflow valve RV4	[0.9996, 0.9999]	[0.9998, 1.0000]	[0.9997, 1.0000]
55	Signal of SDV	[0.9996, 0.9999]	[0.9999, 1.0000]	[0.9997, 1.0000]
56	Hand valve SDV	[0.9996, 0.9999]	[0.9995, 1.0000]	[0.9996, 1.0000]
57	Electric controllable valve DV1	[0.9995, 0.9998]	[0.9991, 1.0000]	[0.9993, 1.0000]
58	Manual hydraulic control valve SV1	[0.9995, 0.9998]	[0.9990, 1.0000]	[0.9993, 1.0000]
59	Liquid control valve HV1	[0.9995, 0.9998]	[0.9994, 1.0000]	[0.9993, 1.0000]
60	Constant pressure throttle valve EV1	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
61	Oil cylinder E1	[0.9995, 0.9998]	[0.9995, 1.0000]	[0.9993, 1.0000]
62	Electric controllable valve DV2	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
63	Manual hydraulic control valve SV2	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9995, 1.0000]
64	Liquid control valve HV2	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
65	Constant pressure throttle valve EV2	[0.9992, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
66	Oil cylinder F1	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
67	Electric controllable valve DV3	[0.9996, 0.9998]	[0.9994, 1.0000]	[0.9993, 1.0000]
68	Manual hydraulic control valve SV3	[0.9995, 0.9998]	[0.9994, 1.0000]	[0.9993, 1.0000]
69	Liquid control valve HV3	[0.9995, 0.9998]	[0.9990, 1.0000]	[0.9993, 1.0000]
70	Constant pressure throttle valve EV3	[0.9992, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
71	Oil cylinder E2	[0.9995, 0.9998]	[0.9994, 1.0000]	[0.9993, 1.0000]
72	Electric controllable valve DV4	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
73	Manual hydraulic control valve SV4	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9995, 1.0000]
74	Liquid control valve HV4	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
75	Oil cylinder F2	[0.9995, 0.9998]	[0.9990, 1.0000]	[0.9993, 1.0000]
76	Electric controllable valve DV5	[0.9995, 0.9998]	[0.9991, 1.0000]	[0.9993, 1.0000]
77	Manual hydraulic control valve SV5	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
78	Liquid control valve HV5	[0.9995, 0.9998]	[0.9991, 1.0000]	[0.9994, 1.0000]
79	Constant pressure throttle valve EV5	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
80	Oil cylinder E3	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
81	Electric controllable valve DV6	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
82	Manual hydraulic control valve SV6	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
83	Liquid control valve HV6	[0.9995, 0.9998]	[0.9993, 1.0000]	[0.9993, 1.0000]
84	Constant pressure throttle valve EV6	[0.9995, 0.9998]	[0.9994, 1.0000]	[0.9993, 1.0000]
85	Oil cylinder F3	[0.9995, 0.9998]	[0.9992, 1.0000]	[0.9993, 1.0000]
86	Pilot valve PV1	[0.9996, 0.9999]	[0.9995, 1.0000]	[0.9997, 1.0000]

TABLE 8: The log-volumes of the GHZ, SA-DIRECT, and dCMA-ES-DIRECT hyperboxes.

Method	GHZ	SA-DIRECT	dCMA-ES-DIRECT
Log-volume	-686.1204	-646.3142	-646.8860

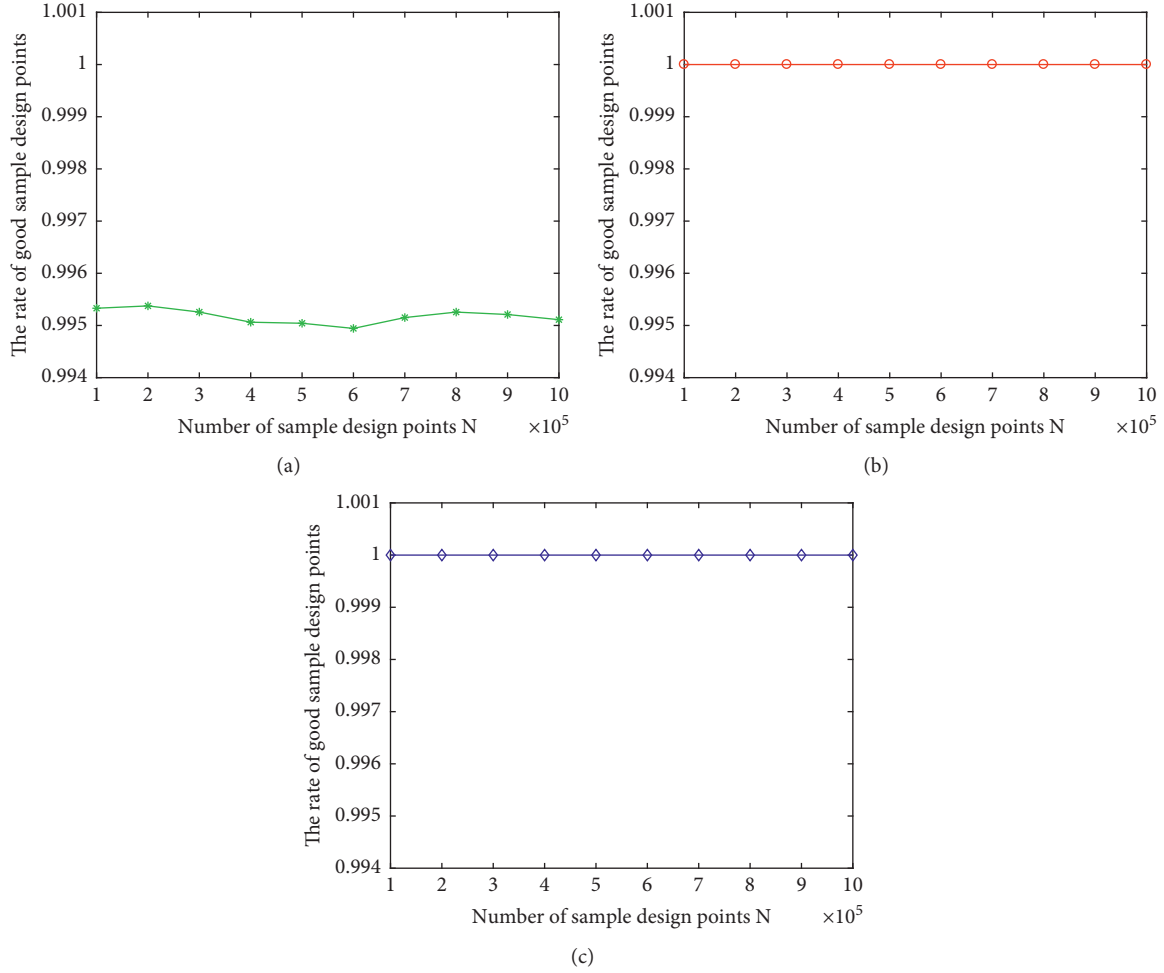


FIGURE 13: The performance criterion satisfaction probabilities for the obtained hyperboxes. (a) GHZ method, (b) SA-DIRECT method, and (c) d CMA-ES-DIRECT method.

TABLE 9: Default strategy parameters values.

Selection and recombination:

Number of parents for recombination $\mu = \lfloor d_s/2 \rfloor$

Normalized array for weighted recombination: $w_{i=1,\dots,\mu} = (\ln(\mu+1) - \ln(i)) / \sum_{j=1}^{\mu} (\ln(\mu+1) - \ln(j))$

Step-size control:

Reciprocal of the backward time horizon of the evolution path $c_{\sigma} = ((\mu_w + 2)/(n + \mu_w + 3))$

Damping parameter $d_{\sigma} = 1 + 2 \max(0, \sqrt{((\mu_w - 1)/(n + 1))} - 1) + c_{\sigma}$

Covariance matrix adaptation:

Reciprocal of the backward time horizon of the evolution path $c_c = (4/(n + 4))$

Relative weighting $\mu_{cov} = \mu_w$

Learning rate $c_{cov} = (1/\mu_{cov})(2/(n + \sqrt{2})^2) + (1 - (1/\mu_{cov}))\min(1, ((2\mu_w - 1)/((n + 2)^2 + \mu_w))$

TABLE 10: Function GO operator in the GO model of PSSTCS.

Unit	No. (operator)	Type (operator)
Oil pan	1	5
Pump P1	5, 6	6
Filter LF2	9, 10	1
Pressure oil tank	15	1
Pump P2	17	6
Filter LF3	20	1
Bypass-valve LF3B	23	1
Power P4 and P5	26	Virtual 5
Filter LF4	28	1
Steering wheel signal	31	Virtual 5
Valve body of SDV1	32	1

TABLE 10: Continued.

Unit	No. (operator)	Type (operator)
Hydraulic cylinder DLU	34	1
Two-way variable displacement pump P5	36	6
One-way valve CV4	38	1
One-way valve CV3	41	1
Reversing motor M	44	21
Control panel	47	6
Handle signal	49	Virtual 5
Switch D1	52	1
Pilot valve PV2	55	22
Active friction plate of left fan	58	6
Power of liquid viscous clutch	60	Virtual 5
Active friction plate of right fan	63	6
Overflow valve OV3	67	1
Throttle valve TV2	69	1
Overflow valve RV4	72	1
Hand valve SDV	74	6
Manual hydraulic control valve SV1	77	6
Constant pressure throttle valve EV1	79	1
Electric controllable valve DV2	82	6
Liquid control valve HV2	85	1
Oil cylinder F1	87	1
Manual hydraulic control valve SV3	91	6
Constant pressure throttle valve EV3	93	1
Electric controllable valve DV4	96	6
Liquid control valve HV4	99	1
Electric controllable valve DV5	102	6
Liquid control valve HV5	105	1
Oil cylinder E3	107	1
Manual hydraulic control valve SV6	111	6
Constant pressure throttle valve EV6	113	6
Pilot valve PV1	116	6
Filter LF1	2, 3	1
Power of P1, P2, and P3	7	Virtual 5
Bypass-valve LF2B	13	1
Pump P3	16	6
One-way valve CV1	18	1
One-way valve CV2	21	1
Constant pressure valve RV1	25	19
Pump P4	27	6
Constant pressure valve RV2	29	19
Closed-loop feedback link	30	24
Piston valve block	33	1
Swash plate pump of P5	35	1
Overflow valve OV1	37	1
Overflow valve OV2	40	1
Group valve RVG	43	21
Power of electronic control system	46	5
Panel switch	48	5
State signal sensor	50	5
Switch D2	53	1
Liquid viscous clutch cylinder of left fan	56	1
Friction plate of left fan	59	1
Liquid viscous clutch cylinder of right fan	61	1
Friction plate of right fan	64	1
Throttle valve TV1	68	1
Overflow valve RV3	71	1
Signal of SDV	73	Virtual 5
Electric controllable valve DV1	75	6
Liquid control valve HV1	78	1
Oil cylinder E1	80	1
Manual hydraulic control valve SV2	84	6
Constant pressure throttle valve EV2	86	1
Electric controllable valve DV3	89	6
Liquid control valve HV3	92	1
Oil cylinder E2	94	1
Manual hydraulic control valve SV4	98	6
Oil cylinder F2	100	1

TABLE 10: Continued.

Unit	No. (operator)	Type (operator)
Manual hydraulic control valve SV5	103	6
Constant pressure throttle valve EV5	106	6
Electric controllable valve DV6	109	6
Liquid control valve HV6	112	1
Oil cylinder F3	114	1

TABLE 11: Logical GO operator and auxiliary GO operator in the GO model of PSSTCS.

Logical relationship	No. (operator)	Type (operator)
OR	4, 8, 11, 19, 66, 70, 39, 42, 51, 57, 62, 65, 117, 118	2
AND	119, 120, 121, 122, 123, 124, 125, 126, 129	10
Condition signal flow	12, 22	20
Standby	14, 24	18A
Auxiliary operator for combinational signal	54	15B
Auxiliary operator for GO operation	76, 83, 90, 97, 103, 110	25A
Auxiliary operator for GO operation	81, 88, 95, 101, 108, 115	25B
Multi-function integration control logic	45, 127, 128	23

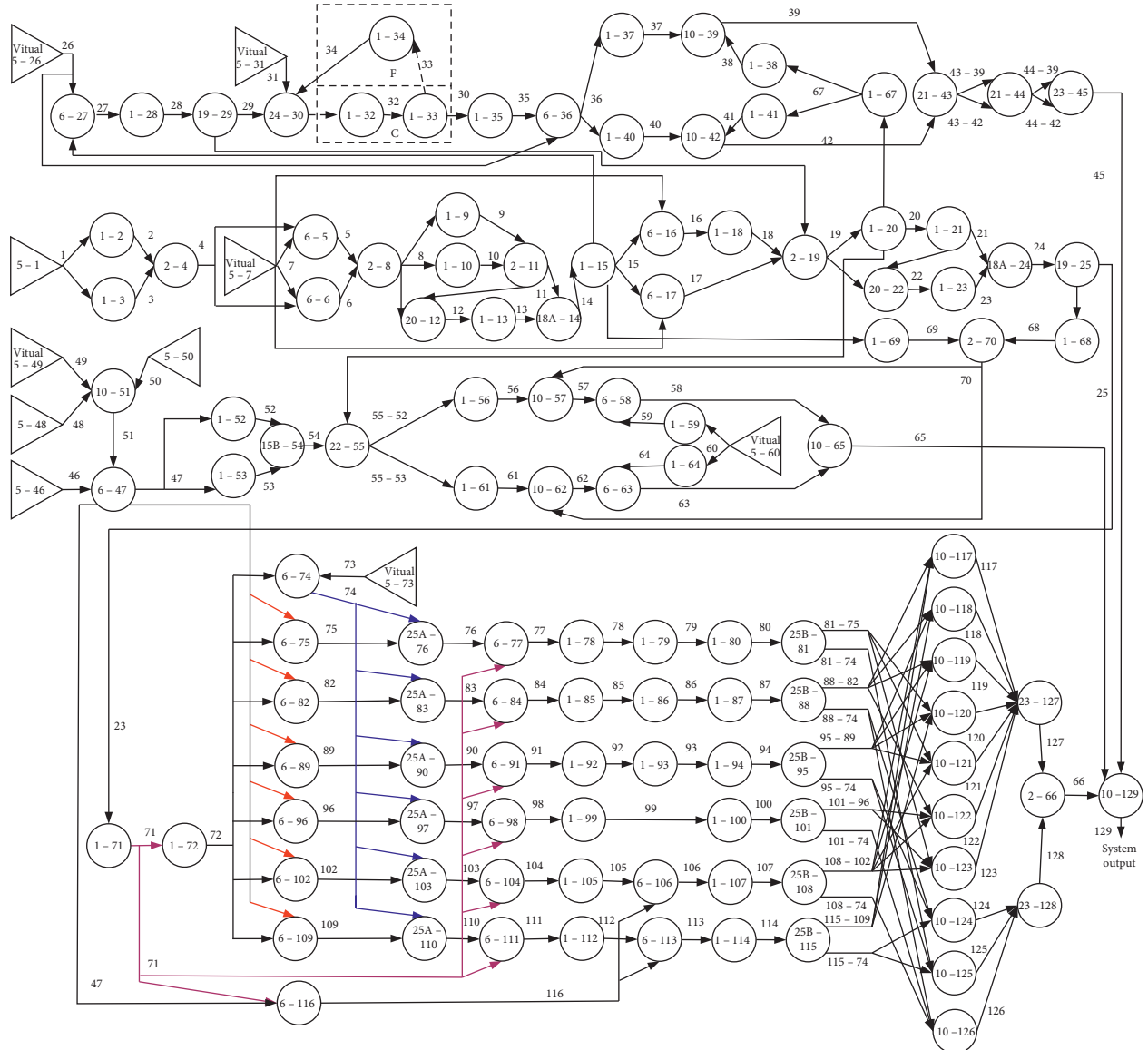


FIGURE 14: GO model of PSSTCS.

7. Conclusions

To improve the design robustness, rather than optimizing one single design point, the approach presented in this paper maximizes the region in the design space where all design points meet the required performance goal. Moreover, regions expressed by hyperboxes are considered in order to decouple parameters.

To this end, an innovative global approach that combines metaheuristic algorithms and the DIRECT algorithm is proposed to seek a maximum solution hyperbox. The metaheuristic algorithm is used to obtain a maximum hyperbox and the DIRECT algorithm is used as a checking technique to guarantee that the obtained hyperbox is a solution hyperbox. More specifically, the SA-DIRECT and *d*CMA-ES-DIRECT methods are presented in detail. The results of studies on complex numerical examples and engineering cases have shown that these two methods have better performance than the GHZ and CES-IA methods. Since the DIRECT algorithm only requires evaluating the values of the performance function at the sample points, the proposed approach can be used for both analytically known and black-box performance functions. The performance function is continuous in most engineering problems; therefore, the proposed approach is a powerful tool that engineering designers can use to obtain a maximum solution hyperbox.

The optimality of the proposed approach mainly depends on the properties of the metaheuristic algorithms. The two metaheuristic algorithms we adopt are global search and have great possibility to reach the globally optimal solution. Our work can be further improved provided that more advanced global search algorithms are developed.

Appendix

A. Covariance Matrix Adaptation Evolution Strategy

An in-depth discussion on initial values of parameters of covariant matrix adaptation evolution strategy is given in [53]. In the *j*th deme, $j = 1, \dots, d_s$, evolution path $\mathbf{p}_{\sigma j} = \mathbf{0}$ and $\mathbf{p}_{c j} = \mathbf{0}$, the initial distribution mean \mathbf{m}_{0j} is chosen randomly and uniformly in the design space $[\mathbf{x}^l, \mathbf{x}^u]$, and the step-size $\sigma_{0j} = 0.3$. However, the initial covariance matrix C_{0j} is problem dependent. Specifically, if the design interval is $[a_i, b_i]$, different design intervals for different variables can be reflected by an initialization of C_{0j} , in which the main diagonal elements of C_{0j} obey $c_{ii} = (b_i - a_i)^2$ and the entries outside the main diagonal are all zero. Note that $b_i - a_i$'s should not disagree by several orders of magnitude. Otherwise, a scaling of the variables should be applied. Therefore, in this paper, we set $C_{0j} = \text{diag}((x_1^u - x_1^l)^2, \dots, (x_m^u - x_m^l)^2, (x_1^u - x_1^l)^2, \dots, (x_m^u - x_m^l)^2)$.

The termination condition for *d*CMA-ES-DIRECT is the number of generations. The total number of generations g_{\max} for the *d*CMA-ES-DIRECT method is the same as the number of the outer iterations for the SA-DIRECT method.

In the covariance matrix adaptation evolution strategy, a population of new individuals is generated by sampling from

a multivariate normal distribution. In the optimization problem (7), the basic equation for sampling d_s individuals, for the generation number *g* and the *j*th deme, is

$$\mathbf{x}_k^{\text{bound}} \sim N(\mathbf{m}_{gj}, \sigma_{gj}^2 C_{gj}), \quad k = (j-1) \times d_s + 1, \dots, j \times d_s, \quad (\text{A.1})$$

where $N(\mathbf{m}_{gj}, \sigma_{gj}^2 C_{gj})$ is a normal distribution with mean \mathbf{m}_{gj} and covariance matrix $\sigma_{gj}^2 C_{gj}$. As defined in Section 2.2, $\mathbf{x}^{\text{bound}} = (\mathbf{x}^{\text{low}}, \mathbf{x}^{\text{up}})$. The *d*CMA-ES depends on selection and recombination and step-size control, as well as covariance matrix adaptation, i.e., how to calculate $\mathbf{m}_{g+1,j}$, $C_{g+1,j}$, and $\sigma_{g+1,j}$ for the next generation *g* + 1 and the *j*th deme.

The new mean value is computed as

$$\mathbf{m}_{g+1,j} = \sum_{i=1}^{\mu} w_i \mathbf{x}_{i: d_s}^{\text{bound}} = \mathbf{m}_{gj} + \sum_{i=1}^{\mu} w_i (\mathbf{x}_{i: d_s}^{\text{bound}} - \mathbf{m}_{gj}), \quad (\text{A.2})$$

where w_i is the recombination weight, $\sum_{i=1}^{\mu} w_i = 1$, $\mathbf{x}_{i: d_s}^{\text{bound}}$ denotes the *i*th best individual out of $\mathbf{x}_{k_1}^{\text{bound}}, \dots, \mathbf{x}_{k_2}^{\text{bound}}$, $k_1 = d_s \times (j-1) + 1$, and $k_2 = d_s \times j$.

The step-size σ_{gj} is updated using the cumulative step-size adaptation (CSA). The evolution path $\mathbf{p}_{\sigma j}$ is updated first:

$$\mathbf{p}_{\sigma j} \leftarrow (1 - c_{\sigma}) \mathbf{p}_{\sigma j} + \sqrt{c_{\sigma} (2 - c_{\sigma})} \mu_w C_{gj}^{-1/2} \frac{\mathbf{m}_{g+1,j} - \mathbf{m}_{gj}}{\sigma_{gj}}, \quad (\text{A.3})$$

$$\sigma_{g+1,j} = \sigma_{gj} \exp\left(\frac{c_{\sigma}}{d_{\sigma}} \left(\frac{\|\mathbf{p}_{\sigma j}\|}{E\|N(\mathbf{0}, I)\|} - 1\right)\right), \quad (\text{A.4})$$

where $\mu_w = (\sum_{i=1}^{\mu} w_i)^{-1}$ is the variance effective selection mass, $E\|N(\mathbf{0}, I)\| = (\sqrt{2} \Gamma((2m+1)/2) / \Gamma(m)) \approx \sqrt{2m} (1 - (1/8m) + (1/84m^2))$, *I* denotes the identity matrix, and d_{σ} is the damping parameter.

Finally, the covariance matrix C_{gj} is updated. The evolution path $\mathbf{p}_{c j}$ is updated first:

$$\mathbf{p}_{c j} \leftarrow (1 - c_c) \mathbf{p}_{c j} + h_{\sigma} \sqrt{c_c (2 - c_c)} \sqrt{\mu_w} \frac{\mathbf{m}_{g+1,j} - \mathbf{m}_{gj}}{\sigma_{gj}}, \quad (\text{A.5})$$

$$C_{g+1,j} = (1 - c_{\text{cov}}) C_{gj} + \frac{c_{\text{cov}}}{\mu_{\text{cov}}} (\mathbf{p}_{c j} \mathbf{p}_{c j}^T + \delta (h_{\sigma}) C_{gj}) + c_{\text{cov}} \left(1 - \frac{1}{\mu_{\text{cov}}}\right) \sum_{i=1}^{\mu} w_i \frac{\mathbf{x}_{i: d_s} - \mathbf{m}_{gj}}{\sigma_{gj}} \left(\frac{\mathbf{x}_{i: d_s} - \mathbf{m}_{gj}}{\sigma_{gj}}\right)^T, \quad (\text{A.6})$$

where T denotes the transpose, $\delta(h_{\sigma}) = (1 - h_{\sigma}) c_c (2 - c_c)$ is minor relevance, and

$$h_{\sigma} = \begin{cases} 1, & \text{if } \frac{\mathbf{p}_{c j}}{1 - (1 - c_{\sigma})^{2(g+1)}} < \left(1.4 + \frac{2}{2m+1}\right) E\|N(\mathbf{0}, I)\|, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A.7})$$

The default strategy parameters are μ , $w_{i=1,\dots,\mu}$, c_σ , d_σ , c_c , μ_{cov} , and c_{cov} . Default strategy parameters values are given in [53] and shown in Table 9. Hansen [53] has pointed out these default parameters are in particular chosen to be a robust setting, and thus are not recommended to be changed.

B. The Goal-Oriented (GO) Reliability Assessment Method

The goal-oriented (GO) reliability assessment method for the power-shift steering transmission control system (PSSTCS) is given in [54] in detail.

The function GO operators, logical GO operators, and auxiliary GO operators are selected to describe the unit itself, its logical relationships, and auxiliary GO operation in the PSSTCS, respectively, as presented in Tables 10 and 11.

The GO model of the PSSTCS is developed by using the signal flows to connect the above GO operators, as shown in Figure 14. The signal flow 129 is the system's reliability output.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 71801196, National Major Project from the Shanghai Nuclear Engineering Research and Design Institute under Grant 2017ZX06002006, and China Postdoctoral Science Foundation under Grant 2018M631606.

References

- [1] W. L. Oberkampf, *Uncertainty Quantification Using Evidence Theory*, Stanford University, Stanford, CA, USA, 2005.
- [2] D. Lim, Y. S. Ong, B. Sendhoff, and B. S. Lee, "Inverse multi-objective robust evolutionary design optimization," *Genetic Programming and Evolvable Machines*, vol. 7, no. 4, pp. 383–404, 2007.
- [3] J. Marczyk, "Stochastic multidisciplinary improvement: beyond optimization," in *Proceedings of the 8th Symposium on Multidisciplinary Analysis and Optimization*, pp. 2000–4929, Long Beach, CA, USA, September 2000.
- [4] G.-J. Park, T.-H. Lee, K. H. Lee, and K.-H. Hwang, "Robust design: an overview," *AIAA Journal*, vol. 44, no. 1, pp. 181–191, 2006.
- [5] G. Taguchi, E. Elsayed, and T. Hsiang, *Quality Engineering in Production Systems*, McGraw-Hill, New York, NY, USA, 1989.
- [6] B. Wang, X. Chen, F. M. Xin, and X. Song, "MI-based robust waveform design in radar and jammer games," *Complexity*, vol. 2019, Article ID 4057849, 14 pages, 2019.
- [7] F. Turki, H. Gritli, and S. Belghith, "Robust position control of a two-sided 1-dof impacting mechanical oscillator subject to an external persistent disturbance by means of a state-feedback controller," *Complexity*, vol. 2019, Article ID 9174284, 14 pages, 2019.
- [8] Y.-G. Zhao and T. Ono, "A general procedure for first/second-order reliability method (FORM/SORM)," *Structural Safety*, vol. 21, no. 2, pp. 95–112, 1999.
- [9] A. Saltelli and M. Scott, "Guest editorial: the role of sensitivity analysis in the corroboration of models and its link to model structural and parametric uncertainty," *Reliability Engineering & System Safety*, vol. 57, no. 1, pp. 1–4, 1997.
- [10] M. Zimmermann and J. E. von Hoessle, "Computing solution spaces for robust design," *International Journal for Numerical Methods in Engineering*, vol. 94, no. 3, pp. 290–307, 2013.
- [11] L. Graff, H. Harbrecht, and M. Zimmermann, "On the computation of solution spaces in high dimensions," *Structural and Multidisciplinary Optimization*, vol. 54, no. 4, pp. 811–829, 2016.
- [12] M. Claudio, S. Rocco, and E. Hernandez, "Robustness and sensitivity analysis in multiple criteria decision problems using rule learner techniques," *Reliability Engineering & System Safety*, vol. 134, pp. 297–304, 2015.
- [13] Y. Chen, J. Shi, and X. J. Yi, "A new reliable operating region design method," *Mathematical Problems in Engineering*, vol. 2020, Article ID 9425031, 12 pages, 2020.
- [14] R. E. Moore, *Interval Analysis*, Prentice-Hall, Englewood Cliff, NJ, USA, 1966.
- [15] C. M. Rocco, J. A. Moreno, and N. Carrasquero, "Robust design using a hybrid-cellular-evolutionary and interval-arithmetic approach: a reliability application," *Reliability Engineering & System Safety*, vol. 79, no. 2, pp. 149–159, 2003.
- [16] E. A. Bender, *An Introduction to Mathematical Modeling*, Dover Publications, New York, NY, USA, 2000.
- [17] E. M. T. Hendrix, C. J. Mecking, and T. H. B. Hendriks, "Finding robust solutions for product design problems," *European Journal of Operational Research*, vol. 92, no. 1, pp. 28–36, 1996.
- [18] D. E. Kvasov and Y. D. Sergeyev, "Deterministic approaches for solving practical black-box global optimization problems," *Advances in Engineering Software*, vol. 80, pp. 58–66, 2015.
- [19] Y. D. Sergeyev and D. E. Kvasov, "A deterministic global optimization using smooth diagonal auxiliary functions," *Communications in Nonlinear Science and Numerical Simulation*, vol. 21, no. 1–3, pp. 99–111, 2015.
- [20] D. Lera and Y. D. Sergeyev, "GOSH: derivative-free global optimization using multi-dimensional space-filling curves," *Journal of Global Optimization*, vol. 71, no. 1, pp. 193–211, 2018.
- [21] Y. D. Sergeyev, P. Pugliese, and D. Famularo, "Index information algorithm with local tuning for solving multidimensional global optimization problems with multiextremal constraints," *Mathematical Programming*, vol. 96, no. 3, pp. 489–512, 2003.
- [22] Y. D. Sergeyev, D. E. Kvasov, and M. S. Mukhametzhano, "On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget," *Scientific Reports*, vol. 8, no. 1, pp. 1–9, 2018.
- [23] D. Lera and Y. D. Sergeyev, "Acceleration of univariate global optimization algorithms working with lipschitz functions and lipschitz first derivatives," *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 508–529, 2013.
- [24] R. Paulavicius, Y. D. Sergeyev, D. E. Kvasov, and J. Žilinskas, "Globally-biased disimpl algorithm for expensive global

- optimization,” *Journal of Global Optimization*, vol. 59, no. 2-3, pp. 545–567, 2014.
- [25] R. Paulavičius, J. Žilinskas, and A. Grothey, “Investigation of selection strategies in branch and bound algorithm with simplicial partitions and combination of lipschitz bounds,” *Optimization Letters*, vol. 4, no. 2, pp. 173–183, 2010.
- [26] S. A. Piyavskii, “An algorithm for finding the absolute minimum of a function,” *Theory of Optimal Decisions*, vol. 2, no. 13–24, 1967.
- [27] B. O. Shubert, “A sequential method seeking the global maximum of a function,” *SIAM Journal on Numerical Analysis*, vol. 9, no. 3, pp. 379–388, 1972.
- [28] L. Chiter, “DIRECT algorithm: a new definition of potentially optimal hyperrectangles,” *Applied Mathematics and Computation*, vol. 179, no. 2, pp. 742–749, 2006.
- [29] L. Chiter, “A new sampling method in the DIRECT algorithm,” *Applied Mathematics and Computation*, vol. 175, no. 1, pp. 297–306, 2006.
- [30] R. Paulavičius, L. Chiter, and J. Žilinskas, “Global optimization based on bisection of rectangles, function values at diagonals, and a set of lipschitz constants,” *Journal of Global Optimization*, vol. 71, no. 1, pp. 5–20, 2018.
- [31] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, “Lipschitzian optimization without the lipschitz constant,” *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.
- [32] M. Björkman and K. Holmström, “Global optimization using the direct algorithm in matlab,” *Matlab Advanced Modeling & Optimization*, vol. 1, no. 2, pp. 1–8, 1999.
- [33] J. M. Gablonsky, “Modifications of the direct algorithm,” Ph. D thesis, North Carolina State University, Raleigh, NC, USA, 2001.
- [34] A. Hedar, “test functions for unconstrained global optimization,” 2005, http://www.optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm.
- [35] D. E. Finkel and C. T. Kelley, “Additive scaling and the direct algorithm,” *Journal of Global Optimization*, vol. 36, no. 4, pp. 597–608, 2006.
- [36] G. Liuzzi, S. Lucidi, and V. Piccialli, “A direct-based approach exploiting local minimizations for the solution of large-scale global optimization problems,” *Computational Optimization and Applications*, vol. 45, no. 2, pp. 353–375, 2010.
- [37] Y. D. Sergeyev and D. E. Kvasov, “Global search based on efficient diagonal partitions and a set of lipschitz constants,” *SIAM Journal on Optimization*, vol. 16, no. 3, pp. 910–937, 2006.
- [38] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [39] W. L. Goffe, G. D. Ferrier, and J. Rogers, “Global optimization of statistical functions with simulated annealing,” *Journal of Econometrics*, vol. 60, no. 1-2, pp. 65–99, 1994.
- [40] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [41] V. Ravi, B. S. N. Murty, and J. Reddy, “Nonequilibrium simulated-annealing algorithm applied to reliability optimization of complex systems,” *IEEE Transactions on Reliability*, vol. 46, no. 2, pp. 233–239, 1997.
- [42] F. Kursawe, “Evolution strategies-simple models of natural process?” *Revue Internationale de Systemique*, vol. 7, no. 5, pp. 627–642, 1993.
- [43] N. Hansen and A. Ostermeier, “Completely derandomized self-adaptation in evolution strategies,” *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [44] N. Hansen, “The CMA evolution strategy: a comparing review, towards a new evolutionary computation,” in *Advances in Estimation of Distribution Algorithms*, Springer, Berlin, Germany, 2006.
- [45] C. Igel, T. Suttorp, and N. Hansen, “A computational efficient covariance matrix update and a (1+1)-cma for evolution strategies,” in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation Conference (GECCO 2006)*, ACM Press, Seattle, WA, USA, July 2006.
- [46] E. Alba and J. M. Troya, “A survey of parallel distributed genetic algorithms,” *Complexity*, vol. 4, no. 4, pp. 31–52, 1999.
- [47] H. S. Huang, “Distributed genetic algorithm for optimization of wind farm annual profits,” in *Proceedings of the 14th International Conference on Intelligent System Applications to Power Systems, ISAP 2007*, Kaohsiung, Taiwan, November 2007.
- [48] A. Hendriks, K. C. Wiese, E. Glen, and A. Deschenes, “A distributed genetic algorithm for rna secondary structure prediction,” in *Proceedings of the Congress on Evolutionary Computation*, Portland, OR, USA, June 2004.
- [49] S. S. Rao, *Engineering Optimization Theory and Practice*, John Wiley & Sons, Hoboken, NJ, USA, 2009.
- [50] H. W. Kuhn and A. W. Tucker, “Nonlinear programming,” in *Proceedings of the 2nd Berkeley Symposium*, University of California Press, Berkeley, CA, USA, pp. 481–492, 1951.
- [51] L. Stripinis, R. Paulavičius, and J. Žand iči, “Improved scheme for selection of potentially optimal hyper-rectangles in direct,” *Optimization Letters*, vol. 12, no. 7, pp. 1699–1712, 2018.
- [52] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press, London, UK, 1990.
- [53] N. Hansen, “The CMA evolution strategy: a tutorial,” 2016, <https://www.researchgate.net/publication/248797670>.
- [54] X.-J. Yi, J. Shi, B. S. Dhillon, P. Hou, and H.-P. Dong, “A new reliability analysis method for repairable systems with closed-loop feedback links,” *Quality and Reliability Engineering International*, vol. 34, no. 3, pp. 298–332, 2018.