

Article

Robust Optimization with Interval Uncertainties Using Hybrid State Transition Algorithm

Haochuan Zhang, Jie Han *, Xiaojun Zhou and Yuxuan Zheng

School of Automation, Central South University, Changsha 410083, China; 8207200618@csu.edu.cn (H.Z.); michael.x.zhou@csu.edu.cn (X.Z.); 224612187@csu.edu.cn (Y.Z.)

* Correspondence: hanjie@csu.edu.cn

Abstract: Robust optimization is concerned with finding an optimal solution that is insensitive to uncertainties and has been widely used in solving real-world optimization problems. However, most robust optimization methods suffer from high computational costs and poor convergence. To alleviate the above problems, an improved robust optimization algorithm is proposed. First, to reduce the computational cost, the second-order Taylor series surrogate model is used to approximate the robustness indices. Second, to strengthen the convergence, the state transition algorithm is studied to explore the whole search space for candidate solutions, while sequential quadratic programming is adopted to exploit the local area. Third, to balance the robustness and optimality of candidate solutions, a preference-based selection mechanism is investigated which effectively determines the promising solution. The proposed robust optimization method is applied to obtain the optimal solutions of seven examples that are subject to decision variables and parameter uncertainties. Comparative studies with other robust optimization algorithms (robust genetic algorithm, Kriging metamodel-assisted robust optimization method, etc.) show that the proposed method can obtain accurate and robust solutions with less computational cost.

Keywords: robust optimization; global optimization; surrogate model; selection mechanism



Citation: Zhang, H.; Han, J.; Zhou, X.; Zheng, Y. Robust Optimization with Interval Uncertainties Using Hybrid State Transition Algorithm. *Electronics* **2023**, *12*, 3035. <https://doi.org/10.3390/electronics12143035>

Academic Editors: Padma Iyengar and Elke Pulvermüller

Received: 8 June 2023

Revised: 28 June 2023

Accepted: 6 July 2023

Published: 11 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Real-world optimization problems are subject to uncertainties due to, for example, the presence of uncontrolled changes in environmental conditions [1,2], the lack of complete knowledge of models [3,4], and the manufacturing tolerances on actual processes [5]. According to the classification of optimization problems under uncertainties [6], deriving an optimal solution that is insensitive to uncertainties is defined as robust optimization (RO).

An optimal solution is robust if, when uncertainties exist, the values of the corresponding objective functions and constraint functions fluctuate within acceptable ranges [7]. By adjusting the decision variables to counteract the effects of uncertainties, the performance of optimal solution can be improved and the degradations of objective or constraint function values can also be avoided. In general, there are two methods to describe the uncertain parameters in the optimization problem: probabilistic models and nonprobabilistic models. Probabilistic uncertainties are customarily based on the statistic information, such as the mean and variance, and they are usually handled by optimizing the expected value of the solution [8]. Moreover, probabilistic uncertainties can also be modeled by other methods, such as fitness approximation [9], multiobjective approach [10,11], and sampling-based methods [12]. Since it is difficult to obtain the accurate probability distribution information of uncertain parameters and the optimal solution can hardly guarantee complete robustness, the applications of probabilistic robust optimization methods have been limited. Nonprobabilistic uncertainties methods are usually based on interval uncertainty modeling [13,14], evidence theory [15], and possibility theory [16]. In this paper, the uncertainties are modeled as a certain interval that can be obtained without a presumed probability

distribution. Such interval uncertainty can be analyzed by the worst-case scenario [17], and this method has received considerable attention in the robust optimization [18]. The main idea of the worst-case scenario is to have the best worst-case performance in the presence of uncertainty. In typical engineering design problems, when knowing information such as the manufacturing tolerance specifications, operating ranges, nominal operating points, and historical data, it is not too difficult to determine the bounds of the uncertain parameters [19]. Hence, robust optimization with interval uncertainty has been applied to a range of engineering design and systems control problems [20].

When using worst-case analysis to solve the robust optimization problem with interval uncertainty, it almost always involves a min–max problem of a nested double loop optimization structure [21,22]. Since the inner optimization is performed iteratively for every candidate evaluation in the outer optimization, the computational efficiency is one of the most critical concerns [23,24]. For inner optimization, the most common way to improve computational efficiency is to find a surrogate model that approximates the original functions. Chen et al. [25] and Zhou et al. [26] used the Taylor series expansion to analyze the robustness of each candidate. Rehman et al. [21] proposed an efficient global optimization method based on Kriging interpolation to reduce the function evaluations. In [27], a modified Benders decomposition method was applied to a variety of robust optimization problems. Note that if the approximate accuracy of the surrogate model is low, then the robustness evaluated by inner optimization may be inaccurate, leading to the incorrect result that the solution fails to meet the robustness requirement. In order to reduce the computational complexity while improving the calculation accuracy for robust optimization, it is necessary to find an efficient inner optimization method. The function of outer optimization is to find promising candidate solutions and choose the best decision variables. The deterministic optimization methods based on gradient information [28] offer a fast convergence rate but the solution often converges to a locally optimal point. Stochastic optimization methods, such as genetic algorithm (GA) [29] and particle swarm optimization (PSO) [30], are well suited for global search [31,32], which can increase the probability of finding the global optimum through randomly searching for candidates. State transition algorithm (STA) [33,34] is a stochastic optimization method consisting of four state transformation operators and each operator has a special searching function. The STA method has been shown to be capable of both local and global search with stable convergence rate [35,36]. It thus appears that there are merits to investigating STA for outer optimization of the robust problem.

The issues of computational efficiency and convergence rate have been an obstacle to a robust optimization method implementing efficient and accurate search. To overcome these problems, a hybrid state transition algorithm for a robust optimization problem is proposed in this paper. The novelty and contribution of this method are three-fold: (1) to reduce the computational cost, the second-order Taylor series surrogate model is used to simplify the calculation of objective function values in the inner optimization, and a low-computational-cost method sequential quadratic programming is used in the outer optimization; (2) to strengthen the convergence, the outer optimization is conducted by the cooperation of state transition algorithm and sequential quadratic programming, which not only avoids the premature convergence but also improves the solution precision; and (3) to balance robustness and optimality, a selection mechanism is proposed, which evaluates the candidates based on their feasibility, robustness, and optimality [37]. By comparing the experimental results obtained by the hybrid state transition algorithm and other robust optimization algorithms (robust genetic algorithm, Kriging metamodel-assisted robust optimization method, etc.), the results denote that the method proposed in this paper has better performance with respect to both accuracy and efficiency.

The remainder of this paper is organized as follows: Section 2 introduces the background to the optimization problems, including the formulation of the robust optimization problem with interval uncertainty, the Taylor series surrogate model, and the state transition algorithm. The hybrid state transition algorithm is derived in Section 3. Section 4

analyzes the robust performance of the proposed method based on a comparative study of eight number of algorithms on seven examples. Section 5 concludes this paper and discusses the directions of future research.

2. Background and Terminology

2.1. Robust Optimization Problem

In general, the deterministic optimization problems can be defined as follows:

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & f(x, p) \\ \text{s.t.} \quad & g_i(x, p) \leq 0, i = 1, \dots, n, \\ & x_l \leq x \leq x_u, \end{aligned} \quad (1)$$

where $f(\cdot)$ is the objective function and $g(\cdot)$ is the constraints function, and n is the number of constraints. The vector x is the decision variable whose lower and upper bounds are x_l and x_u , respectively, and p is the parameter of the problem.

In optimization problems, uncertainties can be involved in both decision variables and parameters. Thus, the formulation of the optimization problem under interval uncertainty is given as

$$\begin{aligned} \min_{[x] \in \mathcal{X}} \quad & f([x], [p]) \\ \text{s.t.} \quad & g_i([x], [p]) \leq 0, i = 1, \dots, n, \\ & x_l \leq [x] \leq x_u, \end{aligned} \quad (2)$$

where $[x]$ and $[p]$ are interval numbers corresponding to the uncertain decision variables and uncertain parameters. They can be expressed as

$$\begin{aligned} [x] &= [x_c + \underline{\Delta x}, x_c + \overline{\Delta x}] \\ [p] &= [p_c + \underline{\Delta p}, p_c + \overline{\Delta p}], \end{aligned} \quad (3)$$

where x_c and p_c are the nominal value of x and p , respectively, with $\underline{\Delta x}$ and $\overline{\Delta x}$ being the lower and upper bounds of decision variable (x) variation, and $\underline{\Delta p}$ and $\overline{\Delta p}$ being the lower and upper bounds of parameter (p) variation. For simplicity, it is usually assumed that the nominal value is the central value of the variation range, which implies that $\underline{\Delta x} = -\overline{\Delta x}$ and $\underline{\Delta p} = -\overline{\Delta p}$.

For evaluating the solutions of the robust optimization problem in (2), three indexes are introduced:

1. Objective robustness: This index, denoted as η_f , is a measure of the sensitivity for the objective function to uncertainties. When decision variables and/or parameters fluctuate in their uncertain intervals, the objective function variations should still within an acceptable range. In engineering problems, the acceptable range of objective function is usually defined by decision makers according to design requirements.
2. Feasibility robustness: This index, denoted as η_g , is a measure of the sensitivity for the constraints to uncertainties. When decision variables and/or parameters fluctuate in their uncertain intervals, the constraints still should be satisfied.
3. Optimality: This index, represented as f , is the objective function value. For a deterministic optimization problem, the optimum should be the solution with best objective value.

Based on the above indexes, the optimization problem in (2) can be reformulated as follows:

$$\begin{aligned} \min_{x_c \in \mathcal{X}} \quad & f(x_c, p_c) \\ \text{s.t.} \quad & g_i(x_c, p_c) \leq 0, i = 1, \dots, n, \\ & \eta_f - \Delta f_0 \leq 0 \end{aligned}$$

$$\begin{aligned}
 & \eta_g \leq 0 \\
 \text{where } & \eta_f = \max_{x \in [x], p \in [p]} |f(x, p) - f(x_c, p_c)| \\
 & \eta_g = \max\{ \max_{x \in [x], p \in [p]} g_i(x, p), i = 1, \dots, n \} \\
 & [x] = [x_c + \underline{\Delta x}, x_c + \overline{\Delta x}] \\
 & [p] = [p_c + \underline{\Delta p}, p_c + \overline{\Delta p}] \\
 & x_l \leq [x] \leq x_u,
 \end{aligned} \tag{4}$$

where Δf_0 means the acceptable variation range of the objective function.

The above formulation about the robust optimization problem contains a nested double-loop optimization structure. The outer optimization functions to find promising nominal values of decision variables and the inner optimization is used to verify the robustness of candidate solutions. The nested double-loop optimization structure incurs high computational costs. Thus, this paper proposes a robust optimization method that solves the computationally costly optimization problem using (i) the state transition algorithm with sequential quadratic programming, called the hybrid state transition algorithm (H-STA), to solve the outer optimization problem, and (ii) the second-order Taylor series expansion to estimate the robustness indexes for inner optimization. In the sections that follow, the two techniques of the proposed RO method are further discussed.

2.2. Taylor Series Surrogate Model

In general, the inner optimization is performed iteratively for every candidate solution in the outer optimization. To reduce the computational cost, the procedure for inner optimization should be as simple as possible. In the proposed RO method, a second-order Taylor series surrogate model is used to approximate the objective function and calculate the extreme points for inner optimization.

Based on the Taylor’s theorem, a multivariable function $f(x, p)$ can be expanded by the Taylor series around $(x, p) = (x_c, p_c)$

$$\begin{aligned}
 f(x, p) = & f(x_c, p_c) \\
 & + \frac{\partial f(x_c, p_c)}{\partial x} (x - x_c) + \frac{\partial f(x_c, p_c)}{\partial p} (p - p_c) \\
 & + \frac{1}{2!} \frac{\partial^2 f(x_c, p_c)}{\partial x^2} (x - x_c)^2 \\
 & + \frac{\partial^2 f(x_c, p_c)}{\partial x \partial p} (x - x_c)(p - p_c) \\
 & + \frac{1}{2!} \frac{\partial^2 f(x_c, p_c)}{\partial p^2} (p - p_c)^2 + \dots
 \end{aligned} \tag{5}$$

Let

$$x - x_c = \Delta x, \quad p - p_c = \Delta p, \tag{6}$$

then (5) can be written as

$$\begin{aligned}
 f(x, p) = & f(x_c, p_c) \\
 & + \frac{\partial f(x_c, p_c)}{\partial x} \Delta x + \frac{\partial f(x_c, p_c)}{\partial p} \Delta p \\
 & + \frac{1}{2!} \frac{\partial^2 f(x_c, p_c)}{\partial x^2} \Delta x^2 \\
 & + \frac{\partial^2 f(x_c, p_c)}{\partial x \partial p} \Delta x \Delta p
 \end{aligned}$$

$$+ \frac{1}{2!} \frac{\partial^2 f(x_c, p_c)}{\partial p^2} \Delta p^2 + \dots \tag{7}$$

The remainder term of the second-order Taylor expansion of $f(x, p)$ around point (x_c, p_c) can be written as follows:

$$R(\Delta x, \Delta p) = \frac{1}{3!} (\Delta x \frac{\partial}{\partial x} + \Delta p \frac{\partial}{\partial p})^3 f(x_c + \theta \Delta x, p_c + \theta \Delta p) \tag{8}$$

where $\theta \in (0, 1)$.

It is worth noting that $R(\Delta x, \Delta p)$ is a function of a cubic polynomial with respect to Δx and Δp , and the values of Δx and Δp are usually small. Meanwhile, the higher-order derivatives have relatively small values compared to the lower-order derivatives [38]. The above two points make $R(\Delta x, \Delta p)$ a tiny value and guarantee the accuracy of the second-order Taylor expansion alternative model.

For solving the maximization problem in (4), the second-order Taylor series is adopted and the inner optimization problem can be transformed to

$$\begin{aligned} \eta_f &\approx \max_{\Delta x, \Delta p} |f(x_c + \Delta x, p_c + \Delta p) - f(x_c, p_c)| \\ &= \max_{\Delta x, \Delta p} |a_1 \Delta x + b_1 \Delta p + c_1 \Delta x^2 + d_1 \Delta x \Delta p \\ &\quad + e_1 \Delta p^2|, \end{aligned} \tag{9}$$

where

$$\begin{aligned} a_1 &= \frac{\partial f(x_c, p_c)}{\partial x}, b_1 = \frac{\partial f(x_c, p_c)}{\partial p} \\ c_1 &= \frac{1}{2!} \frac{\partial^2 f(x_c, p_c)}{\partial x^2}, d_1 = \frac{\partial^2 f(x_c, p_c)}{\partial x \partial p} \\ e_1 &= \frac{1}{2!} \frac{\partial^2 f(x_c, p_c)}{\partial p^2} \\ \Delta x &\in [\underline{\Delta x}, \overline{\Delta x}], \Delta p \in [\underline{\Delta p}, \overline{\Delta p}]. \end{aligned} \tag{10}$$

Similarly, the feasibility robustness index can be transformed to

$$\begin{aligned} \eta_g &\approx \max \{ \max_{\Delta x, \Delta p} g_i(x_c + \Delta x, p_c + \Delta p) \} \\ &= \max \{ \max_{\Delta x, \Delta p} a_2^i \Delta x + b_2^i \Delta p + c_2^i \Delta x^2 + d_2^i \Delta x \Delta p \\ &\quad + e_2^i \Delta p^2 + h^i \}, \end{aligned} \tag{11}$$

where

$$\begin{aligned} a_2^i &= \frac{\partial g_i(x_c, p_c)}{\partial x}, b_2^i = \frac{\partial g_i(x_c, p_c)}{\partial p} \\ c_2^i &= \frac{1}{2!} \frac{\partial^2 g_i(x_c, p_c)}{\partial x^2}, d_2^i = \frac{\partial^2 g_i(x_c, p_c)}{\partial x \partial p} \\ e_2^i &= \frac{1}{2!} \frac{\partial^2 g_i(x_c, p_c)}{\partial p^2}, h^i = g_i(x_c, p_c) \\ \Delta x &\in [\underline{\Delta x}, \overline{\Delta x}], \Delta p \in [\underline{\Delta p}, \overline{\Delta p}]. \end{aligned} \tag{12}$$

In (9) and (11), the extreme points can be computed by the quadratic formula, and the maximum is calculated by backsubstituting into the original function in (4). With the

calculation of the quadratic function, the computational cost of the inner optimization problem can be reduced.

2.3. State Transition Algorithm

For outer optimization problems, most metaheuristic methods have competitive performance. The state transition algorithm (STA) [34,39] is an intelligent optimization method based on the control theory of state space representation. The unified form of the generation of solutions in the STA method can be described as follows:

$$\begin{cases} \mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{u}_k \\ y_{k+1} = f(\mathbf{x}_{k+1}) \end{cases} \quad (13)$$

where \mathbf{x}_k represents a state, corresponding to a candidate solution of the problem; \mathbf{u}_k is a function of historical states; A_k and B_k stand for state transition matrices; and y_k means the fitness value of the objective function f .

In STA method, there are four state transformation operators that generate candidate solutions:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \frac{1}{n \|\mathbf{x}_k\|_2} R_r \mathbf{x}_k, \quad (14)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \beta R_t \frac{\mathbf{x}_k - \mathbf{x}_{k-1}}{\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_2}, \quad (15)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma R_e \mathbf{x}_k, \quad (16)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \delta R_a \mathbf{x}_k, \quad (17)$$

where (14)–(17) are the rotation transformation, translation transformation, expansion transformation, and axesion transformation, respectively. The parameters α , β , γ , and δ represent the transformation factors. The parameters R_r , R_t , R_e , and R_a are the random matrix with specific elements. The rotation transformation is a local search operator and the translation transformation has a function of line search. The expansion transformation is used for global search and the accession transformation is designed to strengthen single-dimensional search ability.

For a given solution, the aforementioned state transformation operators are performed alternately to generate candidate solutions. In general, these four operators in the STA method can find promising candidate solutions and converge to the global optimum point. However, in a robust optimization problem, the solution requires not only to be a superior candidate in the deterministic condition but also to satisfy the robustness requirements. Thus, it is important to strengthen the search ability and design appropriate selection mechanism when solving a robust optimization problem. Since the STA method is a stochastic algorithm which does not use the gradient information, the local search ability is restricted and the precision of the solution still needs some improvements. In this paper, a hybrid state transition algorithm that combines the improved STA operator with a traditional local search procedure is proposed to address the robust optimization problem.

3. Hybrid State Transition Algorithm for Robust Optimization Problem

In the STA method, the expansion transformation, as the main global search operator, still requires further improvement to enlarge the range of global search. In addition, the local search direction of the STA method is stochastic, which may have a slow convergence rate; thus, sequential quadratic programming (SQP) is used to exploit the local area and improve the precision of solutions. In this paper, the hybrid state transition algorithm (H-STA) for robust optimization is proposed and it combines the improved STA and SQP to maximize their advantages of global optimization and minimize their disadvantages of premature convergence. Moreover, in order to balance the feasibility, robustness, and

optimality, an efficient selection mechanism is proposed to evaluate candidate solutions and select the best one as the final result.

3.1. Exploration Stage-Improved STA

The first stage of the H-STA method for robust optimization problems is to explore all search areas and find some promising candidates. In the basic state transition algorithm, there are four different transformation operators and the expansion transformation operator is the main global search operator. As shown in (16), the expansion transformation operator includes a Gaussian distribution matrix R_e , which means that it can generate elements between $[-\infty, +\infty]$ with probability. However, (16) also shows that the search range of expansion transformation not only depends on the expansion factor (γ) and the mean and standard deviation of R_e , but also relates to x_k . Thus, if the value of x_k is small, the search range will be small. For example, if we set the initial point as $[10, 10]$ and $[1, 1]$ separately, and the lower bound and upper bound of x are set to -10 and 10 , respectively, then the parameter setting of the expansion transformation operator is the same as in previous papers [39], which are $\gamma = 1$, and the mean and the standard deviation of R_e equal 0 and 1, respectively. In this study, we use the expansion transformation operator to generate 500 candidates, and if the candidate value is out of the range, a random value within the range will be selected as a substitute.

The performance of the expansion transformation is shown in Figure 1, with Figure 1a showing the expansion transformation operator that can generate a candidate in the search space with the initial point $[10, 10]$. However, when the initial point is set to $[1, 1]$, the search range of the expansion transformation operator becomes narrow (see Figure 1b). Thus, the global search ability of expansion transformation still requires further improvements.

One solution is to take into account the ranges of the decision variables in the expansion transformation:

$$x_{k+1} = x_k + \gamma R_e R_x, \tag{18}$$

where $R_x = (x_u - x_l)/2$ is the search radius of the decision variables, including the upper bound and lower bound of the decision variable.

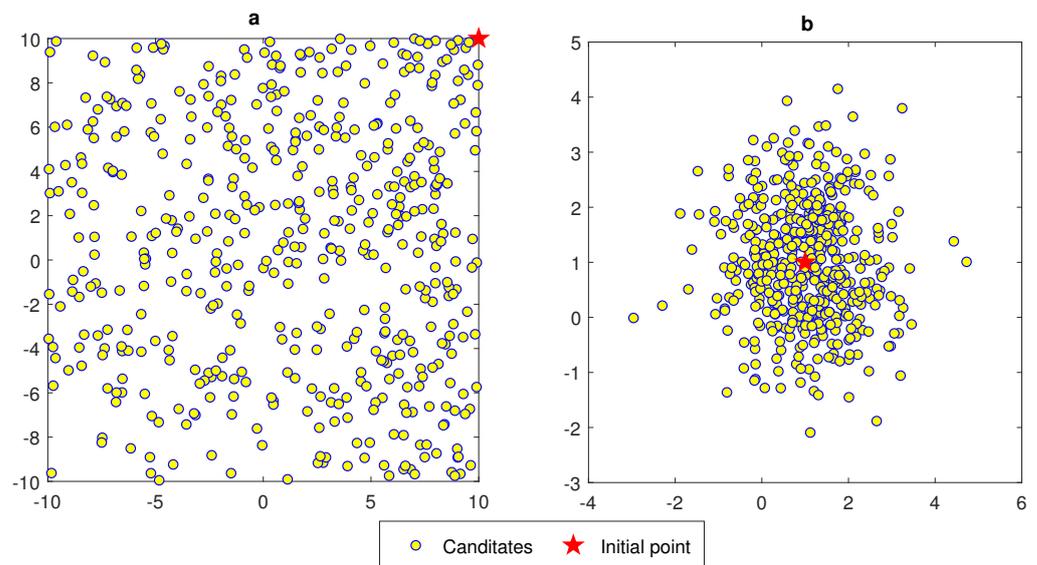


Figure 1. Performance of the original expansion transformation: (a) with initial point $[10, 10]$, and (b) with initial point $[1, 1]$.

The pseudocode of the new expansion transformation operator is shown in Algorithm 1. We use the same parameter settings and initial points to perform the new expansion transformation operator, and the results are shown in Figure 2. It is shown that no matter how the initial

point changes ([10, 10] in Figure 2a and [1, 1] in Figure 2b), the candidates generated by the new expansion transformation can always distribute throughout the entire search space.

The structure of the improved STA method is shown in Algorithm 2. Firstly, the parameters are predefined and the initial solution is generated randomly. The parameter SE is the search enforcement, which means that every transformation operator will generate SE candidate solutions. During the optimization process, the rotation factor α decreases exponentially from a maximum value (α_{\max}) to a minimum value (α_{\min}) with the base fc . If the rotation factor α is less than α_{\min} , it will return to the maximum value α_{\max} .

Algorithm 1 Pseudocode of improved STA method

Require:

$iter_{\max}$: maximum number of iterations

SE : search enforcement

$Best$: initial solution

Ensure:

$Best^*$: optimal solution

```

1: for  $k = 1$  to  $iter_{\max}$  do
2:   if  $\alpha < \alpha_{\min}$  then
3:      $\alpha \leftarrow \alpha_{\max}$ 
4:   end if
5:    $newBest \leftarrow$  new expansion(funcn,  $Best$ ,  $SE$ , ...)
6:   if  $Best \neq newBest$  then
7:      $newBest \leftarrow$  translation(funcn,  $Best$ ,  $newBest$ , ...)
8:   end if
9:    $Best = newBest$ 
10:   $newBest \leftarrow$  rotation(funcn,  $Best$ ,  $SE$ , ...)
11:  if  $Best \neq newBest$  then
12:     $newBest \leftarrow$  translation(funcn,  $Best$ ,  $newBest$ , ...)
13:  end if
14:   $Best = newBest$ 
15:   $newBest \leftarrow$  axesion(funcn,  $Best$ ,  $SE$ , ...)
16:  if  $Best \neq newBest$  then
17:     $newBest \leftarrow$  translation(funcn,  $Best$ ,  $newBest$ , ...)
18:  end if
19:   $Best = newBest$ 
20:   $\alpha \leftarrow \alpha \cdot fc^{-1}$ 
21: end for
22:  $Best^* \leftarrow Best$ 

```

Algorithm 2 Pseudocode of new expansion transformation operator

Require:

$oldBest$: the best solution in the last transformation

Ensure:

$Best^*$: optimal solution

```

1: State  $\leftarrow oldBest + \gamma R_e R_x$ 
2: if  $\exists i, j$ , let  $State(i, j) < x_l$  or  $State(i, j) > x_u$  then
3:    $State(i, j) \leftarrow x_l + (x_u - x_l) * rand$ 
4: end if
5:  $Best^* \leftarrow State$ 

```

Using the improved STA (I-STA) method, the four transformation operators can explore the whole space for neighborhoods that may contain the global optimal solution. Once the STA method converges to a promising candidate and stops at this point after several iterations, then the SQP method is used to improve the precision of the solutions.

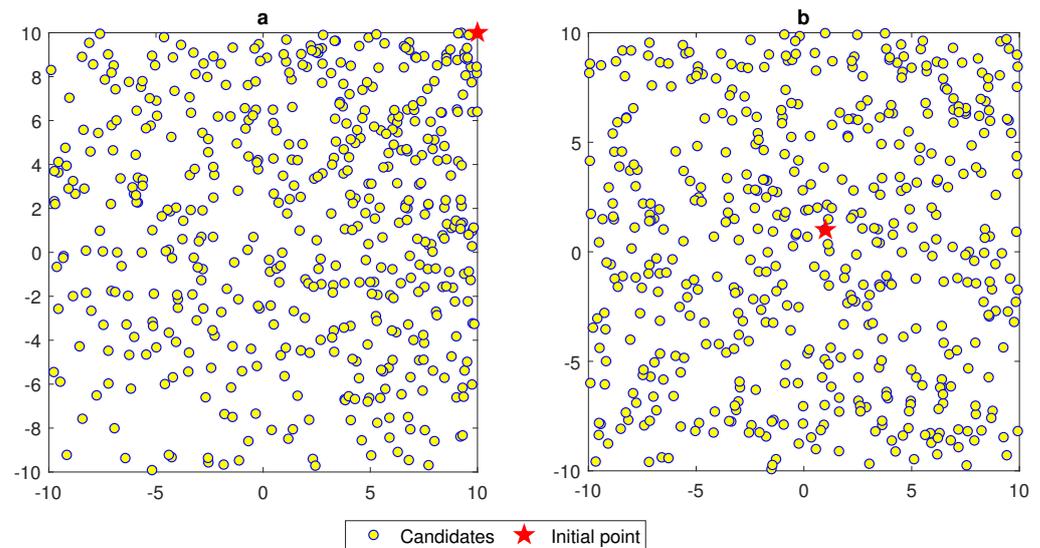


Figure 2. Performance of the new expansion transformation: (a) with initial point $[10, 10]$, and (b) with initial point $[1, 1]$.

3.2. Exploitation Stage-SQP

The second stage of the H-STA method for robust optimization problems needs to exploit the local area based on a promising initial point. Sequential quadratic programming is effective for solving a nonlinearly constrained optimization problem since the goal of SQP is to find a locally optimal solution to the problem, which means that the algorithm will continuously optimize the local area near the current solution in order to reduce the value of the objective function as much as possible. That ensures the accuracy of SQP. As an iterative procedure, the SQP method transforms a nonlinear optimization problem into a quadratic programming subproblem, and by solving that subproblem, the solution will converge to a local minimum. Thus, with the promising candidate found by the STA method and taking it as the initial point, the SQP method will exploit the neighborhood around that point and can improve the accuracy of the solution. The SQP structure [40] is shown in Algorithm 3.

Algorithm 3 Pseudocode of SQP method

Require:

k : number of iterations, and $k = 0$

$Best_0$: the initial solution (final solution obtained by STA)

Ensure:

$Best^*$: optimal solution

- 1: Approximate the problem with a linearly constrained quadratic programming at $Best_k$
 - 2: Solve for the optimal d_k
 - 3: **if** $d_k \approx 0$ **then**
 - 4: $Best^* = Best_k$
 - 5: **Break**
 - 6: **else**
 - 7: $Best_{k+1} = Best_k + d_k$
 - 8: $k = k + 1$
 - 9: **end if**
 - 10: **Go to Step 1**
-

We now explain the searching process of the SQP method; based on the Lagrangian function and quasi-Newton method, at each iteration, the outer optimization problem in (4) can be approximated by the following quadratic programming subproblem:

$$\begin{aligned}
 \min_{\mathbf{x}_c \in \mathcal{X}} \quad & f(\mathbf{x}_c^{(k+1)}) \\
 & = f(\mathbf{x}_c^{(k)} + d) \\
 & \cong f(\mathbf{x}_c^{(k)}) + \nabla f(\mathbf{x}_c^{(k)})^T d + \frac{1}{2} d^T H_k d \\
 \text{s.t.} \quad & g_i(\mathbf{x}_c^{(k+1)}) \\
 & = g_i(\mathbf{x}_c^{(k)} + d) \\
 & \cong g_i(\mathbf{x}_c^{(k)}) + \nabla g_i(\mathbf{x}_c^{(k)})^T d \leq 0, i = 1, \dots, n,
 \end{aligned} \tag{19}$$

where H_k is the Hessian matrix of $f(\mathbf{x})$ and d is the step length. For a given \mathbf{x}_c^k , functions $f(\mathbf{x}_c^{(k)})$ and $g_i(\mathbf{x}_c^{(k)})$ in (19) can be calculated. In order to find a desired step length, the subproblem can be rewritten as

$$\begin{aligned}
 \min_d \quad & \nabla f(\mathbf{x}_c^{(k)})^T d + \frac{1}{2} d^T H_k d \\
 \text{s.t.} \quad & \nabla g_i(\mathbf{x}_c^{(k)})^T d \leq 0, i = 1, \dots, n,
 \end{aligned} \tag{20}$$

By solving (20), the optimal step length is obtained. After the iterative calculation, the SQP method will stop when the optimal step length approaches 0.

Since the SQP method has a good exploitation ability and a low computation cost, it can effectively improve the accuracy of the solutions obtained by the I-STA method. Moreover, the I-STA method can generate a good initial solution for the SQP method, avoiding the possibility of falling into a local optimum due to a poor initial solution.

3.3. Selection Mechanism

Since the robust optimization problem requires to meet not only the constraints but also the robustness requirement, a selection mechanism is needed to balance the feasibility, robustness, and optimality of the solutions. To quantify the performance of the candidate solutions, two definitions are given as follows.

Definition 1 (Constraint violation). For the constraints $g_i(\mathbf{x}, \mathbf{p}) \leq 0$, the value $G(\mathbf{x}, \mathbf{p}) = \sum_{i=1}^n \max\{0, g_i(\mathbf{x}, \mathbf{p})\}$ indicates the relative constraint violation of solution \mathbf{x} on all constraints.

For a solution \mathbf{x} , if $G(\mathbf{x}, \mathbf{p}) = 0$, then \mathbf{x} is a feasible solution; otherwise, \mathbf{x} is an infeasible solution, and the larger the value, the stronger the constraint violation.

Definition 2 (Robustness violation). For the robustness indexes ($\eta_f(\mathbf{x}, \mathbf{p})$ and $\eta_g(\mathbf{x}, \mathbf{p})$), the value $R(\mathbf{x}, \mathbf{p}) = \max\{0, \eta_f(\mathbf{x}, \mathbf{p}) - \Delta f_0\} + \max\{0, \eta_g(\mathbf{x}, \mathbf{p})\}$ provides a reference showing the relative robustness violation of solution \mathbf{x} against the robustness indexes. For a solution \mathbf{x} , if $R(\mathbf{x}, \mathbf{p}) = 0$, then \mathbf{x} is a robust solution; otherwise, \mathbf{x} is a nonrobust solution, and the larger the value, the stronger the robustness violation.

The selection steps for two solutions of the robust optimization problem are as follows:

1. The feasible solution is always preferred to the infeasible solution;
2. If two solutions are both infeasible, then the one having a smaller constraint violation value is preferred;
3. If two solutions are both feasible, then the robustness indexes will be calculated by inner optimization, and the following criteria will be considered:
 - (a) The robust solution is always preferred to the nonrobust solution;

- (b) If two solutions are both nonrobust, then the one having a smaller robustness violation value is preferred;
- (c) If two solutions are both robust, then the one having better objective function value is preferred.

The code structure of the selection mechanism is shown in Algorithm 4. After generating candidate solutions in the outer optimization process, the inner optimization is performed only when the feasible solution exists. Note that the infeasible solution can never satisfy the requirement of feasible robustness.

Algorithm 4 Pseudocode of the selection mechanism

Require:

State: candidate solutions generated by improved STA or SQP (outer optimization)
G(State): constraint violation of the *State*
R(State): robustness violation of the *State*
f(State): objective function value of the *State*

Ensure:

*Best**: optimal solution

```

1:  $num_g = find(G(State) = 0)$ 
2: if  $length(num_g) \geq 1$  then
3:   Calculate the robustness indexes (inner optimization)
4:    $num_r = find(R(State) = 0)$ 
5:   if  $length(num_r) \geq 1$  then
6:      $Best* \leftarrow \min f(State)$ 
7:   else
8:      $Best* \leftarrow \min R(State)$ 
9:   end if
10: else
11:    $Best* \leftarrow \min G(State)$ 
12: end if

```

3.4. Framework of the H-STA for Solving Robust Optimization Problem

The framework of the H-STA method for robust optimization problem with interval uncertainty is shown in Figure 3.

- Step 1** (Initialization): The first step includes the generation of the initial solution and initialization of the parameters.
- Step 2** (Outer optimization - I-STA): Based on the initial solution, the I-STA method generates candidates using four transformation operators.
- Step 3** (Selection mechanism): The proposed selection mechanism is used to select the best solution among many candidates. If there is a feasible solution in the candidate solutions, the inner optimization is conducted based on Taylor series surrogate model.
- Step 4** (Switching criterion): The switching criterion is used to determine whether the I-STA method should be replaced by the SQP method. The I-STA method is a stochastic optimization algorithm, and in the later iterations, the rate of convergence is slow. Nevertheless, the SQP method offers fast convergence rate leading to the optimal solution based on a good initial solution. Thus, if the objective value in the improved STA method changes very slowly, the SQP method will be performed to improve the rate of convergence. In this paper, the switching index is defined as follows:

$$\lambda_k = \left| \frac{f_k - f_{k-1}}{f_k} \right| \quad (21)$$

where f_k and f_{k-1} are the objective function values of x_k and x_{k-1} . If λ_k is less than a threshold value (λ), the rate of convergence for the I-STA method is considered slow and the SQP method is carried out.

- Step 5** (Outer optimization—SQP): Considering the final solution of the I-STA method as an initial point, the SQP method is performed to improve the precision of the solution.
- Step 6** (Selection mechanism): The proposed selection mechanism is used to compare the final solution and the initial solution of the SQP method.
- Step 7** (Stopping criterion): We use the maximum number of iteration as the stopping criterion. It is worth noting that the number of iterations in the SQP method is also taken into account when calculating the total number of iterations.

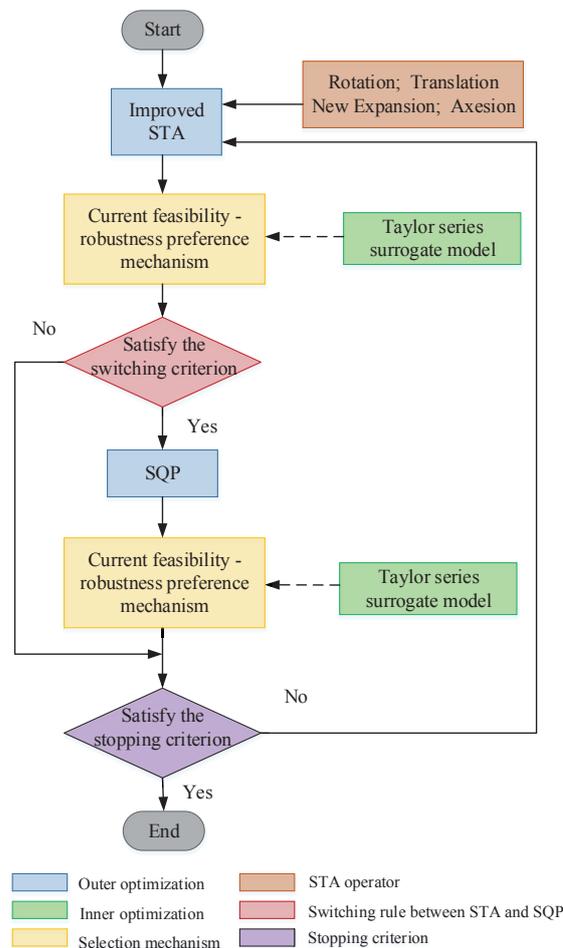


Figure 3. Framework of the hybrid state transition algorithm for robust optimization problem with interval uncertainty.

4. Verification Examples

In this section, the proposed method is applied to seven optimization examples with interval uncertainty. Table 1 gives the uncertainty occurrences in each example. To demonstrate the effectiveness of the H-STA method for robust optimization (H-STA-RO), the following methods are used for comparison: the I-STA method for robust optimization problems (I-STA-RO), the basic STA method for robust optimization problems (STA-RO), the SQP method for robust optimization problems (SQP-RO), and five well-known robust optimizers (the robust optimization method with Chebyshev surrogate models (I-RO) [22], the robust genetic algorithm (GA-RO), the Kriging metamodel-assisted robust optimization method (IK-GA-RO) [41], the robust optimization method based on Benders decomposition (BD-RO) [27], and the robust optimization method using differential evolution and sequential quadratic programming (DE-SQP-RO) [42]).

Table 1. Uncertainty occurrences in each example.

Uncertainty Occurrences	Examples 1, 2, and 3	Example 4	Examples 5 and 6	Examples 7
Decision variables x	✓	✓	✓	
Parameters p		✓		✓

The complexity of H-STA-RO can be represented by the value of function evaluations (FE), which is calculated as

$$FE = SE \times iter_{STA} \times N_{STA} + FE_{SQP} + N_R \times (R_f + R_g), \quad (22)$$

where SE is the search enforcement; $iter_{STA}$ and N_{STA} are the number of generations and transformation operators used in the I-STA method, respectively; the parameter FE_{SQP} is the total function evaluation value in the SQP method. Since the inner optimization is only performed when the feasible solution is found, N_R means the number of times that inner optimization performed, and R_f and R_g present the average value of function evaluations for candidates to obtain their objective robustness index and feasibility robustness index, respectively.

In order to evaluate the performance of different methods, all results are obtained after 20 runs under MATLAB R2016a, Windows 10 machine with 2.40 GHz Intel core i5 and 16.0 GB RAM. The SQP method and the genetic algorithm method are performed by using “fmincon” and “ga” function, respectively, and all the parameters are set by the default values. The parameters included in the H-STA-RO method are selected empirically based on numerous experiments and application cases. In the standard continuous STA, the parameter settings are given as follows: $\alpha_{\min} = 10^{-4}$, $\beta = 1$, $\gamma = 1$, $\delta = 1$, $SE = 30$, and $fc = 2$. Many numerical experiments and engineering applications have shown the effectiveness of the above parameter settings [34,35,39]. In this paper, to obtain better results for different problems, the parameters of α_{\max} , λ , and $iter_{\max}$ are fine-tuned based on the following guidelines:

- The rotation factor α , which controls the search range of the rotation transformation, is bounded as $\alpha_{\min} \leq \alpha \leq \alpha_{\max}$. A larger value of α allows more explorations of the local search space, and a smaller value of α can refine the quality of solutions. The value of α_{\max} is typically set as 1 based on the previous study. However, for the problem in which the ranges of decision variables are less than 1, it is useless to search in a hypersphere with a radius equal to 1. Thus, the parameter α_{\max} in Example 4 ($0 \leq x_1, x_2, x_3, x_4 \leq 1$) is adjusted according to the statistical analysis. As shown in Figure 4, by performing 20 trails in each test, we compare the average iterative results with different α_{\max} value. Given the same initial solution, the iterative curve with $\alpha_{\max} = 0.1$ has fastest rate of convergence to search better solutions. Therefore, α_{\max} in Example 4 is set to 0.1.
- The threshold value of switching index λ controls the frequency of the switching between two search operators. If one operator is trapped in a slow convergence, another search operator is taken into consideration. As shown in Figure 5, a larger value of λ can increase the switching frequency but it may give a low-quality solution under the SQP method (e.g., $\lambda = 10^{-2}$). A smaller value of λ may cause slow convergence (e.g., $\lambda = 10^{-5}$). In this paper, λ , as the threshold value of the relative difference between two objective values, is adjusted between $[10^{-4}, 10^{-3}]$.
- The maximum number of iterations $iter_{\max}$ depends on the complexity of the problem. For the two engineering optimization problems considered in this section, a choice of 80 to 100 iterations may be sufficient. For the four numerical problems, $iter_{\max}$ is set to 40 to 60. Take Example 4 for example (Figure 5), the red dotted line nearly

has no update from 45 generation, thus $iter_{max}$ is set to 60 so that the convergence performance of the H-STA method is guaranteed.

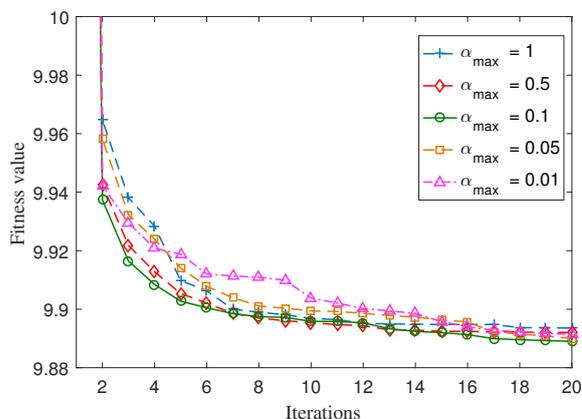


Figure 4. Iterative curve with different α_{max} of Example 4.

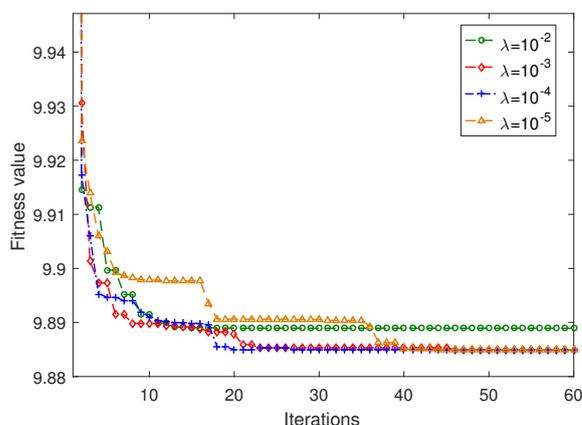


Figure 5. Iterative curve with different λ of Example 4.

Table 2 summarizes the parameter settings of the H-STA method used in the study.

Table 2. Parameter settings of the H-STA method used in the examples.

Parameters	Examples 1, 2, 3	Example 4	Examples 5, 6, and 8
α_{max}	1	0.1	1
λ	10^{-3}	10^{-4}	10^{-3}
$iter_{max}$	60	60	80

4.1. Example 1

This unconstrained optimization problem is used to verify the accuracy of the inner optimization method. The uncertain problem is given by as follows:

$$\begin{aligned}
 \min_{x_{1c}, x_{2c}} \quad & f(x_{1c}, x_{2c}) \\
 & = -x_{2c} - (x_{1c} + 0.25)^2 + (x_{1c} + 0.25)^3 \\
 & \quad + (x_{1c} + 0.25)^4 + 4 \\
 \text{s.t.} \quad & -3 \leq [x_1], [x_2] \leq 3 \\
 \text{with} \quad & [x_1] = [x_{1c} - 0.1, x_{1c} + 0.1] \\
 & [x_2] = [x_{2c} - 0.1, x_{2c} + 0.1].
 \end{aligned} \tag{23}$$

With uncertainties existing in the decision variables, the optimal solution should be the minimum of its objective upper bound, which can be represented as

$$\min_{x_{1c}, x_{2c}} f_u(x_{1c}, x_{2c})$$

$$\text{where } f_u = \max_{x_1 \in [x_1], x_2 \in [x_2]} f(x_1, x_2). \tag{24}$$

The upper bound of the objective function f_u is calculated by inner optimization. Table 3 shows the best results obtained by three different robust optimization methods from 20 trails. The interval robust optimization (I-RO) method [22] uses the multi-island genetic algorithm (MIGA) as the outer optimization method, and its inner optimization is based on the second-order Chebyshev surrogate model. The linearization robust optimization (L-RO) process [22] replaces the inner optimization method of I-RO with the first-order Taylor series surrogate model. In the H-STA-RO method, the inner optimization method is based on the second-order Taylor series surrogate model.

Table 3. Performance comparison of Example 1.

	I-RO (Second-Order Chebyshev) [22]	L-RO (First-Order Taylor Series) [22]	H-STA-RO (Second-Order Taylor Series)
x_1	−1.42108	−1.42542	−1.42046
x_2	2.9000	2.9000	2.9000
f_u	0.1410	0.1030	0.1405
R	0.1410	0.1447	0.1405

Note: R is the reference value of objective upper bound obtained from the Monte Carlo simulation run.

To obtain the accurate value of the objective upper bound, we used the Monte Carlo method in the uncertain range around the design point (using 2×10^8 samples), which can be used as the reference value (R) for assessing the accuracy of inner optimization method.

The results show that the values of f_u based on the second-order Taylor series model and the second-order Chebyshev model are closer to the reference value (R). Figure 6 illustrates the optimization results of these three methods. Figure 6a shows the contour lines of the objective function values, and Figure 6b is the results of the Monte Carlo test with 200 samples. We observe that the H-STA-RO method can find the global optimum and its inner optimization method provides sufficient accuracy to evaluate the robustness index.

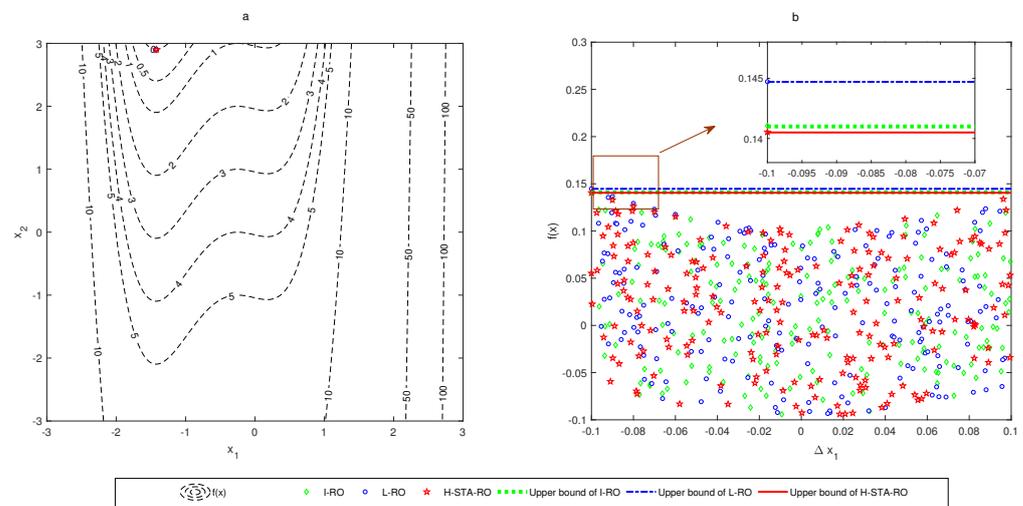


Figure 6. Robust optimization results of three methods for Example 1: (a) contour lines of the objective function values; (b) Monte Carlo test results of I-RO, L-RO, and H-STA-RO.

4.2. Example 2

This example is a nonlinear numerical problem with uncertainty in decision variables [42]. The deterministic version of this example is a classical multimodal function called “peaks function”. In the deterministic problem, the optimal solution is $x = [0.2283, -1.6255]$ and the objective function value is $f = -6.5511$. When the decision variable x_1 is subject to interval uncertainty, the robust optimization problem takes the form:

$$\begin{aligned}
 \min_{x_1, x_2} \quad & f(x_1, x_2) \\
 & = 3(1 - x_1)^2 e^{-(x_1^2 - (x_2 + 1)^2)} \\
 & \quad - 10\left(\frac{x_1}{5} - x_1^3 - x_2^5\right) e^{-x_1^2 - x_2^2} - \frac{1}{3} e^{-(x_1 + 1)^2 - x_2^2} \\
 \text{s.t.} \quad & g_1 = 2x_1^2 - x_2^2 \leq 0 \\
 & g_2 = 8.5x_1 + 1.2x_2 - 0.1 \leq 0 \\
 & -3 \leq [x_1], x_2 \leq 3 \\
 \text{with} \quad & \Delta f_0 = 0.02 \\
 & [x_1] = [x_1 - 0.05, x_1 + 0.05].
 \end{aligned} \tag{25}$$

Based on the proposed selection mechanism, shown in Table 4 are the optimal results obtained by the DE-SQP-RO method, the GA-RO method, the STA-RO method, the I-STA-RO method, and the H-STA-RO method. Since this example is a classical multimodal function, it has a known local optimum and global optimum. To verify the global search ability of the I-STA method, the success rate p_s of removing the local optimum (the percentage of successful runs in total runs) is analyzed.

In Table 4, the decision variable, the constraint function value, and the robustness violation value all correspond to the results with the best objective function value in 20 runs. The values for the success rate p_s and the robust rate p_r (the percentage of robust runs in total runs) are obtained by statistical analysis. The value for FE is the average value of the function evaluations and the standard deviation is also presented. The value of T represents the average runtime (in seconds). From Table 4, we observe the following:

(1) The success rate of removing the local optimum in the STA-RO method is 60%, and there are five results that fall into the local optimum $[-0.2606, 0.4667]$ with the objective function value 0.7881. In the I-STA-RO method, however, all the tests can find the results that are close to the global optimum. Thus, the new expansion transformation operator offers a better global search ability.

(2) The robust rates of the STA-RO method, the I-STA-RO method, and the H-STA-RO method are all 100% (with the proposed selection mechanism), whereas only 95% and 75% of the results in the GA-RO method and the SQP-RO method (without the proposed selection mechanism) are robust, which demonstrates that the proposed selection mechanism can obtain a robust solution with higher probability.

(3) In the DE-SQP-RO method [42], although its objective function value is better than that of the H-STA-RO method, the result cannot satisfy the requirements of the robustness according to the value of R .

(4) The average function evaluations and runtime of the H-STA-RO method are smaller than that of others (except the SQP-RO method); this is because on the one hand, the proposed selection mechanism can avoid useless calculations in inner optimization, and on the other hand, the SQP method can improve the efficiency of the search process. The proposed selection mechanism may also cause the variation of inner optimization computational cost, leading to a high standard deviation of FE .

To verify the robustness of the obtained solution in the H-STA-RO method, the Monte Carlo simulation is conducted. By using 200 samples around the nominal value within the uncertainty interval, the objective robustness index and the feasibility robustness index are calculated. In Figure 7, when decision variable x_1 is perturbed, the variations of

the objective function (shown in Figure 7a) and the constraints (shown in Figure 7b) for the solution obtained by the H-STA-RO method are always within the acceptance range, whereas the deterministic solution cannot satisfy the robustness requirements.

Table 4. Performance comparison of Example 2.

	DE-SQP-RO [42]	GA-RO	SQP-RO	STA-RO	I-STA-RO	H-STA-RO
x_1	0.1944	0.1945	0.1947	0.1942	0.1943	0.1945
x_2	−1.8414	−1.8410	−1.8395	−1.8438	−1.8427	−1.8414
f	−5.9559	−5.9579	−5.9655	−5.9435	−5.9491	−5.9557
g_1	−3.3152	−3.3137	−3.3080	−3.3242	−3.3202	−3.3153
g_2	−0.6573	−0.6559	−0.6526	−0.6616	−0.6596	−0.6568
R	5.00×10^{-5}	9.87×10^{-6}	5.00×10^{-6}	0	0	0
p_s	–	90%	35%	75%	100%	100%
p_r	–	95%	75%	100%	100%	100%
FE	$1,941,630 \pm -$	$27,974 \pm 59.00$	139.9000 ± 62.27	$38,053 \pm 3.44 \times 10^3$	$36,650 \pm 1.28 \times 10^3$	$17,456 \pm 3.53 \times 10^3$
T	–	0.106	0.026	0.149	0.153	0.145

Note: R is the reference value of robustness violation obtained from the Monte Carlo simulation, and – denotes data not available.

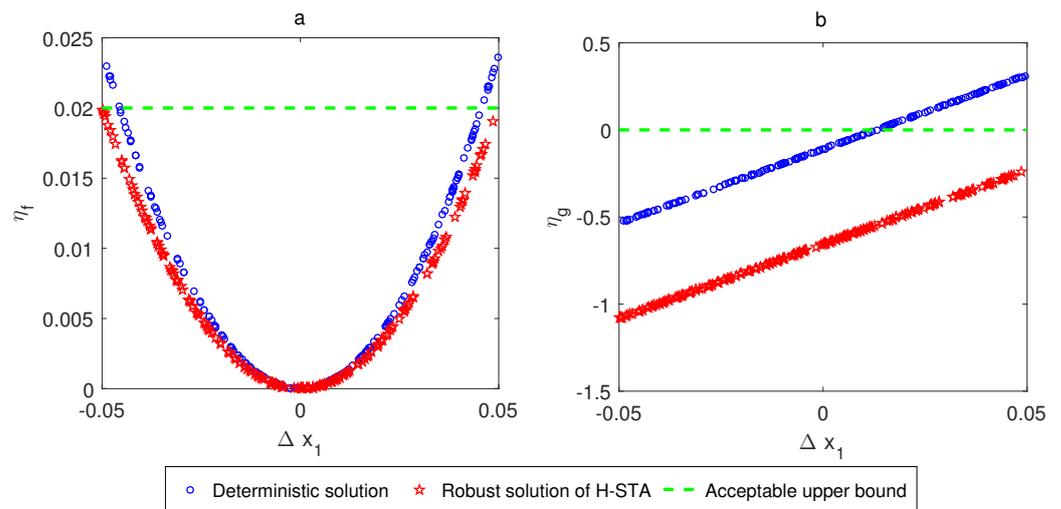


Figure 7. Robustness verification of the deterministic and robust solution for Example 2: (a) objective robustness verification, and (b) feasibility robustness verification.

4.3. Example 3

This constrained nonlinear problem originated from [41]. The problem formulation is:

$$\begin{aligned}
 & \min_{x_{1c}, x_{2c}} f(x_{1c}, x_{2c}) \\
 & = x_{1c}^3 \sin(x_{1c} + 4) + 10x_{1c}^2 + 22x_{1c} + 5x_{1c}x_{2c} \\
 & \quad + 2x_{2c}^2 + 3x_{2c} + 12 \\
 & \text{s.t. } g_1 = x_{1c}^2 + 3x_{1c} - x_{1c} \sin x_{1c} + x_{2c} - 2.75 \leq 0 \\
 & \quad g_2 = -\log(0.1x_{1c} + 0.41) + x_{2c}e^{-x_{1c} + 3x_{2c} - 4} \\
 & \quad \quad + x_{2c} - 3 \leq 0 \\
 & \quad -4 \leq [x_1] \leq 1, -1 \leq [x_2] \leq 1.5 \\
 & \text{with } \Delta f_0 = 2.5 \\
 & \quad [x_1] = [x_{1c} - 0.4, x_{1c} + 0.4]
 \end{aligned}$$

$$[x_2] = [x_{2c} - 0.4, x_{2c} + 0.4], \tag{26}$$

where the decision variables x_1 and x_2 have uncertainty $[-0.4, 0.4]$ around x_{1c} and x_{2c} . Without incurring uncertainties, the optimal solution of the deterministic problem is $x = [-1.8256, 0.7411]$ with $f = -3.2871$. The robust results of the STA-RO method, the I-STA-RO method, the SQP-RO method, the H-STA-RO method, the GA-RO method, and the IK-GA-RO method are shown in Table 5.

Table 5. Performance comparison of Example 3.

	IK-GA-RO [41]	GA-RO	SQP-RO	STA-RO	I-STA-RO	H-STA-RO
x_1	-1.447	-1.4409	-1.4404	-1.4394	-1.4399	-1.4405
x_2	0.267	0.3368	0.3370	0.3372	0.3371	0.3369
f	-1.567	-1.772776	-1.772834	-1.772766	-1.772770	-1.772771
g_1	-6.166	-6.0885	-6.0876	-6.0861	-6.0869	-6.0878
g_2	-1.360	-1.2670	-1.2670	-1.2671	-1.2671	-1.2670
R	0.1481	9.25×10^{-6}	9.58×10^{-5}	0	0	0
p_r	–	60%	65%	100%	100%	100%
FE	$18,558 \pm -$	$97,211 \pm 1.46 \times 10^3$	46.65 ± 19.0740	$38,685 \pm 7.44 \times 10^2$	$37,823 \pm 1.05 \times 10^3$	$18,896 \pm 1.26 \times 10^3$
T	–	0.220	0.025	0.348	0.339	0.084

Note: R is the reference value of robustness violation obtained from the Monte Carlo simulation, and – denotes data not available.

From Table 5, it is observed that the H-STA-RO method can find the optimal solution with 100% robustness, and its function evaluations and runtime are less than that of the GA-RO method, the STA-RO method, and the I-STA-RO method. Although the function calls of the IK-GA-RO method is smaller, the optimality and robustness of the optimum obtained by the IK-GA-RO method are inferior to that of the H-STA-RO method. Figure 8 shows the robustness indexes of the deterministic optimum and robust optimum obtained by the H-STA-RO method. The deterministic optimum violates the robust requirement at some points, but the robust optimum of the H-STA-RO method can remain stable within the uncertain range.

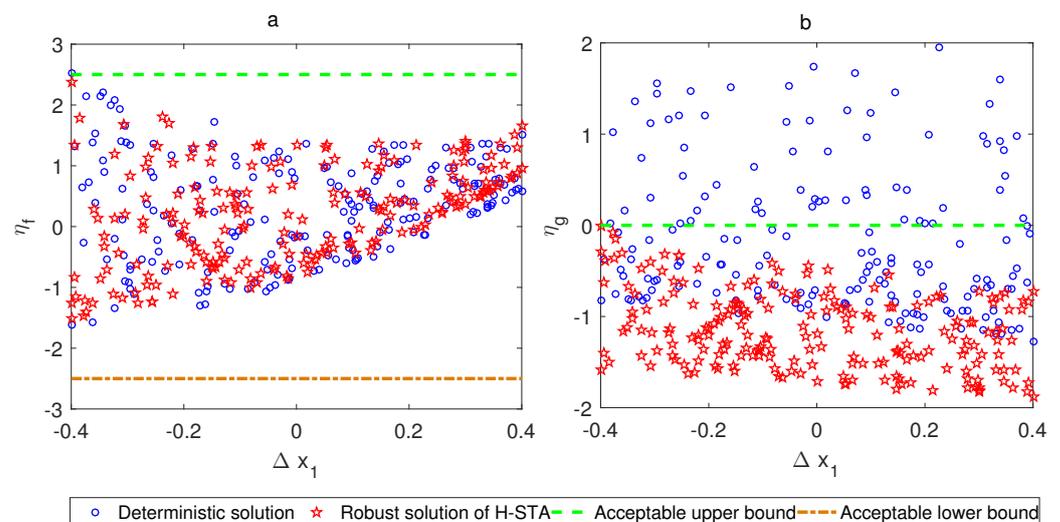


Figure 8. Robustness verification of the deterministic and robust solution for Example 3: (a) objective robustness verification; (b) feasibility robustness verification.

4.4. Example 4

This example [27,43] illustrates the solving of the robust optimization problem with uncertainty in both decision variables and parameters. The formulation of this problem is

$$\begin{aligned}
 & \min_{x_1, x_2, x_{3c}, x_4} && f(x_1, x_2, x_{3c}, x_4) \\
 & && = (x_1 - 0.6)^2 + (x_2 - 0.6)^2 - x_{3c}x_4 + 10 \\
 \text{s.t.} & && g_1 = p_{1c} + x_1 + x_2 \leq 0 \\
 & && g_2 = p_{2c} + x_{3c} + x_4 \leq 0 \\
 & && 0 \leq x_1, x_2, [x_3], x_4 \leq 1 \\
 \text{with} & && p_{1c} = -1, p_{2c} = -1 \\
 & && [x_3] = [x_{3c} - 0.1, x_{3c} + 0.1] \\
 & && [p_1] = [p_{1c} - 0.1, p_{1c} + 0.1] \\
 & && [p_2] = [p_{2c} - 0.1, p_{2c} + 0.1]. \tag{27}
 \end{aligned}$$

Without incurring the uncertainties, the deterministic optimal solution is $x = [0.5, 0.5, 0.5, 0.5]$ with $f = 9.770$. Table 6 shows the results for this example using robust optimization methods. These methods are the BD-RO method, the GA-RO method, the SQP-RO method, the STA-RO method, the I-STA-RO method, and the H-STA-RO method. Figure 9 shows the results of the Monte Carlo tests.

Table 6. Performance comparison of Example 4.

	BD-RO [27]	GA-RO	SQP-RO	STA-RO	I-STA-RO	H-STA-RO
x_1	0.45	0.4500	0.4275	0.4499	0.4550	0.4500
x_2	0.45	0.4500	0.4725	0.4496	0.4449	0.4500
x_3	0.4	0.4005	0.5181	0.3905	0.4000	0.4000
x_4	0.4	0.4004	0.3531	0.4093	0.3999	0.4000
f	9.8850	9.8846	9.8631	9.8853	9.8852	9.8850
g_1	-0.1000	-0.1000	-0.1000	-0.1005	-0.1002	-0.1000
g_2	-0.1000	-0.1990	-0.1288	-0.2002	-0.2001	-0.2000
R	0	9.9988×10^{-4}	0.0712	0	0	0
p_r	–	50%	40%	100%	100%	100%
FE	$21 \pm -$	$76,541 \pm 17.07$	178 ± 67.97	$55,573 \pm 1.94 \times 10^3$	$54,658 \pm 1.05 \times 10^3$	$23,644 \pm 3.75 \times 10^3$
T	–	0.153	0.0252	0.183	0.177	0.071

Note: R is the reference value of robustness violation obtained from the Monte Carlo simulation, and – denotes data not available.

From Table 6, it is observed that the objective function values of the I-STA-RO method and the STA-RO method are close to the optimal value, but it is hard to improve the precision. The SQP-RO method has a good rate of convergence but it cannot ensure the robustness of the solutions. Thus, the H-STA-RO method takes advantage of I-STA and SQP to obtain the robust solution and search the global optimum with less computational cost. For the GA-RO method, it has better objective function value but the computational efficiency and the robustness of the solution need to be improved. The BD-RO method can find the optimum with fewer function calls but its outer optimization approach is based on the gradient information; thus, its results are highly influenced by the initial point. Figure 9 also demonstrates that when there are uncertainties in both decision variable and parameters, the solution obtained by the H-STA-RO method can still satisfy the constraints.

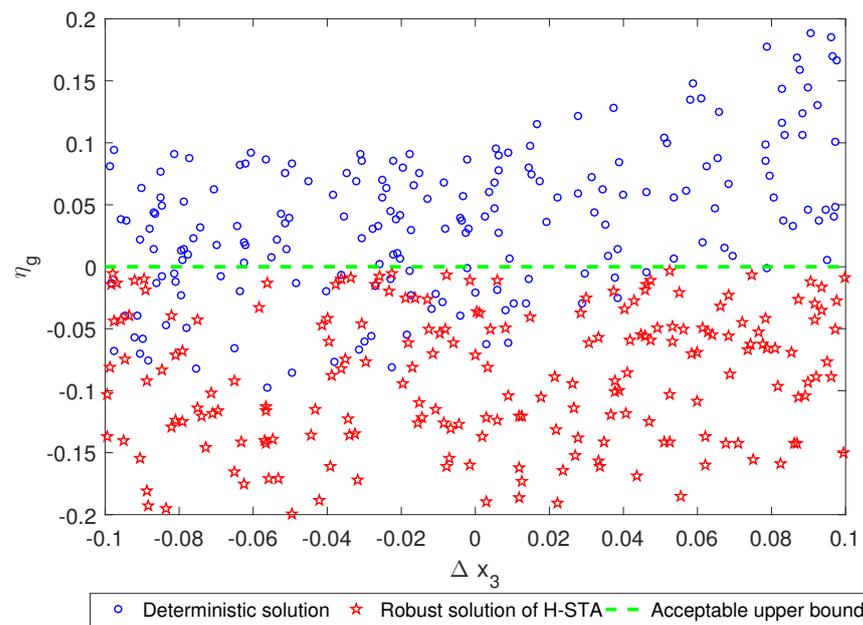


Figure 9. Robustness verification of the the deterministic and robust solution for Example 4.

4.5. Example 5: Welded Beam Design

The welded beam design (Figure 10) is a classical constrained optimization problem in engineering applications [36,44]. The design objective is to minimize the total cost of the welded beam f . The decision variables are the thickness of the weld x_1 , the length of the welded joint x_2 , the width of the beam x_3 , and the thickness of the beam x_4 . The decision variables must satisfy the constraints about the shear stress (τ), the bending stress (σ), the buckling load on the bar (P_c), the end deflection of the beam (δ), and the side constraints. The deterministic optimal result is $x = [0.2053, 3.2604, 9.0366, 0.2057]$ and $f = 1.6956$. When the decision variables are subject to uncertainties, the problem is modified as follows:

$$\begin{aligned}
 & \min_{x_1, x_2, x_{3c}, x_{4c}} f(x_1, x_2, x_{3c}, x_{4c}) \\
 & = 1.10471x_1^2x_2 + 0.04811x_{3c}x_{4c}(14 + x_2) \\
 \text{s.t.} \quad & g_1 = \tau - \tau_{max} \leq 0 \\
 & g_2 = \sigma - \sigma_{max} \leq 0 \\
 & g_3 = x_1 - x_{4c} \leq 0 \\
 & g_4 = 0.125 - x_1 \leq 0 \\
 & g_5 = \delta - \delta_{max} \leq 0 \\
 & g_6 = P - P_c \leq 0 \\
 & g_7 = \zeta_1x_1^2 + \zeta_2x_{3c}x_{4c}(14 + x_2) - 5 \leq 0 \\
 & 0.125 \leq x_1 \leq 2, 0.1 \leq x_2 \leq 10 \\
 & 0.1 \leq [x_3] \leq 10, 0.1 \leq [x_4] \leq 2 \\
 \text{where} \quad & \zeta_1 = 0.10471, \zeta_2 = 0.04811 \\
 & \tau = \sqrt{\tau_1^2 + 2\tau_1\tau_2(\frac{x_2}{2R}) + \tau_2^2}, \\
 & \tau_1 = \frac{P}{\sqrt{2}x_1x_2}, \tau_2 = \frac{MR}{J} \\
 & M = P(L + \frac{x_2}{2}), R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_{3c}}{2})^2}
 \end{aligned}$$

$$\begin{aligned}
 J &= 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_{3c}}{2}\right)^2\right]\right\} \\
 \sigma &= \frac{6PL}{x_{4c}x_{3c}^2}, \delta = \frac{6PL^3}{Ex_{3c}^3x_{4c}} \\
 P_c &= \frac{4.013E\sqrt{(x_{3c}^2x_{4c}^6)/36}}{L^2} \left(1 - \frac{x_{3c}}{2L}\sqrt{\frac{E}{4G}}\right) \\
 G &= 12 \times 10^6 \text{ psi}, E = 30 \times 10^6 \text{ psi}, \\
 P &= 6000 \text{ lb}, L = 14 \text{ in}, \tau_{max} = 13600 \text{ psi} \\
 \sigma_{max} &= 30000 \text{ psi}, \delta_{max} = 0.25 \text{ in} \\
 \text{with } \Delta f_0 &= 0.1 \\
 [x_3] &= [x_{3c} - 0.05, x_{3c} + 0.05] \\
 [x_4] &= [x_{4c} - 0.01, x_{4c} + 0.01]. \tag{28}
 \end{aligned}$$

Table 7 shows the results obtained by the DE-SQP-RO method [42], the GA-RO method, the SQP-RO method, the STA-RO method, the I-STA-RO method, and the H-STA-RO method.

Table 7. Performance comparison of Example 5.

	DE-SQP-RO	GA-RO	SQP-RO	STA-RO	I-STA-RO	H-STA-RO
x_1	0.2057	0.2064	0.2157	0.1849	0.1986	0.2050
x_2	7.0924	3.2452	3.0836	3.6657	3.3634	3.2686
x_3	9.0866	9.0751	9.0867	9.1159	9.1623	9.0774
x_4	0.2157	0.2163	0.2157	0.2157	0.2154	0.2162
f	2.3208	1.7809	1.7696	1.8096	1.7948	1.7818
g_1	-7.05×10^3	-51.0749	-51.2823	-69.9634	-50.2844	-51.1169
g_2	-1.70×10^3	-1.70×10^3	-1.70×10^3	-1.88×10^3	-2.12×10^3	-1.70×10^3
g_3	-0.0100	-0.0099	-4.35×10^{-9}	-0.0309	-0.0167	-0.0111
g_4	-3.0067	-3.3673	-3.3840	-3.3252	-3.3476	-3.3655
g_5	-0.0807	-0.0814	-0.0907	-0.0599	-0.0736	-0.0800
g_6	-0.2364	-0.2364	-0.2364	-0.2366	-0.2367	-0.2364
g_7	-940.3744	-988.3397	-943.0863	955.7257	-944.6512	-979.3933
R	0	9.99×10^{-4}	0.0712	0	0	0
p_r	–	50%	55%	100%	100%	100%
FE	$225,700 \pm -$	$139,081 \pm 1.20 \times 10^4$	202.5 ± 71.0534	$91,493 \pm 1.02 \times 10^4$	$77,743 \pm 1.11 \times 10^3$	$48,522 \pm 7.26 \times 10^2$
T	–	1.371	0.085	0.721	0.721	0.227

Note: R is the reference value of robustness violation obtained from the Monte Carlo simulation, and – denotes data not available.

From Table 7, the H-STA-RO method can generate competitive solutions when compared with other solutions. This is because (1) based on the I-STA method, the H-STA method can search the whole space and choose promising candidates, and (2) the SQP method can search the local area and improve the precision of the solution. The selection strategy and the inner optimization method proposed in this paper can guarantee the robustness of the best solutions. The findings presented in Figure 11 verify the robustness of the final solutions with respect to their objective function (Figure 11a) and constraints (Figure 11b).

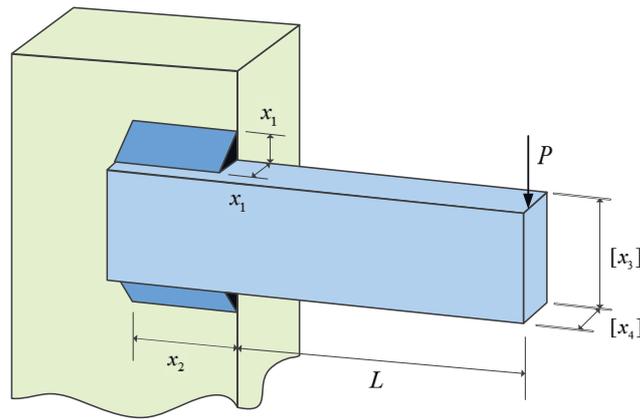


Figure 10. The welded beam.

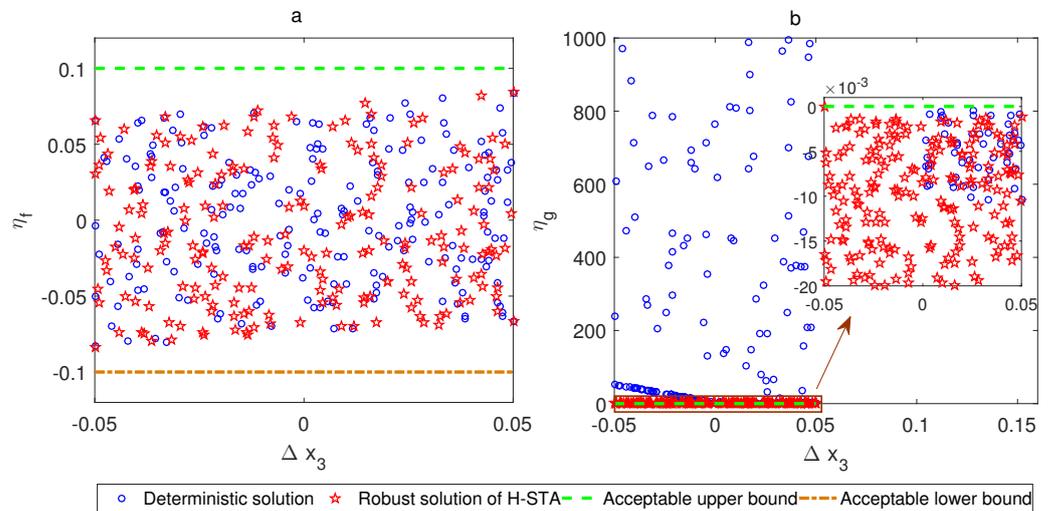


Figure 11. Robustness verification of the deterministic and robust solution for Example 5: (a) objective robustness verification; (b) feasibility robustness verification.

4.6. Example 6: Pressure Vessel Design

The optimal design of pressure vessel (see Figure 12) is a typical application example to illustrate the min–max problem. To minimize the total cost of the vessel f , there are four decision variables to optimize: x_1 (the thickness of the shell), x_2 (the thickness of the head), x_3 (the inner radius), and x_4 (the length of the vessel without the head). The best deterministic result is $f = 5886.4544$, corresponding to $x = [0.7785, 0.3848, 40.3389, 199.7753]$. With the uncertainties in the decision variables taking into account, the pressure vessel optimization problem is formulated as follows:

$$\begin{aligned}
 & \min_{x_{1c}, x_{2c}, x_{3c}, x_{4c}} f(x_{1c}, x_{2c}, x_{3c}, x_{4c}) \\
 & = 0.6224x_{1c}x_{3c}x_{4c} + 1.7781x_{2c}x_3^2 + \\
 & \quad + 3.1661x_{1c}^2x_{4c} + 19.84x_{1c}^2x_3 \\
 \text{s.t.} \quad & g_1 = -x_{1c} + 0.0193x_3 \leq 0 \\
 & g_2 = -x_{2c} + 0.00954x_3 \leq 0 \\
 & g_3 = -\pi x_3^2x_{4c} - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \\
 & g_4 = x_{4c} - 240 \leq 0 \\
 & 0 \leq [x_1] \leq 1.5, 0 \leq x_2 \leq 1.5 \\
 & 30 \leq x_3 \leq 50, 160 \leq [x_4] \leq 200
 \end{aligned}$$

$$\begin{aligned}
 &\text{with } \Delta f_0 = 100 \\
 &[x_1] = [x_{1c} - 0.01, x_{1c} + 0.01] \\
 &[x_4] = [x_{4c} - 0.05, x_{4c} + 0.05].
 \end{aligned}
 \tag{29}$$

where x_{1c} and x_{4c} are the nominal value of x_1 and x_4 , respectively.

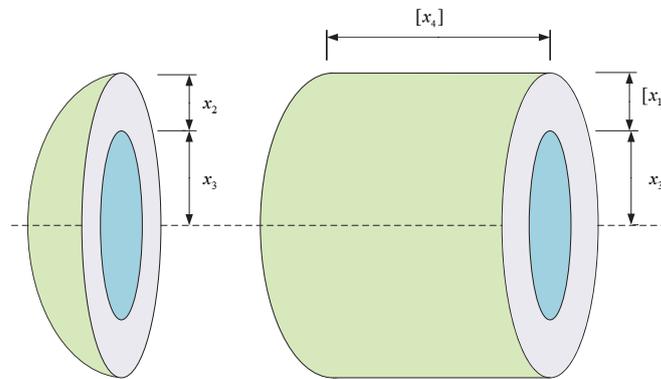


Figure 12. The pressure vessel.

The results obtained by seven interval robust optimization methods are shown in Table 8. It is observed that (1) for feasibility, all the methods can find feasible solutions and satisfy the constraints, (2) for robustness, the methods using IK-GA-RO, STA-RO, I-STA-RO, and H-STA-RO can obtain robust solutions since their robustness violation values (R) equal to 0, and (3) for optimality, although the result of the SQP-RO method has smaller objective function value, it violates the robustness requirements; within the robust solutions, the result of the H-STA-RO method is superior to others because of its lower f value. Moreover, the results of p_r , FE , and T demonstrate the efficiency of the H-STA-RO method. Figure 13 shows the Monte Carlo test results of the deterministic solution and the robust solution of the H-STA-RO method. The deterministic optimum become infeasible in some cases, but the robust optimum of the H-STA-RO method is always feasible even when the decision variables are subject to interval variations.

Table 8. Performance comparison of Example 6.

	IK-GA-RO	GA-RO	SQP-RO	STA-RO	I-STA-RO	H-STA-RO
x_1	0.845	0.7897	0.7804	0.7973	0.7948	0.78831
x_2	0.412	0.3854	0.3847	0.3892	0.3879	0.38472
x_3	43.008	40.3998	40.3268	40.7876	40.6570	40.32681
x_4	165.758	198.9669	199.9500	193.7079	195.5939	199.9500
f	6.09×10^3	5.96×10^3	5.90×10^3	5.97×10^{-3}	5.97×10^3	5.95×10^3
g_1	-0.015	-0.0100	-0.0021	-0.0101	-0.0102	-0.0100
g_2	-0.02	-5.91×10^{-8}	-3.00×10^{-6}	-9.66×10^{-5}	-91.98×10^{-6}	0
g_3	-463.936	-256.4015	-255.7847	-631.8675	-1.23×10^3	-964.9065
g_4	-74.242	-41.0631	-40.00500	-46.2921	-44.4061	-40.0500
R	0	1.61×10^{-5}	0.0079	0	0	0
p_r	-	60%	15%	100%	100%	100%
FE	$14,139 \pm -$	$127,390 \pm 1.16 \times 10^{-6}$	129.7500 ± 58.92	$57,564 \pm 5.85 \times 10^3$	$54,275 \pm 6.57 \times 10^3$	$34,575 \pm 5.88 \times 10^3$
T	-	0.221	0.023	0.156	0.159	0.073

Note: R is the reference value of robustness violation obtained from the Monte Carlo simulation, and - denotes data not available.

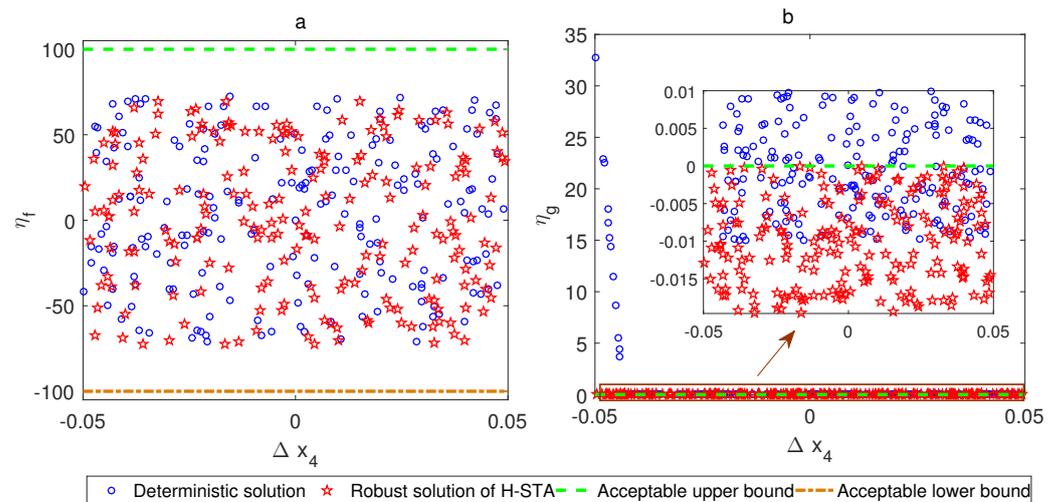


Figure 13. Robustness verification of the deterministic and robust solution for Example 6: (a) objective robustness verification; (b) feasibility robustness verification.

4.7. Example 7: Power Scheduling Design

Power scheduling optimization is an important issue for energy consumption in the process industry. Take the zinc electrowinning process as an example: it accounts for 80% of the total energy consumption of zinc hydrometallurgy. Based on the power time-of-use pricing policy, the aim of power scheduling optimization is to minimize the electricity charge (y) by adjusting the current density (x_1), the concentration of Zn^{2+} (x_2), and H^+ (x_3) in different periods.

A zinc electrowinning process shown in Figure 14 contains seven series potrooms and each potroom has several parallel electrolytic cells. With an appropriate current and zinc acid ratio, zinc ions are deposited on the cathode surface. The electricity use is charged at different prices during three different time periods (peak, shoulder, and off-peak); it is a practice to produce more with lower electricity price. To analyze the power scheduling system, the zinc electrowinning process model is established based on electrochemical reaction mechanism and historical data. With the deterministic parameter, the best result is $y = 1.5922 \times 10^6$, corresponding to $x_1 = [261, 317, 650]$, $x_2 = [60, 45, 60]$, $x_3 = [200, 200, 200]$. With incomplete knowledge of the process model, it is more accurate to estimate some parameters as interval values. The uncertain power scheduling optimization problem is formulated as follows:

$$\begin{aligned} \min_{x_1, x_2, x_3} & f(x_1, x_2, x_3) \\ &= J_0 + \sum_{i=1}^3 \sum_{j=1}^7 P_i T_i V_{ij} L_{ij} \\ \text{s.t.} & g = \sum_{i=1}^3 \sum_{j=1}^7 q E_i T_i L_{ij} = g_{0c} \\ & 100 \leq x_{1,i} \leq 650, 45 \leq x_{2,i} \leq 60 \\ & 160 \leq x_{3,i} \leq 200 \end{aligned}$$

where

$$\begin{aligned} V_{ij} &= V(x_{1,i}, x_{2,i}, x_{3,i}) \\ &= N_j (p_1 - p_2 \ln(p_{3c} x_{3,i}^{-1}) - p_{4c} \ln(p_5 x_{2,i}) \\ &\quad + p_{6c} x_{1,i} (p_{7c} + p_{8c} x_{3,i} - p_{9c} x_{2,i})^{-1} \\ &\quad + p_{10c} \lg x_{1,i} + p_{11c} x_{1,i}) \\ L_{ij} &= L(x_{1,i}) = B_j S x_{1,i} \\ E_i &= E(x_{1,i}, x_{2,i}, x_{3,i}) \end{aligned}$$

$$\begin{aligned}
 &= p_{12c} \exp(p_{13} + p_{14} \lg x_{1,i}) x_{2,i}^{1.6} x_{3,i}^{-0.2} \\
 &+ (p_{15c} \exp(-p_{16c} + p_{17} \lg x_{1,i}) x_{2,i} x_{3,i}^{-0.2} \\
 &+ p_{18c} + p_{19c} x_{2,i}^{0.6})^{-1} x_{1,i}^{-1} \\
 q &= 1.2202 \text{ g/Ah}, S = 1.13 \text{ m}^2, T = [11, 5, 8] \text{ h} \\
 P &= 0.5627 \times [1.6, 1.0, 0.7] \text{ ¥/kWh} \\
 N &= [240, 240, 246, 192, 208, 208, 208] \\
 B &= [34, 46, 54, 56, 56, 57, 57], g_{0c} = 960 \text{ tons} \\
 p &= [1.588, 0.027, 1.1025 \times 10^{-12}, 0.0135, 8.15, \\
 &6.2 \times 10^{-4}, 0.5931, 0.0181, 0.0313, 0.0793, \\
 &5 \times 10^{-4}, 1091.46, -4.06, 2.8, 0.0813, -1.8, \\
 &3.5, 0.35, 2172.45]
 \end{aligned}$$

with $\Delta f_0 = 80000$

$$[p_n] = [0.95 \times p_{nc}, 1.05 \times p_{nc}], n = 3, 4, 6, \dots, 11$$

$$[p_n] = [0.99 \times p_{nc}, 1.01 \times p_{nc}], n = 12, 13, 16, 17, 19$$

$$[g_0] = [g_{0c} - 20, g_{0c} + 20].$$

(30)

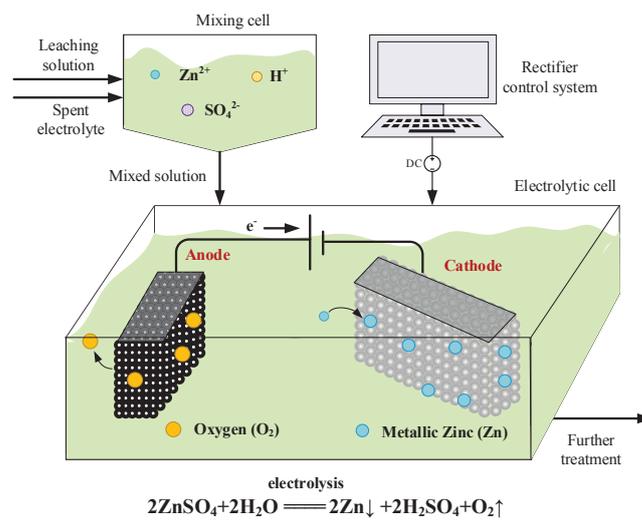


Figure 14. Zinc electrowinning process.

The parameters in the above problem are as follows: J_0 is the capacity electricity charges; P_i is the electricity price in the i th period; T_i is the duration of the i th period; V_{ij} is the cell voltage of the j th plant in the i th period; L_{ij} is the magnitude of the current of the electrolysis process of the j th plant in the i th period; g_0 is the zinc daily output; q is the electrochemical equivalent of zinc; E_i is the current efficiency in the i th period; N_j is the number of cells in the j th plant; B_j is the number of plates in a cell in the j th plant; and S is the area of the cathode plate.

The results obtained by five interval robust optimization methods are shown in Table 9. Compared to GA-RO and SQP-RO, the methods based on STA obtain smaller function values corresponding to a more accurate solution. Meanwhile, methods based on STA could obtain a more robust result. Compared to STA-RO and I-STA-RO, H-STA-RO obtains an accurate solution with less function evaluation and runtime, which denotes the efficiency of the proposed method. Figure 15 shows the Monte Carlo test results of the deterministic solution and the robust solution of the H-STA-RO method. The deterministic optimum becomes infeasible in some cases, but the robust optimum of the H-STA-RO method is always feasible.

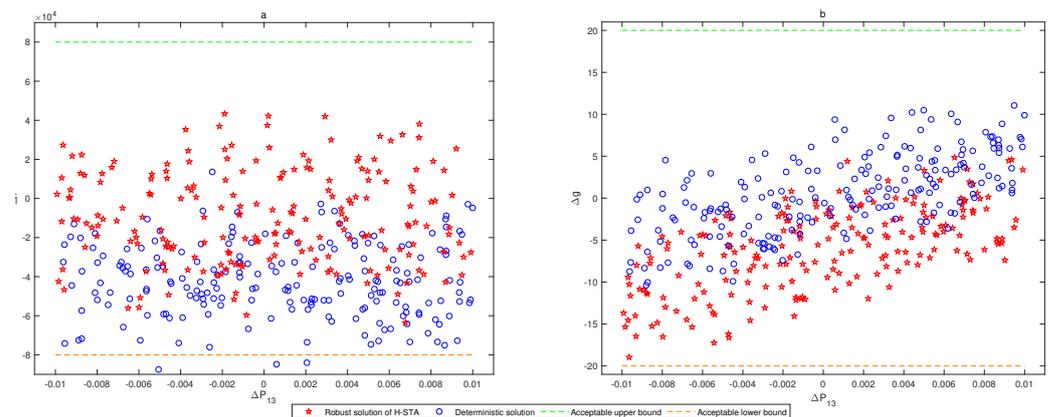


Figure 15. Robustness verification of the deterministic and robust solution for Example 7: (a) objective robustness verification; (b) feasibility robustness verification.

Table 9. Performance comparison of Example 7.

	GA-RO	SQP-RO	STA-RO	I-STA-RO	H-STA-RO
x_1	[255, 324, 600]	[406, 265, 593]	[100, 598, 50]	[189, 404, 650]	[159, 489, 646]
x_2	[60, 60, 60]	[45, 45, 60]	[60, 60, 60]	[60, 59, 60]	[59, 58, 60]
x_3	[184, 184, 200]	[200, 160, 200]	[185, 200, 200]	[160, 200, 179]	[188, 161, 200]
f	1.53×10^6	1.89×10^6	1.48×10^6	1.47×10^6	1.47×10^6
g_1	945.04	951.10	947.64	956.00	954.15
R	0	1.20×10^4	0	0	0
p_r	90%	0%	100%	100%	100%
FE	$110,082 \pm 2.84 \times 10^4$	3392 ± 830.46	$45,529 \pm 5.46 \times 10^3$	$45,334 \pm 4.25 \times 10^3$	$31,470 \pm 5.13 \times 10^3$
T	0.434	0.086	0.487	0.498	0.394

Note: R is the reference value of robustness violation obtained from the Monte Carlo simulation.

5. Conclusions

A hybrid state transition algorithm is proposed to alleviate the problems of robust optimization, including high computational costs and poor convergence. Based on the worst-case analysis, the robust optimization problem can be transformed into a min-max problem. In the outer optimization process (minimization problem), the hybrid state transition algorithm is used to improve the rate of convergence and avoid the local optimum distraction. Meanwhile, the method of sequential quadratic programming is used to strengthen local search ability and reduce computational costs. In the inner optimization process (maximization problem), the second-order Taylor series surrogate model is used to approximate the nonlinear functions and decrease the computational cost. Moreover, to balance the robustness and optimality of candidate solutions, a novel feasibility-checking mechanism is proposed to operate the inner optimization only when a feasible solution is found. Verifying the robustness of the proposed method is conducted using seven examples. The results show that the proposed method offers competitive performance compared with existing robust optimization methods in convergence and efficiency.

In our future work, the robust optimization method for other forms of uncertainties (such as fuzzy uncertainty and interval fuzzy uncertainty) and the applicability enhancement of the surrogate model will be investigated.

Author Contributions: Methodology, J.H.; Validation, H.Z.; Formal analysis, Y.Z.; Investigation, H.Z.; Writing—original draft, H.Z.; Writing—review & editing, J.H.; Supervision, X.Z.; Funding acquisition, J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key Research and Development Program of China (Grant No. 2021YFB1715200), the Hunan Natural Science Foundation (Grant No. 2021JJ40785 and No. 2022JJ40639), and the National Natural Science Foundation of China (Grant No. 62103447 and No. 62203471).

Data Availability Statement: The data presented in this study are available in the verification section of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, J.; Su, C. Robust optimization of microgrid based on renewable distributed power generation and load demand uncertainty. *Energy* **2021**, *223*, 120043. [\[CrossRef\]](#)
2. Witte, D.D.; Qing, J.; Couckuyt, I.; Dhaene, T.; Ginste, D.V.; Spina, D. A robust bayesian optimization framework for microwave circuit design under uncertainty. *Electronics* **2022**, *11*, 2267. [\[CrossRef\]](#)
3. Tirkolaee, E.B.; Mahdavi, I.; Esfahani, M.M.S.; Weber, G.-W. A robust green location-allocation-inventory problem to design an urban waste management system under uncertainty. *Waste Manag.* **2020**, *102*, 340–350. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Ren, Z.; Zhen, X.; Jiang, Z.; Gao, Z.; Li, Y.; Shi, W. Underactuated control and analysis of single blade installation using a jackup installation vessel and active tugger line force control. *Mar. Struct.* **2023**, *88*, 103338. [\[CrossRef\]](#)
5. Shi, G.; Wei, Q.; Liu, D. Optimization of electricity consumption in office buildings based on adaptive dynamic programming. *Soft Comput.* **2017**, *21*, 6369–6379. [\[CrossRef\]](#)
6. Acar, E.; Bayrak, G.; Jung, Y.; Lee, I.; Ramu, P.; Ravichandran, S.S. Modeling, analysis, and optimization under uncertainties: A review. *Struct. Multidiscip. Optim.* **2021**, *64*, 2909–2945. [\[CrossRef\]](#)
7. Sun, X.-K.; Peng, Z.-Y.; Guo, X.-L. Some characterizations of robust optimal solutions for uncertain convex optimization problems. *Optim. Lett.* **2016**, *10*, 1463–1478. [\[CrossRef\]](#)
8. Rahim, S.; Wang, Z.; Ju, P. Overview and applications of robust optimization in the avant-garde energy grid infrastructure: A systematic review. *Appl. Energy* **2022**, *319*, 119140. [\[CrossRef\]](#)
9. Paenke, I.; Branke, J.; Jin, Y. Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation. *IEEE Trans. Evol. Comput.* **2006**, *10*, 405–420. [\[CrossRef\]](#)
10. Shargh, S.; Mohammadi-Ivatloo, B.; Seyedi, H.; Abapour, M. Probabilistic multi-objective optimal power flow considering correlated wind power and load uncertainties. *Renew. Energy* **2016**, *94*, 10–21. [\[CrossRef\]](#)
11. Zhou, X.; Cai, X.; Zhang, H.; Zhang, Z.; Jin, T.; Chen, H.; Deng, W. Multi-strategy competitive-cooperative co-evolutionary algorithm and its application. *Inf. Sci.* **2023**, *635*, 328–344. [\[CrossRef\]](#)
12. Summers, T. Distributionally robust sampling-based motion planning under uncertainty. In Proceedings of the 2018 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 6518–6523.
13. Zhu, X.; Zeng, B.; Dong, H.; Liu, J. An interval-prediction based robust optimization approach for energy-hub operation scheduling considering flexible ramping products. *Energy* **2020**, *194*, 116821. [\[CrossRef\]](#)
14. Ghasemloo, M.; Hejazi, M.A.; Hashemi-Dezaki, H. Flexibility optimization in robust co-optimization of combined power system and gas networks using transmission lines' switching. *Electronics* **2022**, *11*, 2647. [\[CrossRef\]](#)
15. Li, G.; Lu, Z.; Li, L.; Ren, B. Aleatory and epistemic uncertainties analysis based on non-probabilistic reliability and its kriging solution. *Appl. Math. Model.* **2016**, *40*, 5703–5716. [\[CrossRef\]](#)
16. Wang, L.; Liu, D.; Yang, Y.; Wang, X.; Qiu, Z. A novel method of non-probabilistic reliability-based topology optimization corresponding to continuum structures with unknown but bounded uncertainties. *Comput. Methods Appl. Mech. Eng.* **2017**, *326*, 573–595. [\[CrossRef\]](#)
17. Meng, D.; Xie, T.; Wu, P.; He, C.; Hu, Z.; Lv, Z. An uncertainty-based design optimization strategy with random and interval variables for multidisciplinary engineering systems. In *Structures*; Elsevier: Amsterdam, The Netherlands, 2021; Volume 32, pp. 997–1004.
18. Liu, W.; Fu, M.; Yang, M.; Yang, Y.; Wang, L.; Wang, R.; Zhao, a.T. A bi-level interval robust optimization model for service restoration in flexible distribution networks. *IEEE Trans. Power Syst.* **2020**, *36*, 1843–1855. [\[CrossRef\]](#)
19. Kang, Z.; Bai, S. On robust design optimization of truss structures with bounded uncertainties. *Struct. Multidiscip. Optim.* **2013**, *47*, 699–714. [\[CrossRef\]](#)
20. Li, W.; Xiao, M.; Yi, Y.; Gao, L. Maximum variation analysis based analytical target cascading for multidisciplinary robust design optimization under interval uncertainty. *Adv. Eng. Inform.* **2019**, *40*, 81–92. [\[CrossRef\]](#)
21. Rehman, S.U.; Langelaar, M. Efficient global robust optimization of unconstrained problems affected by parametric uncertainties. *Struct. Multidiscip. Optim.* **2015**, *52*, 319–336. [\[CrossRef\]](#)

22. Wu, J.; Luo, Z.; Zhang, N.; Zhang, Y. A new interval uncertain optimization method for structures using Chebyshev surrogate models. *Comput. Struct.* **2015**, *146*, 185–196. [[CrossRef](#)]
23. Xia, B.; Lü, H.; Yu, D.; Jiang, C. Reliability-based design optimization of structural systems under hybrid probabilistic and interval model. *Comput. Struct.* **2015**, *160*, 126–134. [[CrossRef](#)]
24. Hao, P.; Wang, Y.; Liu, C.; Wang, B.; Wu, H. A novel non-probabilistic reliability-based design optimization algorithm using enhanced chaos control method. *Comput. Methods Appl. Mech. Eng.* **2017**, *318*, 572–593. [[CrossRef](#)]
25. Chen, S.H.; Song, M.; Chen, Y.D. Robustness analysis of responses of vibration control structures with uncertain parameters using interval algorithm. *Struct. Saf.* **2017**, *29*, 94–111. [[CrossRef](#)]
26. Zhou, J.; Li, M. Advanced robust optimization with interval uncertainty using a single-looped structure and sequential quadratic programming. *J. Mech. Des.* **2014**, *136*, 021008. [[CrossRef](#)]
27. Siddiqui, S.; Azarm, S.; Gabriel, S. A modified Benders decomposition method for efficient robust optimization under interval uncertainty. *Struct. Multidiscip. Optim.* **2011**, *44*, 259–275. [[CrossRef](#)]
28. Fu, J.; Faust, J.M.M.; Chachuat, B.; Mitsos, A. Local optimization of dynamic programs with guaranteed satisfaction of path constraints. *Automatica* **2015**, *62*, 184–192. [[CrossRef](#)]
29. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*, 28th ed.; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 2006; Volume xiii.
30. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
31. Song, Y.; Liu, Y.; Chen, H.; Deng, W. A multi-strategy adaptive particle swarm optimization algorithm for solving optimization problem. *Electronics* **2023**, *12*, 491. [[CrossRef](#)]
32. Ye, F.; Chen, J.; Tian, Y.; Jiang, T. Cooperative task assignment of a heterogeneous multi-uav system using an adaptive genetic algorithm. *Electronics* **2020**, *9*, 687. [[CrossRef](#)]
33. Dong, Y.; Zhang, H.; Wang, C.; Zhou, X. An adaptive state transition algorithm with local enhancement for global optimization. *Appl. Soft Comput.* **2022**, *121*, 108733. [[CrossRef](#)]
34. Zhou, X.; Yang, C.; Gui, W. A statistical study on parameter selection of operators in continuous state transition algorithm. *IEEE Trans. Cybern.* **2019**, *49*, 3722–3730. [[CrossRef](#)]
35. Han, J.; Yang, C.; Zhou, X.; Gui, W. A new multi-threshold image segmentation approach using state transition algorithm. *Appl. Math. Model.* **2017**, *44*, 588–601. [[CrossRef](#)]
36. Han, J.; Yang, C.; Zhou, X.; Gui, W. A two-stage state transition algorithm for constrained engineering optimization problems. *Int. J. Control Autom. Syst.* **2017**, *16*, 522–534. [[CrossRef](#)]
37. Li, M.; Zhang, W.; Hu, B.; Kang, J.; Wang, Y.; Lu, S. Automatic assessment of depression and anxiety through encoding pupil-wave from hci in vr scenes. *ACM Trans. Multimed. Comput. Commun. Appl.* **2022**. [[CrossRef](#)]
38. Stewart, J. *Calculus: Concepts and Contexts*; Cengage Learning: Boston, MA, USA, 2009.
39. Zhou, X.; Yang, C.; Gui, W. State transition algorithm. *J. Ind. Manag. Optim.* **2012**, *8*, 1039–1056. [[CrossRef](#)]
40. Nocedal, J.; Wright, S.J. *Numerical Optimization*, 2nd ed.; Springer: New York, NY, USA, 2006.
41. Zhou, H.; Zhou, Q.; Liu, C.; Zhou, T. A Kriging metamodel-assisted robust optimization method based on a reverse model. *Eng. Optim.* **2018**, *50*, 253–272. [[CrossRef](#)]
42. Cheng, S.; Li, M. Robust optimization using hybrid differential evolution and sequential quadratic programming. *Eng. Optim.* **2015**, *47*, 87–106. [[CrossRef](#)]
43. Zhou, J.; Cheng, S.; Li, M. Sequential quadratic programming for robust optimization with interval uncertainty. *J. Mech. Des.* **2012**, *134*, 100913. [[CrossRef](#)]
44. Zhou, Q.; Jiang, P.; Shao, X.; Zhou, H.; Hu, J. An on-line Kriging metamodel assisted robust optimization approach under interval uncertainty. *Eng. Comput.* **2017**, *34*, 420–446. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.