

Research Article

An Efficient Hybrid Algorithm for Multiobjective Optimization Problems with Upper and Lower Bounds in Engineering

Guang Yang,¹ Tao Xu,^{1,2} Xiang Li,¹ Haohua Xiu,¹ and Tianshuang Xu¹

¹School of Mechanical Science and Engineering, Jilin University, Changchun 130022, China

²State Key Laboratory of Automotive Simulation and Control, Changchun 130022, China

Correspondence should be addressed to Tianshuang Xu; xts@jlu.edu.cn

Received 12 March 2015; Revised 8 June 2015; Accepted 17 June 2015

Academic Editor: Farhang Daneshmand

Copyright © 2015 Guang Yang 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.

Generally, the inconvenience of establishing the mathematical optimization models directly and the conflicts of preventing simultaneous optimization among several objectives lead to the difficulty of obtaining the optimal solution of a practical engineering problem with several objectives. So in this paper, a generate-first-choose-later method is proposed to solve the multiobjective engineering optimization problems, which can set the number of Pareto solutions and optimize repeatedly until the satisfactory results are obtained. Based on Frisch's method, Newton method, and weighed sum method, an efficient hybrid algorithm for multiobjective optimization models with upper and lower bounds and inequality constraints has been proposed, which is especially suitable for the practical engineering problems based on surrogate models. The generate-first-choose-later method with this hybrid algorithm can calculate the Pareto optimal set, show the Pareto front, and provide multiple designs for multiobjective engineering problems fast and accurately. Numerical examples demonstrate the effectiveness and high efficiency of the hybrid algorithm. In order to prove that the generate-first-choose-later method is rapid and suitable for solving practical engineering problems, an optimization problem for crash box of vehicle has been handled well.

1. Introduction

Most of the practical engineering optimization problems are multiobjective. For example, an airplane design problem might require maximizing fuel efficiency and payload, while minimizing the weight of the structure [1]. Moreover, in automotive industry, people pay more attention to the optimization of occupant restraint systems. In order to meet the design requirements, such as the chest displacement and head injury criterion, the design can be treated as a multiobjective optimization problem [2]. With the increasing demand of multiobjective optimization in engineering problems, researches of multiobjective optimization algorithms are necessary and valuable.

Recently, researchers have proposed various methods with the fast development of multiobjective optimization.

Generally, these methods can be divided into scalar methods and evolution methods by way of solving the optimization problems.

Scalar methods transform the vector optimization problems into adaptive scalar ones. Combining with some gradient methods, a scalar method can easily obtain the optimum by iterations. Typical multiobjective scalar methods include traditional weighted sum method [3, 4], constraint method [5], NBI [6], and multiobjective automatic weighted sum method [7]. With the continuous development, more and more new scalar methods have emerged in this field [8–11].

The idea of evolutionary methods is similar to the biological evolution process, established by Darwin's theory of natural selection. Referring to the biological evolution process, evolutionary methods make use of crossover, mutation, or inheritance operations during iterations to obtain better

results. Evolutionary methods have developed rapidly in the field of multiobjective optimization, such as Multiobjective Genetic Algorithm [12], NSGA-II [13], improving strength Pareto evolutionary algorithm [14], multiobjective particle swarm optimization method [15], and multiobjective artificial immune system optimization algorithm [16]. Recently, some new multiobjective evolutionary methods have been proposed [17, 18].

In recent years, the multiobjective optimization methods are widely used in the engineering problems, such as aerospace [19], automobile [20], information [21], health care [22], and robot and control [23]. The methods can solve specific problems, such as the development of prototype, structural design, and control system design. Most references relative to these fields adopted evolutionary methods. This is because the derivatives of practical engineering optimization problems may not always be obtained, which leads the scalar methods out of work [24]. But the process of converging to the Pareto front by evolutionary methods is slow, random, and difficult to control [25]. According to the analysis of Coello et al., it is hard to determine the stopping criterion definitely. Some researchers often define the iterations as the termination condition of methods [26]. In order to get a satisfactory result, the number of iterations is always defined so large, which means the computation time and efficiency are unacceptable [27].

The purpose of the method proposed in this paper is to provide the design advice to designers quickly. So we hope that, during the process of calculation, the method can rapidly get the Pareto optimal solutions and Pareto front of the multiobjective optimization problems. Meanwhile, if the results are not satisfactory, it is necessary to increase the number of the solutions or reconstruct the models of multiobjective optimization problems. Doubtlessly, the low calculation efficiency of multiobjective evolutionary algorithms dissatisfies this need.

Usually, the engineering optimization problems cannot be solved directly. They can be described approximately by surrogate models and then can be optimized. The surrogate models are always constructed by polynomial response surface method [28], radial basis function method [29], Kriging method [30], and so on, which are generally expressed as secondary or higher order polynomial functions. Obviously, the first-order and second-order derivatives of these functions can be obtained easily, so we take Newton method as basic theory for the proposed method in this paper, which has a fast rate of local convergence.

We should consider that the surrogate models are meaningful only in the constraint interval. So in order to keep the computational accuracy, this paper takes Frisch's method [31] to deal with the constraints, which is one of the interior point methods and can easily obtain the second-order derivatives. Though weighted sums method makes it hard to obtain the uniform Pareto optimal solutions and the Pareto front for nonconvex regions, it is still convenient and effective, as the most commonly used multiobjective scalar method [32]. This paper takes weighted sums method to solve the multiobjective optimization problems, in order to get satisfactory Pareto optimal solutions rapidly.

For obtaining excellent crashworthiness performance of vehicles, some researchers studied about the structural optimization problems. Acara et al. took CFE and SEA as optimization objectives and got a tradeoff solution by sequential quadratic programming [20]. Gu et al. constructed the surrogate models of restraint system by Kriging method, obtained the Pareto front of each objective by NSGA-II, and finally determined the best design points [2]. In this paper, the surrogate models of vehicle's crash box have been constructed and optimized by the proposed method, in order to provide detailed advice for designers rapidly.

The outline of this paper is as follows. Section 2 establishes a generate-first-choose-later method for multiobjective engineering optimization problem, which offers design advice to designers as a reference. Section 3 proposes a hybrid algorithm for multiobjective optimization problems with upper and lower bounds. In Section 4, two numerical examples are calculated by the proposed hybrid algorithm to prove the accuracy and efficiency. In Section 5, a surrogate model of vehicle's crash box is optimized by the proposed method in order to prove the validity for dealing with engineering problems.

2. Strategy for Solving Multiobjective Engineering Optimization Problems

Generally, in engineering optimization problems there are many objectives which are always conflicting with each other. Because of the complex conditions and structural shape, the relationship among objective functions, design variables, and constraint functions is hard to construct directly. Sometimes, in practical engineering problems, numerical simulation and experiments are adopted, but these ways rely on the experience of designers and make it hard to achieve the optimal design globally.

Surrogate models are widely used to improve the efficiency of engineering design and optimization. In this paper, a valid method for solving the engineering optimization problems based on surrogate model is proposed, which is a generate-first-choose-later method. After constructing the surrogate models of multiobjective engineering optimization problems with constraints, the proposed method can calculate the Pareto optimal solutions and Pareto front. The Pareto front is shown intuitively to provide lots of suggestions for designers as a reference. Meanwhile, designers can reset the number of solving Pareto optimal solutions and calculate again, in order to get better results. The flow chart of the proposed method is shown in Figure 1.

In the proposed method, constructing the surrogate models of engineering problems is very important, which will highly influence the accuracy of optimization results. After determining the optimization problem, the surrogate models of objective functions and constraint functions can be constructed by response surface method, radial basis function method, Kriging method, and so on.

It is worth noting that the magnitudes of different objective functions are often different. So we have to unify the expression and normalize the functions before optimization. The theory of the proposed method will be stated in Section 3.

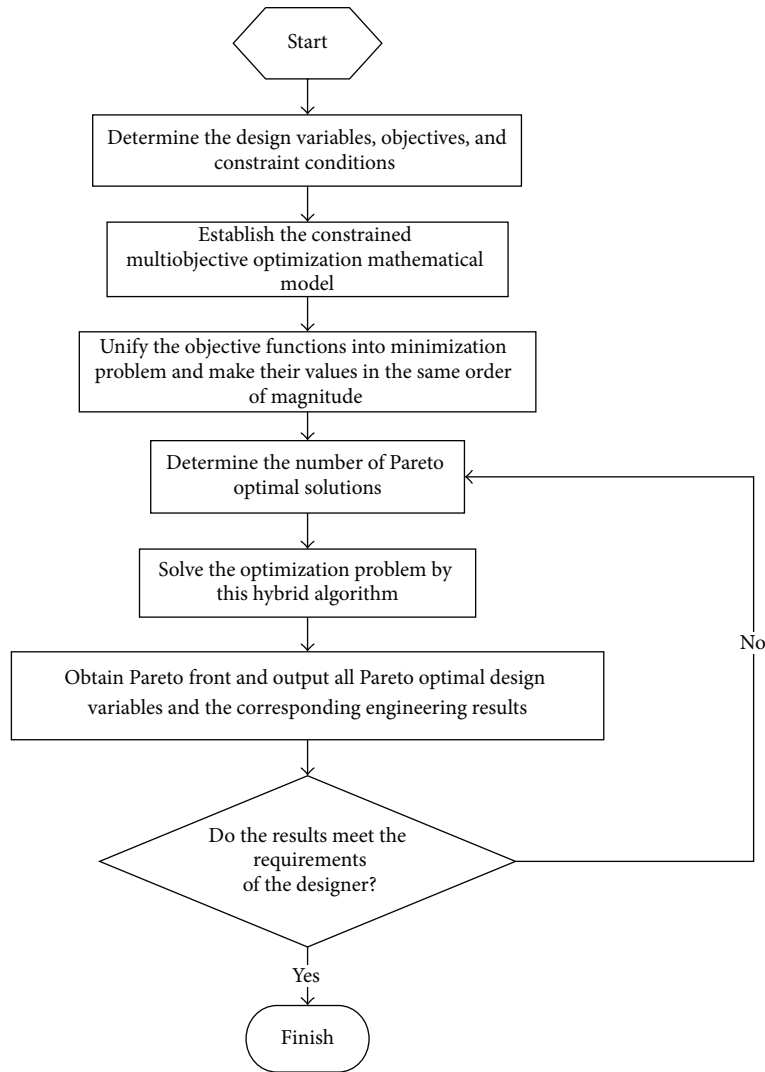


FIGURE 1: The generate-first-choose-later method for multiobjective optimization engineering problems.

3. The Hybrid Algorithm for Multiobjective Optimization Models with Upper and Lower Bounds

3.1. *Disadvantage of Evolutionary Methods.* Evolutionary methods are widely used in engineering multiobjective optimization problems recently. Moreover, the derivative of some engineering problems does not exist or cannot be obtained easily. So the evolutionary methods are more suitable for solving these problems. However, the disadvantages of evolutionary methods should not be neglected [33]. In particular, the slow convergence rate and lack of effective convergence criterion influence the calculation efficiency seriously.

The randomness of searching in the iterative directions leads to the slow convergence rate of evolutionary methods. When the individuals are far from the Pareto optimal solutions, both the field which can be Pareto improved and the opportunity to generate a descent direction randomly are large. With the approach to Pareto optimal solutions, the

conflict among objective functions increases, which leads to the difficulty of finding the descent directions for each objective function. When the individuals are close to the Pareto optimal solutions, both the proportion of the fields which can be Pareto improved and the opportunity to generate a descent direction randomly are small. These are the reasons that convergence rate of evolutionary methods solving the multiobjective optimization problems is fast during the initial stage and slow during the final stage.

Compared with multiobjective evolutionary methods, the advantage of local search methods in efficiency is remarkable, such as Newton method.

3.2. *Newton Method.* The models of multiobjective engineering optimization problems in this paper are established by response surface methods, and also the constraints are handled by log functions. So the optimization models are derivable and the second derivative can be obtained by gradient methods. Meanwhile, the method proposed in this

paper may calculate Pareto optimal solutions more than once in order to provide satisfying advice for designers. Hence, the computational efficiency of the method is very important.

Although evolutionary methods are widely used in multiobjective engineering optimization problems, the computational efficiency is not satisfactory. In this paper, a multiobjective scalar method is researched, which has the advantage of fast convergence. Newton method has been chosen to calculate Pareto optimal solutions in this paper, for its high computational efficiency.

The iteration direction of Newton method includes the gradient and Hessian matrix information of objective functions, so the iteration point can be definitely close to the optimal point. When the iteration point is near the optimal point, the rate of convergence is rapid [34]. If the objective functions satisfy some conditions, it can achieve superlinear convergence or quadratic convergence [10]. In rare cases, the obtained Newton directions are not descent reducing the computational efficiency. This paper adopted a decision mechanism to solve this problem and improve calculation efficiency, which chooses the negative gradient direction as descent direction to replace the nondescent Newton directions.

3.3. Establish the Hybrid Algorithm. Newton method is chosen as the main calculation algorithm for searching the solutions. The process of solving multiobjective engineering optimization problems with upper and lower bounds is described in detail. And the overall construction course of the hybrid algorithm is in the following.

3.3.1. Mathematical Model and Pareto Optimal Solution. In practical engineering optimization problem, there are always several objects which are conflicting to prevent simultaneous optimization of each other. There is no one optimal solution satisfying all the minima of objects. So searching for the Pareto optimal set of these objects is one of the most effective ways. Meanwhile, there are some constraints in the engineering optimization problems, generally about the upper and lower bounds of design variables. In this section, a valid method will be proposed, in order to fast calculate the multiobjective optimization problem with upper and lower bounds.

The m objects of engineering optimization problem can be denoted as

$$\mathbf{F}(\mathbf{x}) = (F_1(\mathbf{x}), F_2(\mathbf{x}), \dots, F_m(\mathbf{x}))^T, \quad (1)$$

where the n -dimensional vector is $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$. The upper and lower bounds of the design variables are always constrained as

$$l \leq x_i \leq u, \quad i = 1, \dots, n. \quad (2)$$

For convenient calculation, the upper and lower bounds can be transformed into upper and lower bounds; that is,

$$-x_i^2 + lx_i + ux_i - lu \geq 0. \quad (3)$$

So the multiobjective optimization engineering problem with n constraints of upper and lower bounds can be uniformly written as

$$\begin{aligned} \min \quad & \mathbf{F}(\mathbf{x}) \\ \text{st.} \quad & -x_i^2 + lx_i + ux_i - lu \geq 0, \end{aligned} \quad (4)$$

where $\mathbf{F}(\mathbf{x})$ is vector function with m objects and \mathbf{x} is a vector with n variables. All the design variables satisfying the constraints construct the feasible region of the optimization problem, denoted as \mathbf{X} .

The aim of solving multiobjective optimization problems is to obtain the Pareto optimal set. For two design variables \mathbf{x}^{\wedge} and \mathbf{x}^* , it is said that \mathbf{x}^* is Pareto dominant, if and only if

$$\begin{aligned} & (\forall j \in \{1, 2, \dots, m\} : F_j(\mathbf{x}^*) \leq F_j(\mathbf{x})) \\ & \wedge (\exists k \in \{1, 2, \dots, m\} : F_k(\mathbf{x}^*) < F_k(\mathbf{x})), \end{aligned} \quad (5)$$

denoted as $\mathbf{x}^* > \mathbf{x}$. The vector $\mathbf{x}^* \in \mathbf{X}$ is a Pareto optimal solution only under the condition that there does not exist $\mathbf{x} \in \mathbf{X}$ for $\mathbf{x} > \mathbf{x}^*$. So a Pareto optimal solution means the reasonable solution, which satisfies the objectives at an acceptable level without being dominated by any other solution. The method proposed in this paper can obtain both the Pareto optimal solutions and the Pareto fronts of the multiobjective optimization problems with constraints.

3.3.2. Handling of the Constraints. The process of solving multiobjective optimization problems with constraints is to find the Pareto optimal set of all the objects in the feasible region under the constraint conditions. The problem should be transformed into an unconstrained one first and then be solved. Hence, a penalty term can be added to the object functions. When the penalty term is closer to zero, it means the design variables satisfy the constraints. During the solving process, the penalty term should be scaled down until it is small enough and can be neglected relative to the object values for meeting the stopping criterion. At the moment, the obtained solution can not only be equivalent to the optimal solution of the original problem but also satisfy the constraints.

The optimization problems discussed in this paper are based on surrogate models. So the interior point method is chosen to deal with the constraints. For solving the gradient and Hessian matrix conveniently, the penalty term is constructed by Frisch's method, which is expressed as

$$q(\mathbf{x}) = -\sigma^{-1} \sum_{i=1}^n \log(lx_i + ux_i - x_i^2 - lu). \quad (6)$$

3.3.3. Iteration Direction for Multiobjective Optimization. Despite having deficiencies in depicting the Pareto optimal set, the weighted sum method for multiobjective optimization continues to be used extensively not only to provide multiple solution points by varying the weights consistently but also to provide the solutions that reflect the preference of each object. In the proposed method of this paper, weighted

sum method is chosen. After the designers set the number of Pareto optimal solutions, the corresponding group of weighting factors can be provided uniformly.

Because the gradient and Hessian matrices of object and constraint functions can be obtained, Newton method is selected for its rapid convergence. In order to improve computational efficiency, the negative gradient direction is considered when the Newton directions are not descent. In this paper, the Pareto optimal solutions will be solved by iteration. The process of deducing iteration direction is as follows.

First, the penalty functions should be constructed. The logarithmic penalty function of the j th ($j = 1, 2, \dots, m$) object function can be denoted as

$$p_j(\mathbf{x}) = F_j(\mathbf{x}) + q(\mathbf{x}). \quad (7)$$

Express $p_j(\mathbf{x} + \mathbf{d})$ by Taylor expansion; that is,

$$p_j(\mathbf{x} + \mathbf{d}) = p_j(\mathbf{x}) + \nabla p_j(\mathbf{x})^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \nabla^2 p_j(\mathbf{x}) \mathbf{d}. \quad (8)$$

For \mathbf{x}_k , the iteration direction \mathbf{d} satisfies $\mathbf{d} = \mathbf{x} - \mathbf{x}_k$, and the penalty function of the j th object function is

$$\begin{aligned} p_j(\mathbf{x}) &= p_j(\mathbf{x}_k + \mathbf{d}) \\ &= p_j(\mathbf{x}_k) + \nabla p_j(\mathbf{x}_k)^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \nabla^2 p_j(\mathbf{x}_k) \mathbf{d}. \end{aligned} \quad (9)$$

Then, adding the penalty function of every object function together, a sum function can be expressed as

$$\begin{aligned} P(\mathbf{x}) &= \sum_{j=1}^m \lambda_j p_j(\mathbf{x}) \\ &= \sum_{j=1}^m \lambda_j \left[p_j(\mathbf{x}_k) + \nabla p_j(\mathbf{x}_k)^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \nabla^2 p_j(\mathbf{x}_k) \mathbf{d} \right], \end{aligned} \quad (10)$$

where $\sum_{j=1}^m \lambda_j = 1$ ($\lambda_j > 0$).

By calculating the derivative of P with respect to \mathbf{d} , one can get the iteration direction at \mathbf{x}_k ; that is,

$$\mathbf{d}_k = - \left[\sum_{j=1}^m \lambda_j \nabla^2 p_j(\mathbf{x}_k) \right]^{-1} \sum_{j=1}^m \lambda_j \nabla p_j(\mathbf{x}_k). \quad (11)$$

This method combines Newton method and linear weighted sum method, and the iteration direction \mathbf{d} is equal to Newton direction. The iteration direction of sum function can be obtained only based on the gradient and Hessian matrices of each penalty function. But Newton method is locally convergent. When the sum function is not continuous twice differentiable, the improper selection of the initial point cannot ensure the iteration direction is descent, which affects the computational efficiency of optimization greatly.

In order to ensure all the iteration directions are descending during the optimization, an identification process is introduced. If Newton direction at some point is ascending, take the negative gradient direction of sum function at this point as the iteration direction. The criterion is the product

of Newton direction and negative gradient direction, denoted as

$$a_k = - \left(\sum_{j=1}^m \lambda_j \nabla p_j(\mathbf{x}_k) \right)^T \left(\sum_{j=1}^m \lambda_j \nabla^2 p_j(\mathbf{x}_k) \right)^{-1} \sum_{j=1}^m \lambda_j \nabla p_j(\mathbf{x}_k). \quad (12)$$

So the iteration direction at \mathbf{x}_k is

$$\mathbf{d}_k = \begin{cases} - \left(\sum_{j=1}^m \lambda_j \nabla^2 p_j(\mathbf{x}_k) \right)^{-1} \sum_{j=1}^m \lambda_j \nabla p_j(\mathbf{x}_k), & \text{if } a_k < 0 \\ - \sum_{j=1}^m \lambda_j \nabla p_j(\mathbf{x}_k), & \text{if } a_k \geq 0. \end{cases} \quad (13)$$

During the process of calculation, the selection of iteration step length is necessary. For the object functions are based on surrogate models, the accuracy can be ensured only if the design variables satisfy the constraint conditions. Hence, a criterion is set to prevent that the design variables dissatisfy the constraint conditions. When the iteration point is beyond the range of constraints, the step length will be scaled down. Until the new iteration point satisfies the constraints, output the current step length.

3.3.4. The Proposed Hybrid Method: Algorithm 1. In this section, details of calculating Pareto optimal solutions then forming the Pareto optimal set and Pareto front will be described.

As in the introduction above, proposing a method to obtain Pareto optimal solutions rapidly is the key of this paper. In this paper, based on Frisch's method, Newton method, and weighted sum method, an effective algorithm is put forward, named Algorithm 1. Not only the reason for choosing these theories, but also the derivation process of penalty term, iteration direction, and step length have been given in the previous sections. Now the iteration steps of Algorithm 1 are stated as follows.

Algorithm 1. The whole process of Algorithm 1 for calculating a Pareto optimal solution is as follows.

Step 1. Establish the logarithmic penalty functions of each objective $P_j(\mathbf{x})$, $j = 1, 2, \dots, m$, and calculate $P(\mathbf{x})$.

Step 2. Choose an initial point $\mathbf{x}_0 \in \mathbf{X}$ and give stopping criteria ε and η and coefficients $\alpha = 0.9$, $\rho = 0.5$, and $\sigma = 2$.

Step 3. Calculate the gradients of each logarithmic penalty function $\nabla p_j(\mathbf{x}_k)$.

If $\| \sum_{j=1}^m \lambda_j \nabla p_j(\mathbf{x}_k) \| \leq \varepsilon$, stop the algorithm; go to Step 8.

Else, go to Step 4.

Step 4. Calculate the iterative direction \mathbf{d}_k .

If $-(\sum_{j=1}^m \lambda_j \nabla p_j(\mathbf{x}_k))' (\sum_{j=1}^m \lambda_j \nabla^2 p_j(\mathbf{x}_k))^{-1} \sum_{j=1}^m \lambda_j \nabla p_j(\mathbf{x}_k) < 0$, $\mathbf{d}_k = -(\sum_{j=1}^m \lambda_j \nabla^2 p_j(\mathbf{x}_k))^{-1} \sum_{j=1}^m \lambda_j \nabla p_j(\mathbf{x}_k)$.

$$\text{Else, } \mathbf{d}_k = -\sum_{j=1}^m \lambda_j \nabla P_j(\mathbf{x}_k).$$

Step 5. Calculate iteration step size α .

If for all of the constraint functions $g_i(\mathbf{x}_k) > 0$, $i = 1, 2, \dots, m$, $\alpha = \alpha$, then go to Step 7.

Else, go to Step 6.

Step 6. Consider $\alpha = \rho\alpha$; go to Step 5.

Step 7. Iteratively calculate $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha\mathbf{d}_k$, define $k = k + 1$, and go to Step 3.

Step 8. Calculate penalty term $q(\mathbf{x}_k)$; if $|q(\mathbf{x}_k)| \leq \eta$, stop and output \mathbf{x}_k .

Else, $\sigma = 10\sigma$; go to Step 1.

In the process, the small positive constants ε and η are stopping criteria and α is step length. ρ is the reduction scale of step length and σ is the coefficient of penalty term.

Algorithm 1 shows the process of solving one Pareto optimal solution, which is the core of the proposed method in this paper. However, in order to provide comprehensive references for designers, one Pareto optimal solution is not enough. So a method for obtaining a Pareto optimal set is proposed based on Algorithm 1. The designers should determine L firstly, which is the number of Pareto optimal solutions. Then, by giving L group of weighting factors uniformly, the method can calculate the Pareto optimal solutions of different weight factors in turn and form the Pareto optimal solutions set. If the number of object functions is not more than 3, the Pareto front can be expressed as a coordinate graph. The whole process is shown in Figure 2.

According to this method, designers should only set the number of solutions and initial point. By calculating automatically, a Pareto optimal set will be output. In addition, the Pareto front will be shown as a coordinate graph for designers.

3.3.5. The Benefits and Shortcomings of the Present Method. The solution of multiobjective function optimization problems obtained by the present method is always a local solution, which converges to the real Pareto front when the stopping criterion is close to zero. If the objective functions are convex then the local solution is global one at the same time. By the popular evolutionary algorithms, global solution can be got, which turns out to be not close to the real Pareto front in a short time. When the constrained and objective functions are continuously differentiable and nonlinear, the solution close to the real Pareto front can be got rapidly by the proposed method. So, a more accurate solution can be obtained by the proposed algorithm in a short time. However, a good solution is hard to obtain by the algorithm when the constrained and objective functions are not continuously differentiable and nonlinear.

Another advantage by the present method is high efficiency in converging to the Pareto front, and the shortcoming is that the objective functions and inequality constraints must be continuously differentiable and nonlinear [10]. All the

benefits and shortcomings are because the Newton method is used to calculate the iteration direction.

On one side, many multiobjective engineering optimization problems can be established as the mathematical models which are nonlinear and continuously differentiable, such as the engineering problems described by surrogate models. On the other side, the hybrid method provides references to designers more rapidly than popular evolutionary algorithms, which will improve working efficiency apparently. So, the method proposed in this paper presents an applicable value for actual engineering optimization problems.

The solutions by different algorithms are compared in detail next.

4. Numerical Examples

Two benchmark numerical examples are chosen to check the algorithm. One of them is an example in the user's guide of MATLAB [35] and the other one is from a published paper [36]. The prototype implementations of them are executed by MATLAB V7.12.0. Test 1 we deal with here is to minimize $F_1 = x_1^4 - 10x_1^2 + x_1x_2 + x_2^4 - x_1^2x_2^2$ and $F_2 = x_2^4 - x_1^2x_2^2 + x_1^4 + x_1x_2$, with $-5 \leq x_1 \leq 5$ and $-5 \leq x_2 \leq 5$. Test 2 is from the paper of Schaffer, which is to minimize the following two objective functions as $F_1 = x^2$ and $F_2 = (x - 2)^2$, with $-1000 \leq x \leq 1000$.

For further evaluation, the tests are also executed by Multiobjective Genetic Algorithm. Then, the results obtained by the proposed method are compared with the one by Multiobjective Genetic Algorithm. The methods' performances are evaluated from three aspects, which include the diversity of solutions in Pareto front, the accuracy of the Pareto solutions, and the computational efficiency. To keep things simple, the Multiobjective Genetic Algorithm is written as MOGA for short and the proposed algorithm is named as NSWFA. In all tests, one hundred initial points are iterated for a Pareto optimal set. The Pareto optimal fronts of the two tests are shown in Figures 3 and 4.

Sometimes, the solutions of multiobjective optimization problems, which are obtained by the algorithms based on weighted sum method, are not well distributed in the Pareto optimal front. Studies can be found in this field [4, 32]. However, the algorithm in this paper is for multiobjective engineering optimization problems, which do not require even-distributed Pareto optimal solutions in the Pareto optimal front. With the purpose of offering reference to engineers, some better designs can usually be obtained by this algorithm, which is enough. In Figures 3 and 4, the difference between the Pareto fronts obtained by the MOGA and NSWFA is clear. Although the Pareto optimal points acquired by NSWFA are not distributed evenly, the spread is much better than the solutions by MOGA. For studying the accuracy of the results, the Pareto optimal fronts obtained by the two algorithms are shown in the same figures. In Figures 5 and 6, the Pareto optimal fronts obtained by NSWFA are closer to the real Pareto optimal fronts than MOGA. So the results by the proposed algorithm have good accuracy.

Another important performance is the computational efficiency, which is studied from the iterative number and

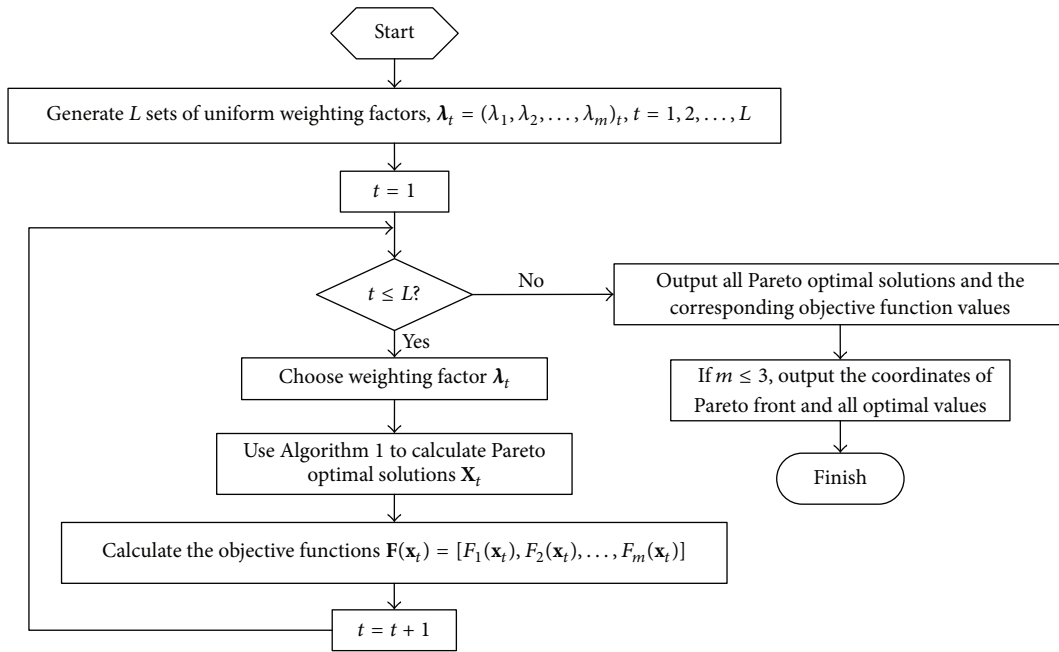


FIGURE 2: The flow chart of the hybrid algorithm for constrained multiobjective optimization problems.

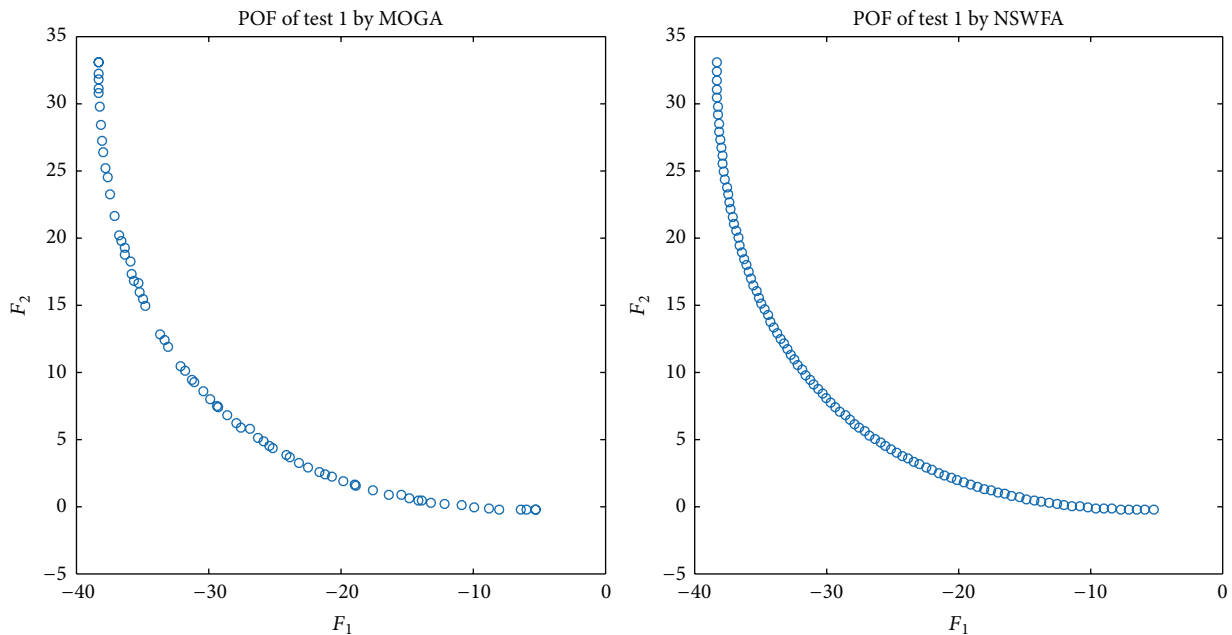


FIGURE 3: The Pareto optimal front of test 1 obtained by the two algorithms.

the consumed CPU time. Facts have proved that the iterative points cannot be convergent to the Pareto optimal front in a short time by evolutionary algorithms.

In this paper, the maximum number of iterations is 2000, and the results by MOGA at the 2000th iteration are recorded. More detailed information is listed in Table 1. In all tests, the calculations are executed ten times by each algorithm and the data in Table 1 are the real results on average.

The stopping criterion is 10^{-5} and the results by MOGA are all not convergent. But the results by NSWFA are

convergent to the Pareto optimal front with less iterations and CPU time. At each iteration, the whole population is iterated at the same time by MOGA, but only one point is iterated by NSWFA. The average iterations of one Pareto optimal solution are only 7.3 and 6.15 times by NSWFA. That is why the CPU time by NSWFA is far less than MOGA. In addition, more Pareto optimal solutions are got by NSWFA than MOGA with the same initial points.

Generally, the Pareto optimal solutions obtained by NSWFA have better accuracy and spread than MOGA. Also,

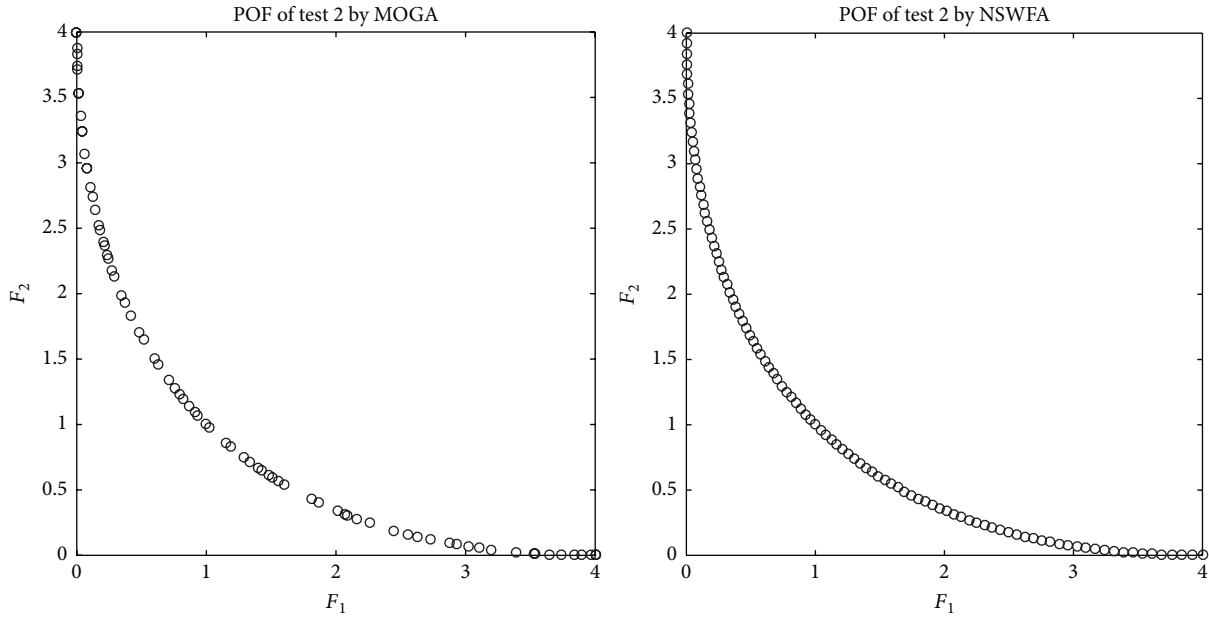


FIGURE 4: The Pareto optimal front of test 2 obtained by the two algorithms.

TABLE 1: The detailed comparison of the two algorithms in computation efficiency.

	Algorithm	Stopping criteria	Initial points	Obtained points	CPU time	Iterations	Convergence
Test 1	MOGA	10^{-5}	100	70	329.72 s	2000	No
	NSWFA	10^{-5}	100	100	13.23 s	730	Yes
Test 2	MOGA	10^{-5}	100	77	361.88 s	2000	No
	NSWFA	10^{-5}	100	100	5.25 s	615	Yes

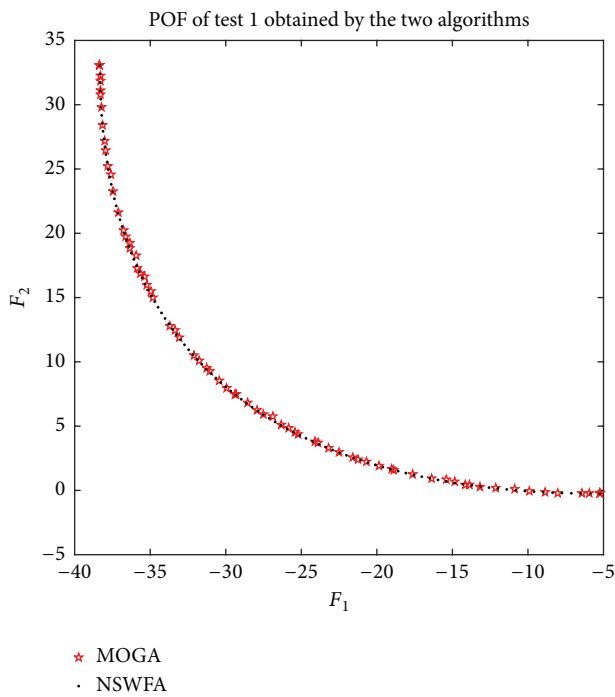


FIGURE 5: The solutions obtained by NSWFA are closer to real Pareto optimal front than MOGA.

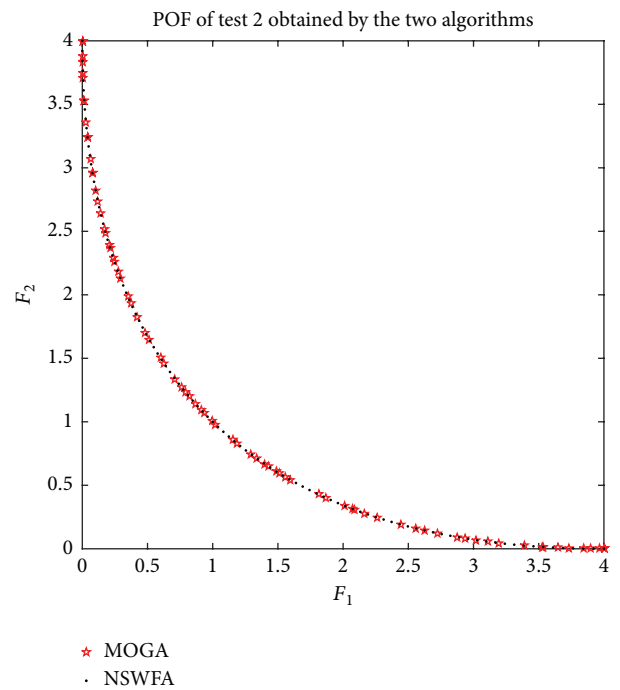


FIGURE 6: The solutions obtained by NSWFA are a little closer to real Pareto optimal front than MOGA.

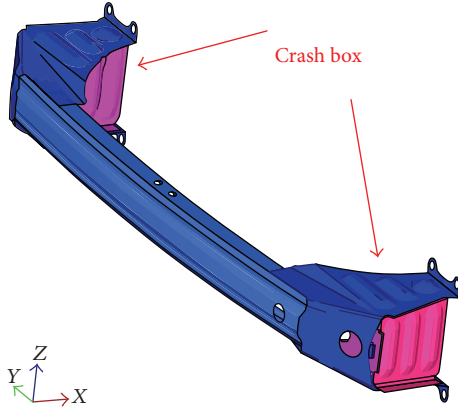


FIGURE 7: The location of crash box.

far less CPU time is consumed by NSWFA than MOGA. All the evidence suggests that NSWFA is perfect in the multi-objective optimization problems whose mathematical models and inequality constraints are nonlinear and continuous.

5. The Design of Crash Box

The crash box is an important part of car collision system, which plays an important role in occupant protection during the collision of vehicles at low speed. In the section, the generate-first-choose-later method with the hybrid algorithm proposed for multiobjective engineering optimization design problems in this paper is adopted to finish the design of the crash box.

5.1. The Multiobjective Engineering Optimization Problem. The properties of energy absorption and maximum crushing force must be considered simultaneously in designing the crash box. The structure design becomes a problem of complicated multiobjective optimization design. The crash box is made of four cutting boards whose thickness can be chosen as design variables to optimize. In the collision of vehicles at low speed, the crash box should absorb the collision energy as much as possible, but the peak force should be small as soon as possible. So in this problem, energy absorbing and the biggest impact are selected to be objectives, and the four wall thickness values of the crash box are selected as design variables. A car collision system with two crash boxes is shown in Figure 7, and the wall thickness as the design variables is shown in Figure 8.

5.2. Design of Experiment, Construct Surrogate Model, and Variance Analysis. In order to effectively simulate the energy absorption characteristic of crash box under axial load in the vehicle frontal impact, choose the bumper and crash box as a whole for research according to real crash process.

The rear part of crash box connecting vehicle body is constrained, while the front part is struck by a rigid wall weighing one ton with speed of 4 m/s. In the simulation of low-speed crash, the model could use elastoplastic material without regard to strain-rate effect. The response surface models are established referring to the paper of Li et al. [37].

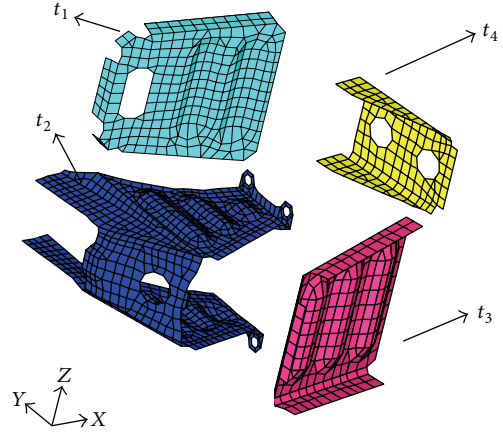


FIGURE 8: The 4 design variables of crash box.

Adopting quadratic regression orthogonal combination design of experiment and distributing experiment point reasonably, 25 simulation experiments are completed.

According to the result, the response surface models of energy absorption E and maximum crushing force F with respect to wall thickness $\mathbf{t} = [t_1, t_2, t_3, t_4]^T$ are obtained by polynomial response surface method:

$$\begin{aligned}
 E &= -2127.7516 + 309.9367t_1 + 2072.4733t_2 \\
 &+ 1018.7752t_3 + 1434.2749t_4 - 25.6145t_1^2 \\
 &- 71.5937t_2^2 - 13.4367t_3^2 + 19.0837t_4^2 \\
 &- 18.6774t_1t_2 + 62.1497t_1t_3 - 16.4747t_1t_4 \\
 &- 39.9887t_2t_3 - 168.8719t_2t_4 - 204.2693t_3t_4, \\
 F &= 40.5927 + 5.9920t_1 - 8.9279t_2 - 6.8074t_3 \\
 &- 9.8654t_4 - 1.8214t_1^2 + 4.3488t_2^2 \\
 &+ 4.2041t_3^2 + 5.2171t_4^2 + 0.4926t_1t_2 \\
 &+ 1.3087t_1t_3 - 1.7178t_1t_4 + 2.4475t_2t_3 \\
 &+ 6.1385t_2t_4 + 0.9046t_3t_4.
 \end{aligned} \tag{14}$$

After getting the surface functions, the variance analysis is used to verify the fitting degree. In the process, the determination coefficient R^2 and the adjusting determination coefficient R_{adj}^2 are calculated to check the fitting precision, which are defined as follows:

$$\begin{aligned}
 R^2 &= \frac{\sum_{i=1}^M (\hat{y}_i - \bar{y}_i)^2}{\sum_{i=1}^M (y_i - \bar{y}_i)^2}, \\
 R_{adj}^2 &= \frac{1 - \sum_{i=1}^M (y_i - \hat{y}_i)^2 (M-1)}{\sum_{i=1}^M (y_i - \bar{y}_i)^2 (M-k-1)}.
 \end{aligned} \tag{15}$$

In the formula, M is the number of samples, k is the number of design variables, and y_i , \hat{y}_i , and \bar{y}_i are the

measured value, predicted value, and the average of measured value.

In general, if the determination coefficient and adjusting determination coefficient are more close to 1, the response surface function with respect to response variables is more precise. The determination coefficients R^2 of E and F are 99.72% and 99.11%, respectively. The adjusting determination coefficients R^2_{adj} are 99.37% and 98.12%, respectively. Thus, the response surface functions in the paper can simulate response variables accurately.

5.3. *The Constrained Multiobjective Optimization Model of Crash Box (The Method Is Proposed for Reference).* In order to reduce passenger injury, the crash box is desired to absorb more energy and generate less maximum crushing force, that is, maximize the objective function E and minimize the objective function F .

The standard multiobjective optimization model should be established firstly. When converting the objective function to minimization problem, the magnitude of objective functions should be comparable since large magnitude results in large deviation. Based on simulation result, the minimum energy absorption (3365.7 J) and maximum crushing force (144.129 kN) are chosen to build corresponding objective function. The constraint of crash box design is upper and lower limitation of wall thickness. The blanking plate should be between 1 mm and 3 mm due to processing factor, that is, wall thickness $t_i \in [t_i^L, t_i^U]$, $t_i^L = 1$, $t_i^U = 3$, $i = 1, \dots, 4$. The standard multiobjective optimization model is

$$\begin{aligned} & \min_{\mathbf{t} \in \mathbb{R}^4} \mathbf{F}(\mathbf{t}) \\ & \text{s.t.} \quad -t_i^2 + 4t_i - 3 \geq 0, \\ & \mathbf{F}(\mathbf{t}) = \begin{cases} F_1 = \frac{3365.7}{E} \\ F_2 = \frac{F}{144.129}, \end{cases} \quad \mathbf{t} = [t_1, t_2, t_3, t_4]^T. \end{aligned} \tag{16}$$

To solve constrained multiobjective optimization problem by the proposed method, set 100 as the number of Pareto optimal solutions and pick a random initial design variable in accordance with constraint. The Pareto optimal front of standard model by rapid calculation is shown in Figure 9.

According to Pareto optimal solution set, the corresponding energy absorption and maximum crushing force are obtained to form Pareto front of real crash problem, which helps engineers choose design proposal more conveniently. The Pareto front end of crash problem is shown in Figure 10.

The design of crash box mainly aims to improve performance. However, the mass of structure is also a very important factor. The mass of crash box varies according to the design variables. Their relation is geometrical. It is easy to get mass based on design variables in the simulation software. The coordinates graph about the energy absorption, maximum crushing force, and mass is shown in Figure 11.

Figure 11 provides designers with a more comprehensive and intuitive reference. They can check the performance and mass of crash box with different wall thickness. Designers

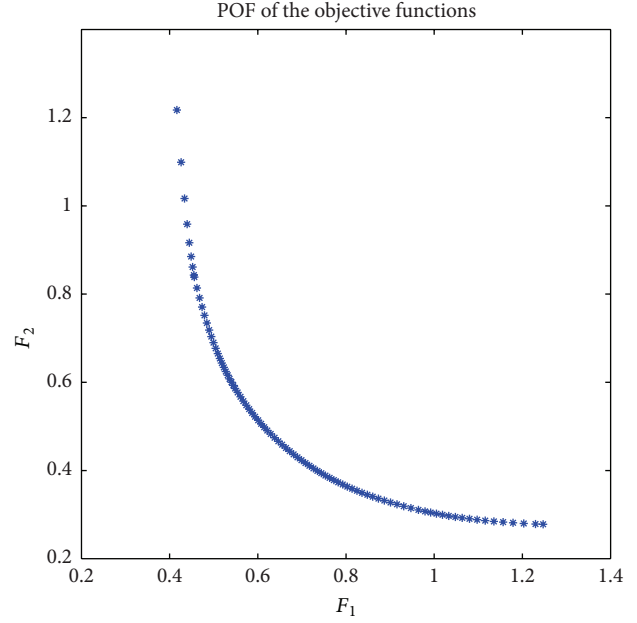


FIGURE 9: The Pareto optimal front of the standard multiobjective optimization model.

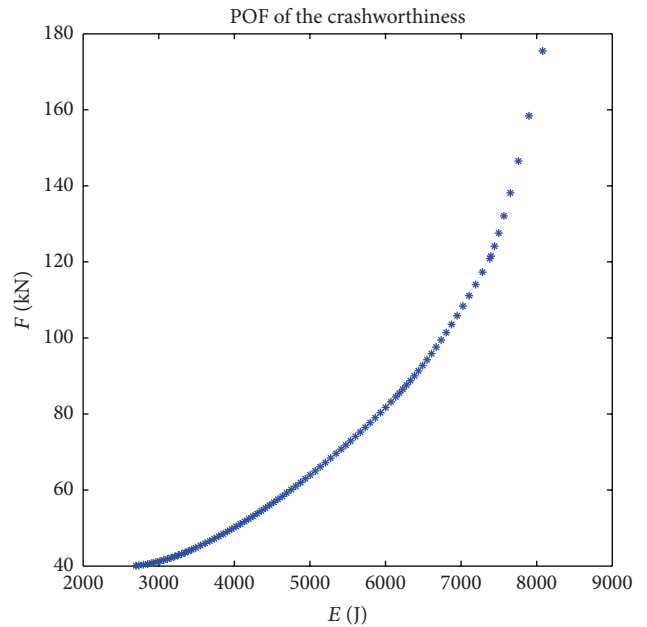


FIGURE 10: The corresponding properties of the Pareto optimal solutions.

could choose reasonable point based on actual needs and find corresponding design variables as initial reference. The Pareto optimal solutions set and their performance calculated in this paper are shown in Table 2.

5.4. *Choose the Design and Check the Properties by Simulation.* According to the results in Table 2, the maximum crushing force increases with the increase of the energy absorption

TABLE 2: The results of 100 pieces of design advice for reference.

序号	λ		t/mm				E/J	F/kN	M/kg
	λ_1	λ_2	t_1	t_2	t_3	t_4			
1	0.01	0.99	3.0000	1.0000	1.0000	1.0000	2699.0	40.0865	2.4183
2	0.02	0.98	3.0000	1.0000	1.0000	1.0000	2699.0	40.0865	2.4183
3	0.03	0.97	3.0000	1.0000	1.0000	1.0000	2699.0	40.0865	2.4183
4	0.04	0.96	3.0000	1.0000	1.0000	1.0362	2737.0	40.1824	2.4250
5	0.05	0.95	3.0000	1.0000	1.0000	1.0937	2797.5	40.3628	2.4357
6	0.06	0.94	3.0000	1.0000	1.0000	1.1486	2855.4	40.5669	2.4458
7	0.07	0.93	3.0000	1.0000	1.0000	1.2011	2910.9	40.7920	2.4556
8	0.08	0.92	3.0000	1.0000	1.0000	1.2518	2964.5	41.0361	2.4650
9	0.09	0.91	3.0000	1.0000	1.0000	1.3007	3016.4	41.2977	2.4741
10	0.10	0.90	3.0000	1.0000	1.0000	1.3483	3066.9	41.5755	2.4829
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
40	0.40	0.60	3.0000	2.0071	1.0123	1.4285	4692.9	59.2051	3.0899
41	0.41	0.59	3.0000	2.0575	1.0298	1.3985	4753.4	60.1006	3.1171
42	0.42	0.58	3.0000	2.1092	1.0480	1.3664	4814.7	61.0222	3.1447
43	0.43	0.57	3.0000	2.1621	1.0668	1.3323	4876.9	61.9718	3.1728
44	0.44	0.56	3.0000	2.2164	1.0862	1.2958	4940.1	62.9514	3.2014
45	0.45	0.55	3.0000	2.2720	1.1064	1.2570	5004.4	63.9635	3.2305
46	0.46	0.54	3.0000	2.3291	1.1274	1.2158	5070.0	65.0103	3.2601
47	0.47	0.53	3.0000	2.3877	1.1492	1.1719	5137.0	66.0947	3.2903
48	0.48	0.52	3.0000	2.4478	1.1718	1.1252	5205.5	67.2194	3.3210
49	0.49	0.51	3.0000	2.5095	1.1952	1.0758	5275.7	68.3873	3.3522
50	0.50	0.50	3.0000	2.5729	1.2195	1.0235	5347.7	69.6017	3.3841
51	0.51	0.49	3.0000	2.6194	1.2400	1.0000	5414.4	70.7405	3.4107
52	0.52	0.48	3.0000	2.6526	1.2574	1.0000	5476.5	71.8186	3.4333
53	0.53	0.47	3.0000	2.6862	1.2751	1.0000	5539.2	72.9277	3.4562
54	0.54	0.46	3.0000	2.7202	1.2932	1.0000	5602.7	74.0695	3.4794
55	0.55	0.45	3.0000	2.7547	1.3117	1.0000	5667.0	75.2463	3.5030
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
85	0.85	0.15	3.0000	3.0000	3.0000	1.0000	7395.8	121.5131	3.9545
86	0.86	0.14	3.0000	3.0000	3.0000	1.0005	7396.0	121.5209	3.9546
87	0.87	0.13	3.0000	3.0000	3.0000	1.1523	7442.5	124.1540	3.9827
88	0.88	0.12	3.0000	3.0000	3.0000	1.3319	7498.7	127.5789	4.0160
89	0.89	0.11	3.0000	3.0000	3.0000	1.5455	7567.1	132.0896	4.0556
90	0.90	0.10	3.0000	3.0000	3.0000	1.8029	7651.9	138.1593	4.1034
91	0.91	0.09	3.0000	3.0000	3.0000	2.1180	7759.1	146.5312	4.1618
92	0.92	0.08	3.0000	3.0000	3.0000	2.5106	7898.0	158.4110	4.2346
93	0.93	0.07	3.0000	3.0000	3.0000	2.9999	8079.3	175.4670	4.3253
94	0.94	0.06	3.0000	3.0000	3.0000	3.0000	8079.4	175.4708	4.3253
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

capacity. A tradeoff design should be chosen so that the two properties of the crash box are all satisfactory. The Pareto optimal solution in the fifty-five group is chosen to be a reference design, which is $[3.0000 \ 2.7547 \ 1.3117 \ 1.0000]^T$. And the corresponding performance is $[5667.0 \ 75.2463]^T$. Subjecting to the practical mechanical processing technology, the approximate design variable of the chosen Pareto optimal solution is adopted, which is $[3.0 \ 2.8 \ 1.3 \ 1.0]^T$.

Computer simulation is employed to examine the design. The obtained result of energy absorption is 5680.40 J and the maximum crashing force is 80.50 kN. For having a strong ability of energy absorption and accompanying acceptable maximum crashing force, the simulation results demonstrate that this design is fine. The simulation model before and after the collision is shown in Figure 12.

In order to examine the efficiency of the generate-first-choose-later method with the hybrid algorithm proposed in

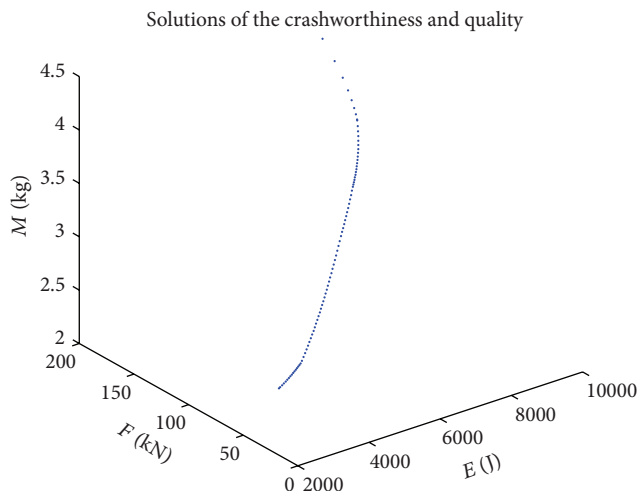


FIGURE 11: The referable properties and the corresponding mass.

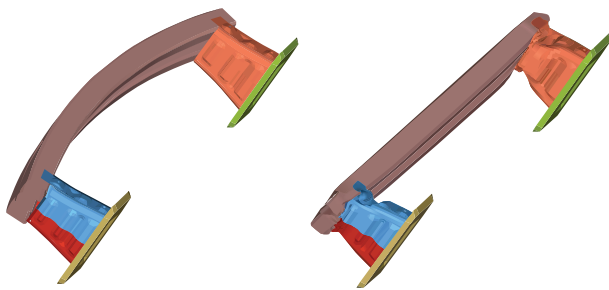


FIGURE 12: The structure comparison by crash simulation.

this paper, the engineering example is executed 20 times by the computer whose CPU is P8400. All the results in Table 2 as well as the above Pareto fronts are obtained within an average time of 4 minutes so that the method solves the multiobjective engineering optimization problem efficiently.

Overall, the method proposed in this paper for multiobjective engineering optimization problems can offer many effective suggestions to designer as a comprehensive reference, and the short computing time speeds up the design.

6. Conclusions

The proposed generate-first-choose-later method is an effective and efficient approach for multiobjective engineering optimization problems. In the example of crash box, this method gives some valuable reference to design the structure. Relying on the generated reference, the designers can understand the relationship between the design variables and the properties of structures. In addition, the preliminary shape design can be chosen from the optimal solutions.

According to the numerical examples and the engineering example, Algorithm 1 proposed in this paper can solve the multiobjective optimization problem with upper and lower bounds with high efficiency, and the Pareto optimal solutions as well as Pareto front are often obtained in a short time. So

the proposed method with Algorithm 1 can solve multiobjective engineering optimization problems rapidly.

For the weighted sum approach being adopted, this method has difficulty in searching the solutions when the Pareto curve is not convex. Improving Algorithm 1 to obtain more entire Pareto optimal solutions is the future research, so that more potential design advice can be offered.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The research is supported by the Fund for the National Natural Science Foundation of China (no. 50975121), the Doctoral Program of Higher Education (no. 20130061120035), the Plan for Scientific and Technology Development of Jilin Province (20130522150JH), and the Fund for Postdoctoral Scientific Research of Jilin Province (RB201337).

References

- [1] A. López-Jaimes and C. A. Coello Coello, "Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems," *Information Sciences*, vol. 277, pp. 1–20, 2014.
- [2] X. Gu, G. Sun, G. Li, X. Huang, Y. Li, and Q. Li, "Multiobjective optimization design for vehicle occupant restraint system under frontal impact," *Structural and Multidisciplinary Optimization*, vol. 47, no. 3, pp. 465–477, 2013.
- [3] A. Messac, C. Puemi-Sukam, and E. Melachrinoudis, "Aggregate objective functions and pareto frontiers: required relationships and practical implications," *Optimization and Engineering*, vol. 1, no. 2, pp. 171–188, 2000.
- [4] M. Zarepisheh, E. Khorram, and P. M. Pardalos, "Generating properly efficient points in multi-objective programs by the nonlinear weighted sum scalarization method," *Optimization*, vol. 63, no. 3, pp. 473–486, 2014.
- [5] G. Mavrotas, "Effective implementation of the ϵ -constraint method in Multi-Objective Mathematical Programming problems," *Applied Mathematics and Computation*, vol. 213, no. 2, pp. 455–465, 2009.
- [6] I. Das and J. E. Dennis, "Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.
- [7] I. Y. Kim and O. L. de Weck, "Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation," *Structural and Multidisciplinary Optimization*, vol. 31, no. 2, pp. 105–116, 2006.
- [8] R. S. Burachik, C. Y. Kaya, and M. M. Rizvi, "A new scalarization technique to approximate Pareto fronts of problems with disconnected feasible sets," *Journal of Optimization Theory and Applications*, vol. 162, no. 2, pp. 428–446, 2014.
- [9] G. Eichfelder, "Scalarizations for adaptively solving multiobjective optimization problems," *Computational Optimization and Applications*, vol. 44, no. 2, pp. 249–273, 2009.
- [10] J. Fliege, L. M. G. Drummond, and B. F. Svaiter, "Newton's method for multiobjective optimization," *SIAM Journal on Optimization*, vol. 20, no. 2, pp. 602–626, 2009.

- [11] N. Rastegar and E. Khorram, "A combined scalarizing method for multiobjective programming problems," *European Journal of Operational Research*, vol. 236, no. 1, pp. 229–237, 2014.
- [12] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization," in *Proceedings of the 5th International Conference on Genetic Algorithms (ICGA '93)*, San Mateo, Calif, USA, 1993.
- [13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [14] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength pareto evolutionary algorithm," in *Proceedings of the Conference on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problem*, K. C. Giannakoglou, D. T. Tsahalis, and J. Periaux, Eds., pp. 95–100, Barcelona, Spain, 2002.
- [15] C. A. C. Coello and M. S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, pp. 1051–1056, May 2002.
- [16] C. A. C. Coello and N. C. Cortés, "Solving multiobjective optimization problems using an artificial immune system," *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, 2005.
- [17] L. Nguyen, L. T. Bui, and H. Abbass, "A new niching method for the direction-based multi-objective evolutionary algorithm," in *Proceedings of the IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM '13)*, pp. 1–8, April 2013.
- [18] M. Shokouhifar and A. Jalali, "An evolutionary-based methodology for symbolic simplification of analog circuits using genetic algorithm and simulated annealing," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1189–1201, 2015.
- [19] A. Mukhopadhyay and U. Maulik, "Unsupervised pixel classification in satellite imagery using multiobjective fuzzy clustering combined with SVM classifier," *IEEE Transactions on Geoscience & Remote Sensing*, vol. 47, no. 4, pp. 1132–1138, 2009.
- [20] E. Acar, M. A. Guler, B. Gereker, M. E. Cerit, and B. Bayram, "Multi-objective crashworthiness optimization of tapered thin-walled tubes with axisymmetric indentations," *Thin-Walled Structures*, vol. 49, no. 1, pp. 94–105, 2011.
- [21] Á. Rubio-Largo, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez, "Multiobjective metaheuristics for traffic grooming in optical networks," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 457–473, 2013.
- [22] S. M. K. Heris and H. Khaloozadeh, "Open- and closed-loop multiobjective optimal strategies for HIV therapy using NSGA-II," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 6, pp. 1678–1685, 2011.
- [23] I. A. Griffin, A. Molina-Cristobal, P. Fleming, and D. Owens, "Multiobjective controller design: optimising controller structure with genetic algorithms," in *Proceedings of the IFAC World Congress on Automatic Control*, 2005.
- [24] J. Andersson, *A Survey of Multiobjective Optimization in Engineering Design*, Department of Mechanical Engineering, Linköping University, Linköping, Sweden, 2000.
- [25] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000.
- [26] C. Coello Coello, "Recent trends in evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*, A. Abraham, L. Jain, and R. Goldberg, Eds., pp. 7–32, Springer, London, UK, 2005.
- [27] M. Preuss, B. Naujoks, and G. Rudolph, "Pareto set and EMOA behavior for simple multimodal multiobjective functions," in *Parallel Problem Solving from Nature—PPSN IX*, T. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, J. D. Whitley, and X. Yao, Eds., vol. 4193 of *Lecture Notes in Computer Science*, pp. 513–522, Springer, Berlin, Germany, 2006.
- [28] P. L. Goethals and B. R. Cho, "Solving the optimal process target problem using response surface designs in heteroscedastic conditions," *International Journal of Production Research*, vol. 49, no. 12, pp. 3455–3478, 2011.
- [29] S. Simonenko, V. Bayona, and M. Kindelan, "Optimal shape parameter for the solution of elastostatic problems with the RBF method," *Journal of Engineering Mathematics*, vol. 85, no. 1, pp. 115–129, 2014.
- [30] I. Pan and S. Das, "Kriging based surrogate modeling for fractional order control of microgrids," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 36–44, 2015.
- [31] K. R. Frisch, *The Logarithmic Potential Method of Convex Programming*, University Institute of Economics, Oslo, Norway, 1959.
- [32] I. Das and J. E. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems," *Structural Optimization*, vol. 14, no. 1, pp. 63–69, 1997.
- [33] K. Sindhya, K. Miettinen, and K. Deb, "A hybrid framework for evolutionary multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 495–511, 2013.
- [34] C. Durazzi, "On the newton interior-point method for nonlinear programming problems," *Journal of Optimization Theory and Applications*, vol. 104, no. 1, pp. 73–90, 2000.
- [35] The MathWorks, *Genetic Algorithm and Direct Search Toolbox, MATLAB Version 2.4.1, User's Guide*, The MathWorks, 2009.
- [36] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the the 1st International Conference on Genetic Algorithms*, pp. 93–100, Hillsdale, NJ, USA, 1987.
- [37] Y.-W. Li, T. Xu, and T.-S. Xu, "Optimal design of energy-absorbing structure of autobody under low-speed crash," *Transaction of Beijing Institute of Technology*, vol. 30, no. 10, pp. 1175–1179, 2010.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

