A Nature-Inspired Multi-Objective Optimisation Strategy Based on a New Reduced Space Searching Algorithm for the Design of Alloy Steels

Qian Zhang and Mahdi Mahfouf
Department of Automatic Control and Systems Engineering,
The University of Sheffield,
Mappin Street, S1 3JD,
Sheffield, UK
E-mail addresses: Qian.Zhang@sheffield.ac.uk, M.Mahfouf@sheffield.ac.uk

Abstract:

In this paper, a salient search and optimisation algorithm based on a new reduced space searching strategy, is presented. This algorithm originates from an idea which relates to a simple experience when humans search for an optimal solution to a 'real-life' problem, i.e. when humans search for a candidate solution given a certain objective, a large area tends to be scanned first; should one succeed in finding clues in relation to the predefined objective, then the search space is greatly reduced for a more detailed search. Furthermore, this new algorithm is extended to the multi-objective optimisation case. Simulation results of optimising some challenging benchmark problems suggest that both the proposed single objective and multi-objective optimisation algorithms outperform some of the other well-known Evolutionary Algorithms (EAs). The proposed algorithms are further applied successfully to the optimal design problem of alloy steels, which aims at determining the optimal heat treatment regime and the required weight percentages for chemical composites to obtain the desired mechanical properties of steel hence minimising production costs and achieving the overarching aim of 'right-first-time production' of metals.

Keywords: Nature-Inspired Algorithm, Search Strategy, Reduced Space Searching, Multi-Objective Optimisation, Evolutionary Algorithms, Optimal Design, Alloy Steel, Mechanical Property, Tensile Strength

1. Introduction

In the steel industry, determining the optimal heat treatment regime and the required weight percentages for the chemical composites to obtain the desired mechanical properties of the steel is always a challenging multi-objective optimisation problem. Usually, some objectives may conflict with each other, such as the Ultimate Tensile Strength (UTS) and the ductility index, the Reduction of Area (ROA). In this paper, details relating to the optimal design of alloy steels are presented and discussed, which employs a salient nature-inspired optimisation technique, the Reduced Space Searching Algorithm (RSSA).

Inspired by natural and social behaviours, researchers have developed many successful optimisation algorithms. For example, the Genetic Algorithm (GA) (Holland, 1975; Goldberg, 1989) originates from the simulation of natural evolution, while the Particle Swarm Optimisation (PSO) (Kennedy and Eberhart, 1995) algorithm is motivated by the simulation of the social behaviour of birds flock. In the same way, a new search and optimisation algorithm, named Reduced Space Searching Algorithm (RSSA) throughout, is described in (Zhang and Mahfouf, 2007)[1], which is inspired by the simple human experience when searching for an 'optimal' solution.

Compared with conventional optimisation techniques, such as hill climbing (gradient descent), Newton's method and Quasi-Newton method, the proposed algorithm has the ability to tackle a wider spectrum of problems, for it does not need the information relating to derivatives, which is essential for the above conventional techniques. Thus, this new algorithm can deal with not only the well-defined but also the more complex, uncertain and ill-defined problems. Unlike most of the evolutionary and social inspired algorithms, such as Genetic Algorithm and Particle Swarm Optimisation, which are population-based algorithms, the proposed algorithm does not rely on defining a population of candidate solutions. This feature often enables the algorithm to perform faster and use less evaluation times to locate the final solutions. Furthermore, the most important difference between the proposed algorithm and other algorithms is the operation emphases within a search. Most of the optimisation algorithms concentrate on generating new solutions using various equations (derivative-related equations, PSO equations, etc.) or operators (mutation, recombination, etc.), while the new method concentrates on transforming the search space so as to find the 'optimal' sub-space. The generation of solutions within a sub-space does not constitute the real emphasis; in this paper, new solutions are created as a uniform random set of solutions. Hence, this proposed method aims to provide an alternative optimisation and search idea and inspire people to think in a different way when facing such optimisation problems.

The remaining parts of this paper are organised as follows. Section 2 introduces the Reduced Space Searching (RSS) strategy and outlines the explicit steps included in the proposed algorithm RSSA. In Section 3, RSSA is extended to include the multi-objective optimisation case and the details about this new multi-objective optimisation algorithm MO-RSSA are introduced. Section 4 presents the results of applying RSSA to optimise some well-known single objective benchmark functions. A comparative study between RSSA and other three evolutionary algorithms is also conducted. In Section 5, MO-RSSA is validated using the well-known ZDT (Zitzler et al., 2002) and DTLZ (Deb et al., 2001) series test problems. Section 6 describes how such algorithms perform within the realm of a real industrial application relating to the optimal design of

[1] The present paper includes an extended version of the algorithm originally published at the 2007 IEEE Congress on Evolutionary Computation (Zhang and Mahfouf, 2007).

mechanical properties of alloy steel. Finally, concluding remarks are given in Section 7.

2. Reduced Space Searching Algorithm (RSSA)

2.1 Reduced Space Searching (RSS) Strategy

If one approaches the optimisation issue from a totally unbiased angle it would be legitimate to postulate that 'common' sense should dictate that when searching for a candidate solution under predefined objectives, a relatively large search space area must be initially targeted. When clues are available that the objective may be met in a particular area, the initial search area is then justifiably reduced. This simple principle is being widely used in our every-day life and has proved to be effective. In the light of the above, a strategy of constructing a new optimisation algorithm, named Reduced Space Searching (RSS) throughout, is proposed.

The 'rationale' behind this RSS strategy is as follows: given an optimisation problem, one should divide the initial search space into parts and rank these parts according to the probability of the candidates satisfying the objective(s). First, a search is conducted in the partial space where the probability is the highest followed by the one with the lowest probability. The diagram of Figure 1 illustrates the idea behind the RSS strategy.

2.2 Basic Ideas behind RSSA

In order to develop an algorithm following the RSS strategy one must first define how to divide the search space into parts and how to rank such parts in terms of priorities. In this work, a simple (but no simpler) method to achieve this purpose is proposed. The basic idea is that the search space should initially be divided into two parts: one part being located around the best solution found so far while the other part should represent the space left. The partial space around the best solution should be top-ranked (the best). To simplify the method, the marginal partial space can be neglected and only the space that includes the best solution is kept for search purposes. If the process of dividing the search space into smaller parts is repeated a number of times, then a small search space as well as a relatively good solution to the problem will be obtained.

It was found that reducing the search space all the time is not the most effective way of locating the optimal solution. Sometimes, a too-small search space will decrease the speed of solution convergence and at the same time will reduce the probability of the solution jumping out of the local optimum. Thus, a search space 'increase' mechanism is proposed to cooperate with the original 'decrease' mechanism. In this new mechanism, if no better solution can be found in the optimisation search process, then the search space is reckoned to be too large for such a search and should be decreased to reinforce the local search. If better solutions can always be found in a particular reduced space, then the algorithm may certainly have got trapped in a 'local optimum' area.

Given this situation, the search space should be increased to reinforce the global search. It is worth noting that this proposed method attempts to strike a balance between the 'global' and the 'local' searches to make the optimisation search process more adaptive.

Figure 2 shows an example of the size of the search space decreasing or increasing in a two-dimensional problem. In Figure 2(a), the rectangular Region 1 is the search space of an optimisation problem. Solution 'A' is the best solution hitherto located. If there are several continuous randomly selected candidate solutions worse than 'A' in the fitness to the optimisation problem, as is shown in Figure 2(b), then the size of the search space should be decreased around the best solution 'A'. The partial space (Region 2) containing 'A', as the centre, is set to the new space one should perform the search in. On the other hand, if there are several continuous randomly selected candidate solutions better than 'A' in the fitness, which is shown in Figure 2(c), then the size of the search space should be increased around the best solution 'B'. The increasing space (Region 3) containing 'B', as the centre, is set to the new space one should perform the search in. If there are better solutions (but not continuous) that can be found in the search space (shown in Figure 2(d)), then the size of the search space should not be changed and the centre of the search space (Region 4) will be moved to the new best solution 'C'.

In the light of the above considerations, a good solution is obtained after a finite number of iterations. However, it must be stressed that the above method may only find a local optimal solution rather than a global optimal one. In the case of a crooked, multimodal fitness landscape, the RSS operator might lead to premature exit from the region where the global optimum actually belongs. To solve this problem and obtain the global optimal solution, a variation operator is employed to cooperate with the RSS operator. Figure 3 shows the flow chart of the overall RSSA algorithm.

Three variation strategies are designed as follows:

1. One-dimensional variation: Only one element of the decision variable vector will be varied. The position of this element will be randomly chosen and the element will be set at a random value within the search bounds.
2. Multi-dimensional variation: The number of elements of the decision variable that will be varied and the positions of these elements will be randomly generated. These elements will then be set to some random values.
3. All-dimensional variation: All the elements of the decision variable vector will be randomly varied.

The RSS strategy introduced above is based on the concept of splitting the search (decision) space into sub-spaces. It is worth noting that some literatures (Zhang and Mahfouf, 2006; Chakraborti et al., 2008) also proposed the methods of splitting the functional (objective) space. For example, in (Chakraborti et al., 2008), a multi-

objective optimisation genetic algorithm was developed using a neighbourhood concept. It splits the functional space into discrete grids and each candidate solution is mapped to one grid. A neighbourhood is assigned to each solution based on its functional grid position. Then, a genetic recombination is only conducted between the solution and one of its neighbours.

2.3 The RSSA Algorithm

Consider a single objective optimisation problem with N decision variables as follows: Minimise $f(X)$, $X \in [X\min_1, X\max_1] \times [X\min_2, X\max_2] \times \ldots \times [X\min_N, X\max_N]$.

The proposed RSSA algorithm can be summarised as follows:

1. Randomly select one candidate solution Xa ($x_1, x_2, \ldots, x_N$) in the original search space and save it as the best solution $Xbest = Xa$. Set $n = 0$, which is used to control the bounds of the search space.
2. Randomly select the candidate solutions Xb(s) in the current search space. If C_1-continuous Xb(s) satisfies $f(Xb) < f(Xbest)$ and n > 1, then $Xbest = Xb$ and $n = n - 1$. If C_2-continuous Xb(s) satisfies $f(Xb) > f(Xbest)$, then $n = n + 1$. If non-continuous Xb(s) satisfies $f(Xb) < f(Xbest)$, then $Xbest = Xb$.
3. Change the size of the search space using the ratio K ($0 < K < 1$, in this paper $K = 0.5$ without any loss of generality). $Xbest$ is located at the centre of the new space. $Y\min_i$ is the lower bound of the ith decision variable in the new search space and $Y\max_i$ is the upper bound. To avoid the new bounds stepping outside the original constraints, the following equations are used to define the new bounds:
$$Y\min_i = \max\left(X\min_i, Xbest(i) - K^n L(i)\right),$$
$$Y\max_i = \min\left(X\max_i, Xbest(i) + K^n L(i)\right). \quad (1)$$
where $i = 1, 2, \ldots N$; $0 \le n \le m$; $L(i) = X\max_i - X\min_i$. m is a threshold value that depends on the precision needed and relates to the value of K. If $K = 0.5$, a value of $m = 15$ to 30 should prove adequate. For example, if $m = 20$ and $K = 0.5$, the search space can be reduced to as small as $(K)^m = (1/2)^{20}$ ($\approx$ 9.54e-7) of its original space. It also means the solution obtained in this sub-space have a precision close to 1e-6 of its value range for a decision variable.
4. Repeat Steps 2 and 3 until $n = m$.
5. Perform the variation operator on $Xbest$ and obtain Xc. If $f(Xc) < f(Xbest)$, then $Xbest = Xc$, $n = 0$ and repeat Steps 2 to 4.
6. Repeat Step 5 until the 'optimal' solution is found or the termination criterion is reached.

It is worth noting that the decreasing parameter C_1 and the increasing parameter C_2 play important roles in the RSSA algorithm. They are used to balance the 'global' search as

well as the 'local' search in the optimisation process. In (Zhang, 2009), experiments were carried out to investigate the influence of C_1 and C_2 general settings. It is generally recommended that $C_1 = C_2 \times (D/2 + 8)$, where D is the dimension of the test problem.

3. Extension of RSSA to Multi-Objective Optimisation Problems

To extend the RSSA algorithm for optimising multi-objective problems, the Random Weighted Aggregation (RWA) technique (Murata et al., 1996) is employed and an archiving approach is also included to preserve the Pareto-optimal solutions.

3.1 The Random Weighted Aggregation Approach

Assume a multi-objective problem that consists of finding a vector

$$X^* = (x_1^*, x_2^*, x_3^*, \ldots, x_D^*) \tag{2}$$

that will optimise the following vector function:

$$\vec{f}(X) = [f_1(X), f_2(X), f_3(X), \ldots, f_k(X)]. \tag{3}$$

The Weighted Aggregation is one of the most common approaches for solving multi-objective problems. In this type of approach, all the objectives are summed to a weighted combination as follows:

$$F = \sum_{i=1}^{k} w_i f_i(X), \ \sum_{i=1}^{k} w_i = 1 \tag{4}$$

where w_i, $i = 1, 2, \ldots, k$, are non-negative weights.

In the Conventional Weighted Aggregation (CWA) method, the above weights are fixed during the optimisation process. By using CWA, only a single Pareto-optimal solution can be obtained in every optimisation run. If one wishes to obtain different Pareto solutions, the algorithm has to be repeated several times with different weights settings. In addition, this method cannot locate the Pareto solutions when there are concave regions in the true Pareto front.

Random Weighted Aggregation (RWA) can overcome the limitations of CWA. In the RWA method, the weights are modified after every certain number of iterations during the optimisation. The weights are defined by the following equation:

$$w_i(t) = \begin{cases} \dfrac{rand_i(t)}{\sum_{j=1}^{k} rand_j(t)}, & if \ rem(t, H) = 1; \\ w_i(t-1), & else. \end{cases} \tag{5}$$

where t is the index of iteration and H is the frequency of the weight changing; $rand_i(t)$ is a function to create a uniformly distributed random value in the range [0, 1]; $rem(t, H)$ is a function to obtain the remainder from dividing t by H.

In this paper, the frequency parameter is calculated using the following equation:

$$H = round\left(E_{\max} / (4 * Obj)\right) \tag{6}$$

where $round(x)$ is the function that allows to round-off x to the nearest integer, $E_{\max}$ is the maximum function evaluation number and Obj is the number of objectives. Equation (6) aims to calculate an H, which can control the objective weights to vary ($4*Obj$) times during the whole optimisation procedure.

As far as recently-developed multi-objective optimisation algorithms, one can distinguish them into two categories considering the fitness assignment strategies, which are weighted-aggregation-based and Pareto-dominance-based. Most of the algorithms are based on Pareto-dominance concept. They have demonstrated their capability in finding a well-converged and well-distributed set of near Pareto-optimal solutions (Zitzler and Thiele, 1998; Knowles and Corne, 2000; Deb et al., 2002). However, recent studies have discovered that the Pareto-dominance-based algorithms may face some difficulties in solving the problems with a large number of objectives, because the emphasis of all non-dominated solutions keeping in the population may not produce enough selection pressure for the population to move towards the Pareto-optimal region fast enough (Deb et al., 2006). While the algorithms based on the varying weighted aggregation strategies (Jin et al., 2001; Chang et al., 2002) were shown to be computationally efficient. For some specific multi-objective optimisation problems, such as those which consist of finding the solutions near the desired region of decision-maker's interest (Deb et al., 2006) or the problems to find the 'knees' (Branke et al., 2004) out of all possible Pareto-solutions, the weighted-aggregation-based algorithms are more practical and relatively more straightforward. It is worth noting at this stage that the weighted aggregation strategy has been investigated further in other algorithms, such as the predator-prey approach (Li, 2003; Pettersson et al., 2007), where the 'preys' employ the weighted aggregation method to assign fitness values. To maintain diversity between generated solutions, a 'prey' uses different sets of weights when facing different 'predators'.

3.2 Archive Design

In the RWA method, the population cannot keep all the found Pareto solutions. Thus, an archive is used to record the Pareto solutions found so far during the optimisation search. To update the archive with appropriate Pareto solutions, a non-dominated selection and a diversity selection mechanism are employed. The non-dominated selection aims to obtain the Pareto-optimal solutions from the candidates. This is easy to implement. The diversity selection tends to obtain the solutions with a good diversity from the candidates. In this paper, a simple method is proposed to achieve this purpose, which works as follows:

1. If the number of solutions in the present archive is more than the predefined maximum number, go to Step 2; else terminate this selection and return.

2 For every solution in the archive, calculate the value of its closeness criterion. The closeness criterion of the ith solution is defined as follows:

$$cri_i = d_{i1} + d_{i2} \tag{7}$$

where d_{i1} is the distance between the ith solution to its closest neighbour and d_{i2} is the distance between the ith solution to its second closest neighbour.

3 Find the solution with the minimum criterion value and remove it from the archive.

4 Go to Step 1.

In this paper and for this particular application, the size of the archive is set to be 100.

3.3 Algorithm Formulation

By applying the RWA method and by maintaining an archive for preserving the Pareto-optimal solutions, the RSSA is thus extended to a multi-objective optimisation algorithm, named as the Multi-Objective Reduced Space Searching Algorithm (MO-RSSA). In summary, the entire MO-RSSA can be described via the following procedure:

1 Randomly generate the initial weights for the optimisation objectives.

2 Optimise the related problem, whose objective is the weighted sum of the multiple objectives, using RSSA for one iteration.

3 Add the present best position to the archive as the candidate solution.

4 Execute the non-dominated selection to the archive.

5 Execute the diversity selection to the archive.

6 Vary the weights of the objectives using the method RWA.

7 Repeat Step 2 to Step 6 until a stopping criterion (e.g., a maximum number of iterations or a sufficiently good fitness value) is achieved.

4 Experimental Studies using RSSA

4.1 Benchmark Test Functions

In the field of evolutionary computation, it is common to compare different algorithms using a large test set. When an algorithm is evaluated, one must look for the type of problems where its performance is good, in order to characterise the type of problems for which the algorithm is suitable. In this work, the test set with some well-characterised functions is used as it allows one to obtain and generalise the results regarding the kind of functions involved. All these functions are used as minimisation problems and the following shows their expressions and the summary of their features about separability and multimodality.

1. Sphere function (Unimodal, Separable and D-dimensional):

$$f_1(x) = \sum_{i=1}^{D} x_i^2 \,,\ x_i \in [-10, 10]\,,\ \min(f_1) = f_1(0,...,0) = 0\,.$$

2. Schwefel's function 2.22 (Unimodal, Non-separable and D-dimensional):

$$f_2(x)=\sum_{i=1}^{D}|x_i|+\prod_{i=1}^{D}|x_i|,\ x_i\in[-10,10],\ \min(f_2)=f_2(0,...,0)=0.$$

3. Schwefel's function 1.2 (Unimodal, Non-separable and *D*-dimensional):

$$f_3(x)=\sum_{i=1}^{D}\left(\sum_{j=1}^{i}x_j\right)^2,\ x_i\in[-10,10],\ \min(f_3)=f_3(0,...,0)=0.$$

4. Schwefel's function 2.21 (Unimodal, Non-separable and *D*-dimensional):

$$f_4(x)=\max_i\{|x_i|,1\le i\le D\},\ x_i\in[-10,10],\ \min(f_4)=f_4(0,...,0)=0.$$

5. Rosenbrock's function (Multimodal, Non-separable and *D*-dimensional):

$$f_5(x)=\sum_{i=1}^{D-1}\left(100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right),\ x_i\in[-2,2],\ \min(f_5)=f_5(1,...,1)=0.$$

6. Schwefel's function 2.26 (Multimodal, Separable and *D*-dimensional):

$$f_6(x)=-\sum_{i=1}^{D}\left(x_i\sin\left(\sqrt{|x_i|}\right)\right),\ x_i\in[-500,500],$$

$$\min(f_6)=f_6(420.9687,...,420.9687)=-12569.5.$$

7. Rastrigin's function (Multimodal, Separable and *D*-dimensional):

$$f_7(x)=\sum_{i=1}^{D}\left(x_i^2-10\cos(2\pi x_i)+10\right),\ x_i\in[-5,5],\ \min(f_7)=f_7(0,...,0)=0.$$

8. Ackley's function (Multimodal, Non-separable and *D*-dimensional):

$$f_8(x)=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)+20+e,$$

$$x_i\in[-30,30],\ \min(f_8)=f_8(0,...,0)=0.$$

9. Griewank's function (Multimodal, Non-separable and *D*-dimensional):

$$f_9(x)=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1,\ x_i\in[-600,600],$$

$$\min(f_9)=f_9(0,...,0)=0.$$

10. Bohachevsky's function (Multimodal, Separable and *D*-dimensional):

$$f_{10}(x)=\sum_{i=1}^{D-1}\left(x_i^2+2x_{i+1}^2-0.3\cos(3\pi x_i)-0.4\cos(4\pi x_{i+1})+0.7\right),\ x_i\in[-15,15],$$

$$\min(f_{10})=f_{10}(0,...,0)=0.$$

11. Schaffer's function (Multimodal, Non-separable and *D*-dimensional):

$$f_{11}(x)=\sum_{i=1}^{D-1}(x_i^2+x_{i+1}^2)^{0.25}\left(\sin^2\left(50(x_i^2+x_{i+1}^2)^{0.1}\right)+1.0\right),\ x_i\in[-100,100],$$

$$\min(f_{11})=f_{11}(0,...,0)=0.$$

12. Six-hump Camel-Back function (Multimodal, Non-separable and 2-dimensional):

$$f_{12}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4,\ x_i\in[-5,5],$$

$\min(f_{12}) = f_{12}(0.08983,-0.7126) = f_{12}(-0.08983,0.7126) = 0$.

13. Branin function (Multimodal, Non-separable and 2-dimensional):

$$f_{13}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10,$$

$x_1 \in [-5, 10]$, $x_2 \in [0, 15]$, $\min(f_{13}) = f_{13}(-3.142,12.275)$

$= f_{13}(3.142,2.275) = f_{13}(9.425,2.425) = 0.398$.

14. Goldstein-Price function (Multimodal, Non-separable and 2-dimensional):

$$f_{14}(x) = \left(1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right)$$

$$\times\left(30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right),\ x_i \in [-2, 2],$$

$\min(f_{14}) = f_{14}(0,-1) = 3$.

A function of D variables is separable if it can be rewritten as a sum of D functions of just one variable. Non-separable functions are more difficult to optimise as the accurate search direction depends on two or more variables. On the other hand, separable functions can be optimised for each variable in turn. A function is multimodal if it has two or more local optima. The problem is more difficult if the function is multimodal. The search process must be able to avoid the regions around local optima in order to approximate, as far as possible, the global optimum.

4.2 Effects of the Variation Strategies

Three types of variation operators were tested and compared in this experiment. For this purpose, the 30-dimensional multimodal benchmark problems f_5 to f_{11} were used as test beds. The decreasing parameter C_1 was set to be 23 and the increasing parameter C_2 was set to be 1 (Zhang, 2009). For each setting, 20 runs were conducted. In each run, the maximal function evaluation number was set to 10^6 and the optimisation process was regarded as successful and stopped, when the best solution Fb satisfied the following condition: $Fb < 10^{-5}$ if the true global minimum $Gb = 0$ or $|(Fb - Gb) / Gb| < 10^{-5}$ if $Gb \neq 0$.

From Table 1, it can be seen that the one-dimensional variation strategy performs best on the functions f_5, f_6, f_7, f_8, f_{10} and f_{11}, while the all-dimensional variation strategy performs best on the problems f_9. For a broad adaptation to various problems, it is recommended to use both the one-dimensional and multi-dimensional variation strategies simultaneously.

4.3 A Comparison between RSSA and Other Evolutionary Algorithms

In this section, experiments were carried-out between RSSA and other three salient evolutionary algorithms, which are the Covariance Matrix Adaptation Evolution

Strategy (CMA-ES) (Hansen et al., 2003), the Differential Evolution (DE) (Storn and Price, 1995) and the Generalised Generation Gap model with the Parent-Centric Recombination operator (G3+PCX) (Deb et al., 2002).

The parameter settings for these algorithms are described as follows:

1. RSSA: $C_1 = D/2 + 8$, $C_2 = 1$ (Zhang, 2009), $K = 0.5$, $m = 30$, where D is the dimension of the test problem. The variation operator worked as a combination of the one-dimensional variation strategy (with the 50% probability of usage) and the multi-dimensional variation strategy (with the 50% probability of usage).
2. CMA-ES: There are 8 parameters to be predefined for this algorithm. All settings followed the instructions given in (Hansen, 2007). For instance, the population size $\lambda = 4 + floor(3\times\ln D)$, the parent number $\mu = floor(\lambda/2)$, etc., where $floor(x)$ is the function that allows to round-off x to the nearest integer towards $-\infty$.
3. DE: The DE/Rand/1 scheme was employed. The parameter settings followed the instructions in (Storn, 1996). The population size $N = 10\times D$; the crossover probability $CR = 0.9$ and the weighting factor $F = 0.8$.
4. G3+PCX: Following the papers by (Deb et al., 2002; Deb, 2005), the population size $N = 10 \times D$; the parent size was set to 3; the offspring size was set to 2 and the replacement size was set to 2. For the PCX operator, the distribution parameter $\sigma_\zeta = 0.1$ and $\sigma_\eta = 0.1$.

The optimisation process was regarded as successful and stopped when the best solution Fb satisfied the following condition: $Fb < 10^{-5}$ if the true global minimum $Gb = 0$ or $|(Fb - Gb) / Gb| < 10^{-5}$ if $Gb \neq 0$. For every individual experiment, the result was obtained after 20 runs. For each run, the maximal function evaluation number was set to 10^6.

Table 2 shows the optimisation results of different algorithms on various problems. From this table, one can observe the following:

1. For the unimodal problems f_1 to f_4, CMA-ES performs best in most of the situations. RSSA performs best using the fewest function evaluation for f_1. For f_3, RSSA can achieve the minimum with a small function evaluation number, but it cannot obtain the optima of the problems f_2 and f_4.
2. For the high-dimensional multimodal problems f_5 to f_{11}, RSSA performs better than other algorithms. For instance, for f_7, f_8 and f_{10}, RSSA is able to locate the global optimum with the fewest function evaluations; for f_6 and f_9, RSSA performs better than the other algorithms. In most of the situations, RSSA can achieve the optima, while other algorithms often cannot find the 'true' optimal solutions.
3. For the low-dimensional multimodal problems f_{12} to f_{14}, RSSA is able to obtain the global optimum and needs fewer function evaluations, compared with other

algorithms.

5. Experimental Studies using MO-RSSA

To validate the effectiveness of the proposed multi-objective optimisation algorithm, a set of experimental tests were carried-out using the well-known multi-objective optimisation problems - the ZDT series benchmark problems (Zitzler et al., 2000) and DTLZ series problems (Deb et al., 2001).

5.1 ZDT Series Benchmark Problems

The ZDT series benchmark functions include 2 minimisation objectives and they are considered to be difficult to optimise, especially ZDT2, ZDT3 and ZDT4 (Zitzler et al., 2000). The maximal function evaluation for every experiment was set to 25000, which is the same as the experiments configuration referred to the experiments in (Deb, 2001). The configuration of the algorithm was set as follows: decreasing parameter $C_1 = 3$, increasing parameter $C_2 = 1$, changing ratio $K = 0.5$, $m = 15$, frequency parameter $H = 10000$ and a variation strategy of the combination of the one-dimensional variation (with the 75% probability of usage) and the multi-dimensional variation (with the 25% probability of usage).

Figure 4 shows the graphical results produced by MO-RSSA. The true optimal Pareto fronts of the problems are represented with a continuous 'red' curve and the 'round' dots are the solutions obtained using the new algorithm. It can be observed that the algorithm possesses very good convergence properties while maintaining a good diversity among the Pareto solutions. Compared with the optimisation results in (Deb, 2001), which used PAES (Knowles and Corne, 2000), SPEA (Zitzler and Thiele, 1998) and NSGA-II (Deb et al., 2002), MO-RSSA performs as well as and sometimes better than the other three salient EAs in terms of both accuracy and diversity.

5.2 DTLZ Series Benchmark Problems

In the second experiment, MO-RSSA was used to optimise the DTLZ series problems (Deb et al., 2001). All the DTLZ problems were set so as to include three objectives. For a meaningful comparison, MO-RSSA used the same numbers of function evaluations as the experiments in (Deb et al., 2001). The parameters of the algorithm were set the same as the previous experiments, except the weight changing frequency parameter H, which is now taken to be 1000. Figure 5 shows the 3-D Pareto fronts obtained by MO-RSSA. It can be seen that, in most of the situations, the algorithm can convergence to the real Pareto-optimal front with a good diversity among the solutions. Compared with the optimisation results in (Deb et al., 2001), MO-RSSA performs as well as and more often than not better than the salient EAs, SPEA2 and NSGA-II, both

in terms accuracy and diversity.

6. Applications in Alloy Steel Design

In recent years, multi-objective optimisation techniques have been applied to the design of alloys, including steels (Mahfouf et al., 2005), superalloys (Egorov-Yegorov et al., 2005), bulk metallic glasses (Dulikravich et al., 2008), based on the developed intelligent models. Researchers have also employed multi-objective optimisation techniques in the structural material design based on the interatomic potentials (Chakraborti et al., 2009) or interionic potentials (Sreevathsan et al., 2009). In this work, the proposed algorithms, which perform very well on benchmark problems, were further applied to the optimal design of alloy steels for achieving the overarching aim of 'right-first-time production' of metals (Mahfouf et al., 2009).

In the steel industry, determining the optimal heat treatment regime and the required weight percentages for the chemical composites to obtain the desired mechanical properties of the steel is always a challenging multi-objective optimisation problem. Usually, some objectives may conflict with each other, such as the ultimate tensile strength (UTS) and the ductility. The steel ductility can also be reflected by its Reduction of Area (ROA).

Previously published research included the development of intelligent models based on fuzzy systems in order to predict the mechanical test results for the steels characterised by a wide range of training data (Zhang and Mahfouf, 2008). These models can be used to facilitate the findings relating to the optimal heat treatment regime and the weight percentages for the chemical composites to obtain the desired mechanical properties. Figure 6 shows the prediction results of one UTS model and one ROA model. These two models include the same 15 input variables, which are the weight percentages for the chemical composites, namely Carbon (C), Silica (Si), Manganese (Mn), Sulphur (S), Chromium (Cr), Molybdenum (Mo), Nickel (Ni), Aluminium (Al) and Vanadium (V), the test depth, the size and the site of the alloy steel, the cooling medium, as well as the hardening and tempering temperatures. They were developed based on 3760 and 3710 industrial data sets, respectively. In the following studies, all alloy design experiments are conducted using the two developed fuzzy models mentioned above. Figure 7 shows the strategy how robust prediction models can be exploited in a reverse-engineering fashion to identify 'optimal' recipes for system design. The parameter configurations of the algorithms were similar to the ones set in the experiments already described in the previous sections.

6.1 The Optimal Design of UTS

In this case, the aim is to find the optimal solution for achieving a predefined target

UTS value. The decision vector consists of weight percentages for the chemical composites, namely Carbon (C), Silica (Si), Manganese (Mn), Sulphur (S), Chromium (Cr), Molybdenum (Mo), Nickel (Ni), Aluminium (Al) and Vanadium (V), the test depth, the size and the site of the alloy steel, the cooling medium, as well as the hardening and tempering temperatures.

The objective function was designed to be as follows:

$$\text{Minimise } J_1 = \left(\frac{UTS - UTS_{Target}}{900} \right)^2 \tag{8}$$

where UTS_{Target} is the target UTS value.

In this experiment, the UTS_{Target} was set to 900 MPa. Figure 8 shows the optimisation process and Table 3 provides the optimisation results relating to 10 different runs. The average function evaluation number used in the RSSA algorithm is only 36. From Table 3, it can be seen that the differences between the 10 solutions are somewhat stark, which means that there are many possible solutions satisfying the same defined objective.

6.2 The Optimal Design of ROA

In this section, details relating to finding the optimal solution for achieving a predefined target ROA value are presented. In this case, the decision vector is the same as the one used for the UTS design problem in Section 6.1. The optimisation objective function was designed as follows:

$$\text{Minimise } J_2 = \left(\frac{ROA - ROA_{Target}}{60} \right)^2 \tag{9}$$

where ROA_{Target} is the target ROA value.

In the first experiment, the ROA_{Target} was set to 60%. Table 4 provides the optimisation results for 10 different runs and Figure 9 shows the variation of the average fitness of these 10 runs during the optimisation process. The average function evaluation number used in the RSSA algorithm is only 28.

6.3 The Optimal Design of both UTS and ROA

In the design of alloy steels, sometimes it is required to achieve a predefined target UTS value and a predefined target ROA value simultaneously. For this problem, one should first judge whether such requirements are possible. If the answer is ‘yes’, then the problem can be solved as a single objective optimisation problem by combining these two objectives into a weighted sum formulation. However, if the answer is ‘no’, then the problem should be solved using the multi-objective optimisation technique, which is able to offer a set of approximate candidate solutions (Pareto-optimal solutions). In

order to ascertain both scenarios, the achievable minimum and maximum boundaries are needed. Such boundaries will, as a result, act as guide to the search for the Pareto fronts and as a result will speed up the optimisation search outcome.

In this section, the decision vector of these design problems consists of weight percentages of Carbon (C), Manganese (Mn), Chromium (Cr), Molybdenum (Mo), and tempering temperature.

6.3.1 Boundaries for the UTS and ROA Design

To obtain the mechanical property boundaries for alloy steels design, the multi-objective optimisation technique was employed. Two distinct relevant multi-objective optimisation problems were defined as follows:

1. Minimising UTS and ROA simultaneously, i.e.:
   Objective 1: Minimise UTS
   Objective 2: Minimise ROA (10)
2. Maximising UTS and ROA simultaneously, i.e.:
   Objective 1: Maximise UTS
   Objective 2: Maximise ROA (11)

The MO-RSSA algorithm was employed to optimise the above problems and the maximum function evaluations number was set to 10,000. The obtained Pareto fronts using MO-RSSA are displayed in Figure 10. The region between the two fronts is where one can design the properties (UTS and ROA).

6.3.2 The Single Objective Optimisation

If the target UTS and ROA are located between the design boundaries, then the single objective optimisation technique can be used to obtain the desired solution by optimising the following objective function:

$$\text{Minimise } J_3 = \left(\frac{UTS - UTS_{Target}}{900}\right)^2 + \left(\frac{ROA - ROA_{Target}}{60}\right)^2 \quad (12)$$

where UTS_{Target} is the target UTS value and ROA_{Target} is the target ROA value.

For instance, if UTS_{Target} is 900 MPa and ROA_{Target} is 60%, it can be seen from Figure 10 that the targets are located between the design boundaries. Table 5 shows the results of applying RSSA to optimise Problem (12) for 10 different runs. The average number of function evaluations needed for these 10 runs is 133.

6.3.3 The Multi-Objective Optimisation

If the target UTS and ROA are located outside the design boundaries, then no precise solutions can be found to satisfy the desired targets. In this case, the multi-objective optimisation technique can be used to obtain a set of Pareto-optimal solutions, which are regarded as the possible candidate solutions. The design problem can be described as follows:

$$\text{Objective 1: Minimise } J_1 = \left(\frac{UTS - UTS_{Target}}{900} \right)^2$$

$$\text{Objective 2: Minimise } J_2 = \left(\frac{ROA - ROA_{Target}}{60} \right)^2 \tag{13}$$

where UTS_{Target} is the target UTS value and ROA_{Target} is the target ROA value.

For example, if the design targets UTS_{Target} is 600 MPa and ROA_{Target} is 50%, then from Figure 10 it can be seen that the targets are beyond the lower design boundary. In this type of a situation, the multi-objective optimisation algorithm MO-RSSA should suitably be employed to optimise the above Problem (13) with a maximum function evaluations number being set to 10,000 for instance. The obtained Pareto-optimal solutions are shown in Figure 11 and Table 6 provides details of 10 of these solutions. For those users who tend to prioritise 'hardness' more, they could choose a design that is close to the target UTS. For those users who are more concerned with ductility, they may choose a design that is close to the target ROA. Finally, for those users who have no preference between hardness and ductility, a 'median' design, whereby the mechanical properties are relatively close to the target values, may be the suitable choice.

From the experimental results in this section, the following can be observed:

1. For an optimal design problem with two conflicting targets, MO-RSSA is able to find the design boundaries, which is used to ascertain two different design scenarios.
2. If the target values are located between the design boundaries, RSSA can be used to obtain the desired precise solutions successfully.
3. If the target values are located outside the design boundaries, MO-RSSA can be used to obtain a set of approximate candidate solutions (Pareto-optimal solutions) successfully.

6.4 The Optimal Alloy Design Considering both the Mechanical Properties and the Economical Factors

This study consists of finding the optimal chemical compositions and heat-treatment process parameters in order to obtain the required UTS and ROA while minimising the production costs. The production costs of heat-treated steels include the costs of the

addition of alloying elements, such as Cr, Mo, V, etc. and the costs of energy consumption during the heat-treatment process.

In this experiment, five decision variables, C, Mn, Cr, Mo and Tempering Temperature, have been considered although other composites and temperatures could also be included. The factors contributing to the cost of heat treatment operation are summarised in Tables 7, 8 (Mahfouf et al., 2002).

6.4.1 The Optimal Design Considering both UTS and the Cost

According to the contribution of the chemical composites and the tempering process to the cost of heat-treated steels, a new objective function to reflect such costs was introduced as follows:

$$J_{\cos t} = \left(\frac{18Mn + 42Cr + 52Mo + 4.88Temp/600}{100} \right)^2 \tag{14}$$

By taking into account such economic consideration, the problem of designing an alloy steel with a predefined target UTS property becomes a two-objective optimisation problem described as follows:

$$\text{Objective 1: Minimise } J_1 = \left(\frac{UTS - UTS_{T\arg et}}{900} \right)^2$$

$$\text{Objective 2: Minimise } J_{\text{cost}} \tag{15}$$

Figure 12 displays the obtained Pareto-optimal solutions in the objective space with the UTS target value UTS_{Target} = 900 (MPa). Ten various solutions around the UTS target value are selected from the Pareto-optimal solutions and listed in Table 9.

6.4.2 The Optimal Design Considering both ROA and the Cost

By considering both the ROA and the economical factors, the following two-objective optimisation problem can be set:

$$\text{Objective 1: Minimise } J_2 = \left(\frac{ROA - ROA_{T\arg et}}{60} \right)^2$$

$$\text{Objective 2: Minimise } J_{\text{cost}} \tag{16}$$

Figure 13 shows the obtained Pareto-optimal solutions in the objective space, where the ROA target value ROA_{Target} is 60%. Ten different solutions around the ROA target value are selected from the Pareto-optimal solutions and listed in Table 10.

6.4.3 The Optimal Design Considering UTS, ROA and the Cost

Taking into account all the three factors, i.e. UTS, ROA and the cost of the heat treatment, the problem of designing an alloy steel can be described as follows:

$$\text{Objective 1: Minimise } J_1 = \left(\frac{UTS - UTS_{Target}}{900} \right)^2$$

$$\text{Objective 2: Minimise } J_2 = \left(\frac{ROA - ROA_{Target}}{60} \right)^2$$

$$\text{Objective 3: Minimise } J_{\text{cost}} \tag{17}$$

An optimisation experiment has been conducted based on the above objectives where the target values UTS_{Target} = 900 (MPa) and ROA_{Target} = 60 (%). The result of this experiment is shown in Figure 14. Ten solutions out of all the obtained Pareto-optimal solutions are selected and listed in Table 11.

From the above experiments, it can be seen that, for the optimal design problems that consider both the mechanical properties and the economical factors, MO-RSSA is able to obtain a set of optional solutions (Pareto-optimal solutions), which are close to the predefined UTS and/or ROA targets while providing various levels of heat treatment costs.

7. Conclusion

In this paper, a new optimisation algorithm RSSA was introduced, which is inspired from the simulation of the simple human societal behaviour when searching for optimal solutions in our daily routines. This new algorithm has been validated using a set of well-known benchmark problems. Compared with the recently developed and most salient optimisation algorithms, CMA-ES, DE and G3-PCX, RSSA performs as well as and sometimes better than these algorithms. RSSA was then extended to the multi-objective optimisation case, in which the random weighted aggregation was employed and an archive was maintained for preserving the suitable Pareto-optimal solutions. The experimental results of optimising some challenging problems ZDT and DTLZ series problems show that the proposed MO-RSSA perform as well as the other well-known EAs, such as PAES, SPEA and NSGA-II.

Furthermore, RSSA and MO-RSSA have been successfully applied to single objective and multi-objective optimal design of alloy steels. This research aims at determining the optimal heat treatment regime and the required weight percentages for the chemical composites to obtain the desired mechanical properties of steel such as UTS and ROA. In addition, the work was later extended to include economic factors, such as the costs associated with the composites and the tempering operation. The simulation results showed that MO-RSSA is able to produce a range of well-spread optional solutions

around the property targets while maintaining reasonable production costs.

Acknowledgement

We thank the anonymous reviewers for their useful comments which helped to improve the quality of this paper.

References:

Branke, J., Deb, K., Dierolf, H., Osswald, M., 2004. Finding Knees in Multi-objective Optimization. Lecture Notes in Computer Science Volume 3242, 722-731.

Chakraborti, N., Siva Kumar, B., Satish Babu, V., Moitra, S., Mukhopadhyay, A., 2008. A new multi-objective genetic algorithm applied to hot-rolling process. Applied Mathematical Modelling 32 (9), 1781-1789.

Chakraborti, N., Sreevathsan, R., Jayakanth, R., Bhattacharya, B., 2009. Tailor-made material design: an evolutionary approach using multi-objective genetic algorithms. Computational Materials Science 45(1), 1-7.

Chang, P., Hsieh, J., Lin, S., 2002. The development of gradual-priority weighting approach for the multi-objective flowshop scheduling problem. International Journal of Production Economics 79 (3), 171-183.

Deb, K., 2005. A population-based algorithm-generator for real-parameter optimisation. Soft Computation 9(4), 236-253.

Deb, K., Anand, A., Joshi, D., 2002. A computationally efficient evolutionary algorithm for real-parameter optimisation. Evolutionary Computation 10(4), 371-395.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182-197.

Deb, K., Sundar, J., Udaya, N., Chaudhuri. S., 2006. Reference point based multi-objective optimization using evolutionary algorithms. International Journal of Computational Intelligence Research 2(6), 273—286

Deb, K., Thiele, L., Laumanns, M., Zitzler, E., 2001. Scalable Test Problems for Evolutionary Multi-Objective Optimisation. Technical Report 112, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, Swiss.

Dulikravich, G.S., Egorov, I.N., Colaco, M.J., 2008. Optimizing chemistry of bulk metallic metallic glasses for improved the thermal stability. Modelling and Simulation in Materials Science and Engineering 16(7), 075010.

Egorov-Yegorov, I.N., Dulikravich, G.S., 2005. Chemical composition design of superalloys for maximum stress, temperature, and time-to-rupture using self-adapting response surface optimization. Materials and Manufacturing Processes 20(3), 569-590.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Reading, MA.

Hansen, N., 2007. The CMA evolution strategy: a tutorial. Available from: http://www.bionik.tu-berlin.de/user/niko/cmatutorial.pdf.

Hansen, N., Muller, S.D., Koumoutsakos, P., 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolutionary Computation 11(1), 1-18.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor, MI.

Jin, Y., Olhofer, M., Sendhoff, B., 2001. Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization: Why Does It Work and How? In: Proceedings of the Genetic and Evolutionary Computation Conference, 1042-1049.

Kennedy, J., Eberhart, R.C., 1995. Particle Swarm Optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1942-1948.

Knowles, J.D., Corne, D.W., 2000. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. Evolutionary Computation 8(2), 149-172.

Li, X., 2003. A real-coded predator-prey genetic algorithm for multiobjective optimization. Lecture Notes in Computer Science Volume 2632, 207-221.

Mahfouf, M., Gama, M.A., Panoutsos, G., 2009. "Right-First-Time" Production: A Reality or a Myth? Materials and Manufacturing Processes 24(1), 78-82.

Mahfouf, M., Jamei, M., Linkens, D.A., 2005. Optimal design of alloy steels using multiobjective genetic algorithms. Materials and Manufacturing Processes 20(3), 553-567.

Mahfouf, M., Linkens, D.A., Abbod, M.F., 2002. Optimisation of steel production incorporating economic factors. In: Proceedings of the 15th IFAC World Congress on Automatic Control.

Murata, T., Ishibuchi, H., Tanaka, H., 1996. Multi-objective genetic algorithm and its applications to flowshop scheduling. Computers & Industrial Engineering 30(4), 957-968.

Pettersson, F., Chakraborti, N., Saxen, H., 2007. A genetic algorithms based multi-objective neural net applied to noisy blast furnace data. Applied Soft Computing 7 (1), 387-397.

Sreevathsan, R., Bhattacharya, B., Chakraborti, N., 2009. Designing Ionic Materials Through Multiobjective Genetic Algorithms. Materials and Manufacturing Processes 24(2), 162-168.

Storn, R., 1996. On the usage of differential evolution for function optimisation. In: Proceedings of the 1996 Biennial Conference of the North American, Fuzzy Information Processing Society, pp. 519-523.

Storn, R., Price, K., 1995. Differential Evolution – a simple and efficient adaptive

scheme for global optimisation over continuous spaces. Technical Report TR-95-012, ICSI, Berkeley, CA.

Zhang, Q., 2009. Nature-Inspired Multi-Objective Optimisation and Transparent Knowledge Discovery via Hierarchical Fuzzy Modelling. PhD Thesis, The Department of Automatic Control and Systems Engineering, The University of Sheffield, UK.

Zhang, Q., Mahfouf, M., 2006. A New Structure for Particle Swarm Optimization (nPSO) Applicable to Single Objective and Multiobjective Problems. In: Proceedings of the 3rd International IEEE Conference on Intelligent Systems, pp. 176-181.

Zhang, Q., Mahfouf, M., 2007. A new Reduced Space Searching Algorithm (RSSA) and its application in optimal design of alloy steels. In: Proceedings of the IEEE Congress on Evolutionary Computation 2007, pp. 1815-1822.

Zhang, Q., Mahfouf, M., 2008. Mamdani-Type Fuzzy Modelling via Hierarchical Clustering and Multi-Objective Particle Swarm Optimisation (FM-HCPSO). International Journal of Computational Intelligence Research 4(4), 314-328.

Zitzler, E., Deb, K., Thiele, L., 2002. Comparison of Multiobjective Evolutionary Algorithm: Empirical Results. Evolutionary Computation 8(2), 173-195.

Zitzler, E., Thiele, L., 1998. An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach. Technical Report 43, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, Swiss.

Tables:

Table 1. Average performance of RSSA with different variation strategies in optimising f_5 to f_{11}: The integer in every cell is the average function evaluation number in successful runs; the value between the parentheses is the average result in the unsuccessful runs; the percentage value in the square brackets indicates the percentage of the successful runs out of all the runs; the bold values represent the best results.

Function	One-dimensional Variation	Multi-dimensional Variation	Multi-dimensional Variation
f_5	**N/A** **(0.0017)** **[0%]**	N/A (0.8025) [0%]	N/A (0.0043) [0%]
f_6	**76645** **(N/A)** **[100%]**	108630 (N/A) [100%]	N/A (-7712) [0%]
f_7	**88647** **(N/A)** **[100%]**	318860 (N/A) [100%]	N/A (1.7491e+2) [0%]
f_8	**N/A** **(1.2877e-5)** **[0%]**	N/A (1.7127e-5) [0%]	N/A (1.3796e+1) [0%]
f_9	2788 (0.0193) [40%]	2935 (0.0158) [50%]	**2753** **(0.0108)** **[55%]**
f_{10}	**46969** **(N/A)** **[100%]**	312660 (N/A) [100%]	N/A (1.7105e+1) [0%]
f_{11}	**N/A** **(1.7736e+2)** **[0%]**	N/A (1.8662e+2) [0%]	N/A (2.0294e+2) [0%]

Table 2. Average performance of various algorithms in optimising f_1 to f_{14}: The integer in every cell is the average function evaluation number in successful runs; the value between parentheses is the average result in the unsuccessful runs; the percentage value in the square brackets indicates the percentage of the successful runs out of all the runs; the bold values represent the best results.

Function	RSSA	CMA-ES	DE	G3+PCX
f_1	**1806** **(N/A)** **[100%]**	3207 (N/A) [100%]	391770 (N/A) [100%]	7140 (N/A) [100%]
f_2	N/A (0.0038) [0%]	**11751** **(N/A)** **[100%]**	655110 (N/A) [100%]	N/A (12.0469) [0%]
f_3	24287 (N/A) [100%]	**10830** **(N/A)** **[100%]**	N/A (1.8527) [0%]	25937 (N/A) [100%]
f_4	N/A (0.0147) [0%]	**8929** **(N/A)** **[100%]**	N/A (0.2004) [0%]	117414 (N/A) [100%]
f_5	N/A (0.0074) [0%]	**46072** **(N/A)** **[100%]**	N/A (0.0158) [0%]	140430 (N/A) [100%]
f_6	**73451** **(N/A)** **[100%]**	N/A (-6665) [0%]	616080 (N/A) [100%]	N/A (-6878) [0%]
f_7	**94499** **(N/A)** **[100%]**	N/A (106.1617) [0%]	940560 (N/A) [100%]	N/A (142.8754) [0%]
f_8	**209440** **(N/A)** **[100%]**	8575 (19.3625) [40%]	694560 (N/A) [100%]	N/A; (3.1199) [0%]
f_9	**2717** **(0.0112)** **[50%]**	5586 (0.0100) [75%]	586740 (N/A) [100%]	10983 (0.0110) [65%]
f_{10}	**52774** **(N/A)** **[100%]**	N/A (2.2897) [0%]	510180 (N/A) [100%]	N/A (15.1530) [0%]
f_{11}	N/A (197.8232) [0%]	N/A (248.84) [0%]	**N/A** **(0.1217)** **[0%]**	N/A (184.4355) [0%]
f_{12}	**329** **(N/A)** **[100%]**	221 (-19.8160) [95%]	853 (N/A) [100%]	N/A (-0.4128) [0%]
f_{13}	322 (N/A) [100%]	**224** **(N/A)** **[100%]**	1182 (N/A) [100%]	N/A (0.8862) [0%]
f_{14}	**366** **(N/A)** **[100%]**	253 (141.0000) [95%]	777 (N/A) [100%]	N/A (35.3369) [0%]

Table 3. Optimisation solutions of 10 independent runs for the UTS design problem with UTS_{Target} = 900 (MPa).

Solutions	1	2	3	4	5	6	7	8	9	10
Test Depth (mm)	61.8	67.8	111.4	41.9	129.6	58.1	78.6	18.8	93.9	74.2
Size (mm)	268.9	88.1	283.2	41.5	271.3	136.6	206.7	137.6	279.6	254.9
Site Number	2	5	5	2	5	3	6	3	3	4
C (wt%)	0.364	0.440	0.503	0.182	0.354	0.203	0.496	0.220	0.413	0.354
Si (wt%)	0.112	0.235	0.216	0.270	0.285	0.174	0.272	0.289	0.204	0.319
Mn (wt%)	1.554	1.189	0.939	0.954	1.397	0.644	0.521	0.488	0.742	0.940
S (wt%)	0.100	0.096	0.127	0.169	0.080	0.112	0.066	0.036	0.148	0.132
Cr (wt%)	0.263	0.589	3.025	0.613	2.733	0.615	0.790	0.650	0.140	0.489
Mo (wt%)	0.079	0.735	0.780	0.157	0.111	0.659	0.094	0.335	0.231	0.327
Ni (wt%)	0.609	2.069	0.241	0.379	3.765	0.312	2.967	2.557	2.003	1.023
Al (wt%)	0.641	0.028	0.029	0.842	0.190	0.093	0.086	0.253	0.260	0.495
V (wt%)	0.163	0.149	0.095	0.181	0.047	0.225	0.043	0.030	0.203	0.077
Hardening Temperature (°C)	970.3	971.9	908.6	979.4	860.2	907.6	889.1	975.2	933.1	923.8
Cooling Medium Number	2	3	2	3	2	1	2	1	1	2
Tempering Temperature (°C)	497.5	644.8	590.6	475.8	596.1	660.8	625.3	704.0	629.7	651.9
UTS (MPa)	900.1	899.9	899.9	899.9	900.0	899.6	899.9	899.8	899.9	900.0

Table 4. Optimisation solutions of 10 independent runs for the ROA design problem with ROA_{Target} = 60 (%).

Solutions	1	2	3	4	5	6	7	8	9	10
Test Depth (mm)	62.4	102.1	21.0	137.8	59.7	20.0	110.7	59.2	71.8	68.8
Size (mm)	74.4	351.6	251.9	71.5	101.0	310.7	250.6	94.6	303.6	276.9
Site Number	1	3	4	4	4	4	3	4	1	2
C (wt%)	0.434	0.249	0.246	0.248	0.517	0.191	0.243	0.204	0.240	0.239
Si (wt%)	0.297	0.295	0.129	0.226	0.222	0.193	0.154	0.227	0.157	0.281
Mn (wt%)	1.321	1.339	1.164	0.805	0.823	0.809	1.156	0.391	1.191	1.141
S (wt%)	0.033	0.041	0.128	0.208	0.114	0.158	0.181	0.189	0.012	0.095
Cr (wt%)	1.874	1.952	1.794	2.293	1.645	2.830	2.462	2.315	1.549	1.468
Mo (wt%)	0.207	0.747	0.384	0.151	0.024	0.152	0.335	0.667	0.955	0.797
Ni (wt%)	0.317	3.024	3.116	2.525	1.138	1.699	0.959	2.926	0.131	2.323
Al (wt%)	0.262	0.339	0.491	0.983	0.419	0.706	0.479	0.018	0.121	0.164
V (wt%)	0.180	0.079	0.139	0.233	0.096	0.045	0.172	0.187	0.237	0.177
Hardening Temperature (°C)	924.4	958.2	901.6	885.2	823.9	893.6	880.0	936.9	962.2	962.3
Cooling Medium Number	1	1	2	2	1	2	1	2	1	1
Tempering Temperature (°C)	534.2	300.4	413.1	513.2	640.4	316.6	383.1	595.2	289.0	680.1
ROA (%)	60.07	60.06	59.96	59.92	60.019	59.93	59.97	59.96	60.01	59.98

Table 5. Optimisation solutions of 10 independent runs for the design problem with UTS_{Target} = 900 (MPa) and ROA_{Target} = 60 (%).

Solutions	1	2	3	4	5	6	7	8	9	10
C (wt%)	0.427	0.530	0.516	0.453	0.438	0.503	0.404	0.436	0.426	0.502
Mn (wt%)	1.642	1.484	1.160	1.281	1.511	0.362	1.189	1.561	0.692	1.047
Cr (wt%)	1.341	0.136	0.436	0.583	1.186	1.116	0.316	1.196	0.431	0.639
Mo (wt%)	0.886	0.230	0.208	0.252	0.884	0.233	0.633	0.879	0.848	0.183
Tempering Temperature (°C)	868.7	940.3	917.3	906.0	897.5	925.4	945.0	890.8	888.5	936.3
UTS (MPa)	900.0	900.6	899.5	900.3	900.5	899.9	899.1	900.6	900.2	900.3
ROA (%)	59.94	60.03	59.99	59.94	59.99	60.00	59.98	59.99	60.04	60.02

Table 6. Pareto-optimal solutions for the design problem with UTS_{Target} = 600 (MPa) and ROA_{Target} = 50 (%).

Solutions	1	2	3	4	5	6	7	8	9	10
C (wt%)	0.528	0.211	0.474	0.408	0.235	0.209	0.225	0.224	0.226	0.136
Mn (wt%)	0.523	1.485	1.668	1.535	1.157	1.535	0.731	1.128	0.474	0.634
Cr (wt%)	1.706	3.130	1.874	2.158	1.218	2.158	0.253	0.278	0.257	1.933
Mo (wt%)	0.751	0.679	0.987	0.988	0.897	0.988	0.662	0.063	0.056	0.890
Tempering Temperature (°C)	969.3	903.5	978.9	941.5	880.2	839.8	847.0	847.1	847.0	914.4
UTS (MPa)	985.9	853.9	817.4	812.9	665.6	619.0	610.8	600.0	599.9	591.8
ROA (%)	50.43	49.41	52.02	52.02	53.29	60.70	67.52	69.80	71.35	69.22

Table 7. Contribution of composites to the cost of heat treatment.

Composite	Cost (US$ per tonne)
Manganese	18
Chromium	42
Molybdenum	52

Table 8. Contribution of tempering (annealing) to the cost of heat treatment.

Item	Cost (US$: 1.3GJ/tonne at 600 ^{o} C)
Annealing (tempering)	4.88

Table 9. Ten of the Pareto-optimal solutions for the design problem of UTS_{Target} = 900 (MPa) and minimising the heat treatment cost.

Solutions	1	2	3	4	5	6	7	8	9	10
C (wt%)	0.619	0.618	0.619	0.618	0.619	0.619	0.619	0.619	0.619	0.619
Mn (wt%)	1.661	0.738	1.101	1.031	0.921	0.853	0.846	0.799	0.734	0.694
Cr (wt%)	0.050	0.050	0.051	0.061	0.050	0.050	0.060	0.050	0.060	0.051
Mo (wt%)	0.010	0.205	0.050	0.047	0.053	0.051	0.017	0.010	0.010	0.010
Tempering Temperature (°C)	821.9	822.4	821.6	823.6	823.1	821.7	821.7	821.6	821.3	821.9
UTS (MPa)	900.0	891.5	877.4	870.8	860.6	850.7	838.0	827.8	819.3	811.3
Cost (US$)	39.22	32.75	31.30	30.33	28.20	26.81	25.38	23.73	22.96	21.88

Table 10. Ten of the Pareto-optimal solutions for the design problem of ROA_{Target} = 60 (%) and minimising the heat treatment cost.

Solutions	1	2	3	4	5	6	7	8	9	10
C (wt%)	0.436	0.611	0.467	0.599	0.607	0.607	0.614	0.562	0.562	0.562
Mn (wt%)	0.839	0.820	0.995	0.454	0.597	0.579	0.448	0.351	0.351	0.351
Cr (wt%)	0.242	0.149	0.089	0.050	0.113	0.076	0.050	0.050	0.050	0.050
Mo (wt%)	0.126	0.194	0.058	0.169	0.010	0.025	0.010	0.010	0.010	0.010
Tempering Temperature (°C)	960.2	870.7	888.6	882.3	868.6	867.0	830.3	862.7	820.8	820.1
ROA (%)	60.04	60.57	62.07	62.67	63.10	63.28	63.51	63.75	63.95	64.87
Cost (US$)	39.65	38.23	31.94	26.27	23.11	22.03	17.44	15.96	15.62	15.61

Table 11. Ten of the Pareto-optimal solutions for the design problem of UTS_{Target} = 900 (MPa), ROA_{Target} = 60 (%) and minimising the heat treatment cost.

Solutions	1	2	3	4	5	6	7	8	9	10
C (wt%)	0.612	0.602	0.604	0.598	0.441	0.613	0.606	0.536	0.531	0.619
Mn (wt%)	0.608	0.740	1.332	0.796	0.701	0.903	0.458	0.811	0.795	0.998
Cr (wt%)	0.357	0.295	0.050	0.050	0.878	0.050	0.366	0.244	0.208	0.050
Mo (wt%)	0.233	0.195	0.118	0.253	0.325	0.143	0.199	0.287	0.276	0.012
Tempering Temperature (°C)	892.1	895.2	849.3	840.9	898.8	852.2	862.4	849.6	856.5	831.9
UTS (MPa)	921.4	906.8	900.6	894.5	891.9	882.5	877.4	873.9	853.3	851.7
ROA (%)	60.24	61.23	64.00	64.32	59.87	63.69	62.29	60.07	59.80	64.44
Cost (US$)	45.34	43.21	39.14	36.45	73.75	32.73	41.05	46.71	44.44	27.51

Figures:

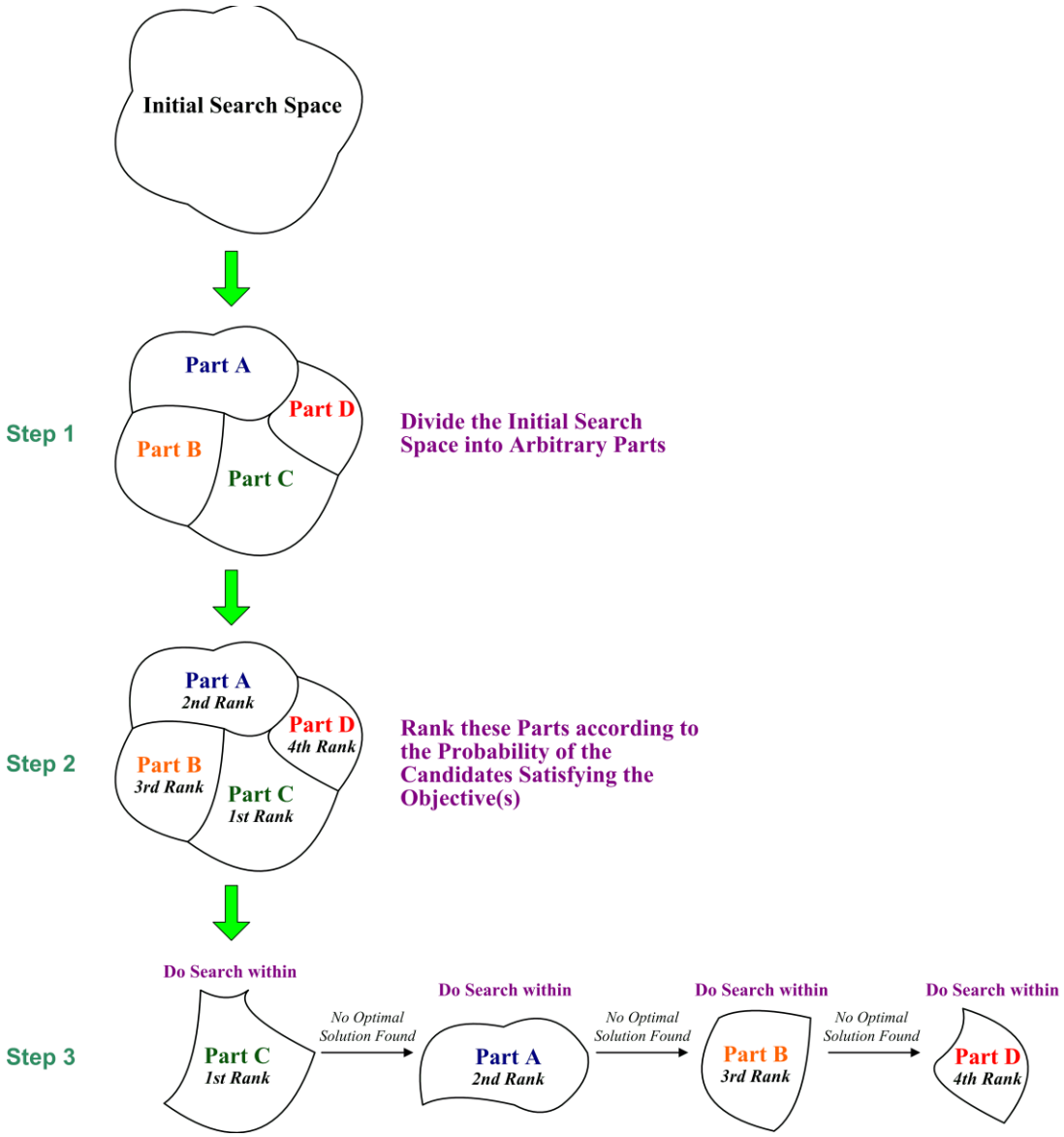


Figure 1. The RSS strategy for dealing with optimisation problems.

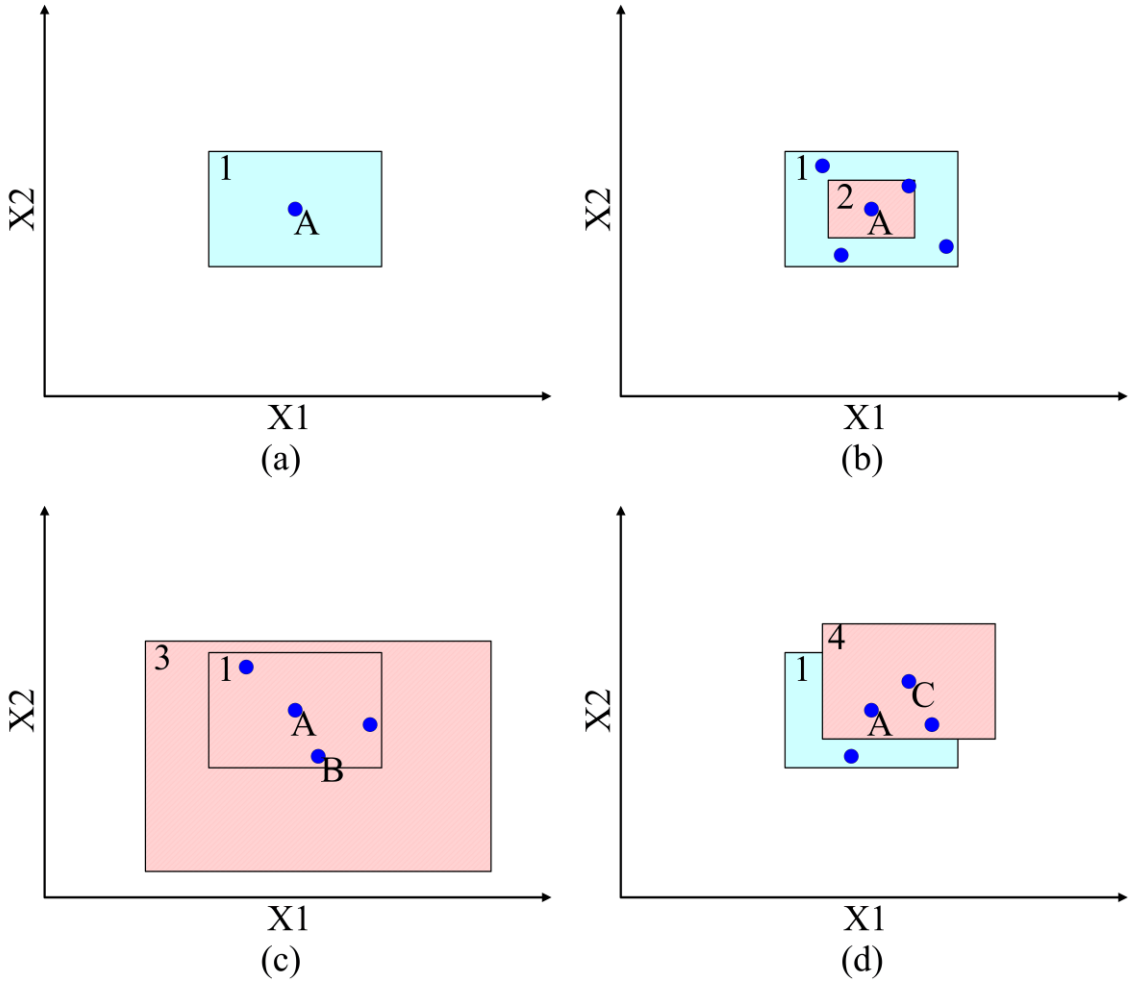


Figure 2. An example of how to divide the search space in the case a two-dimensional problem.

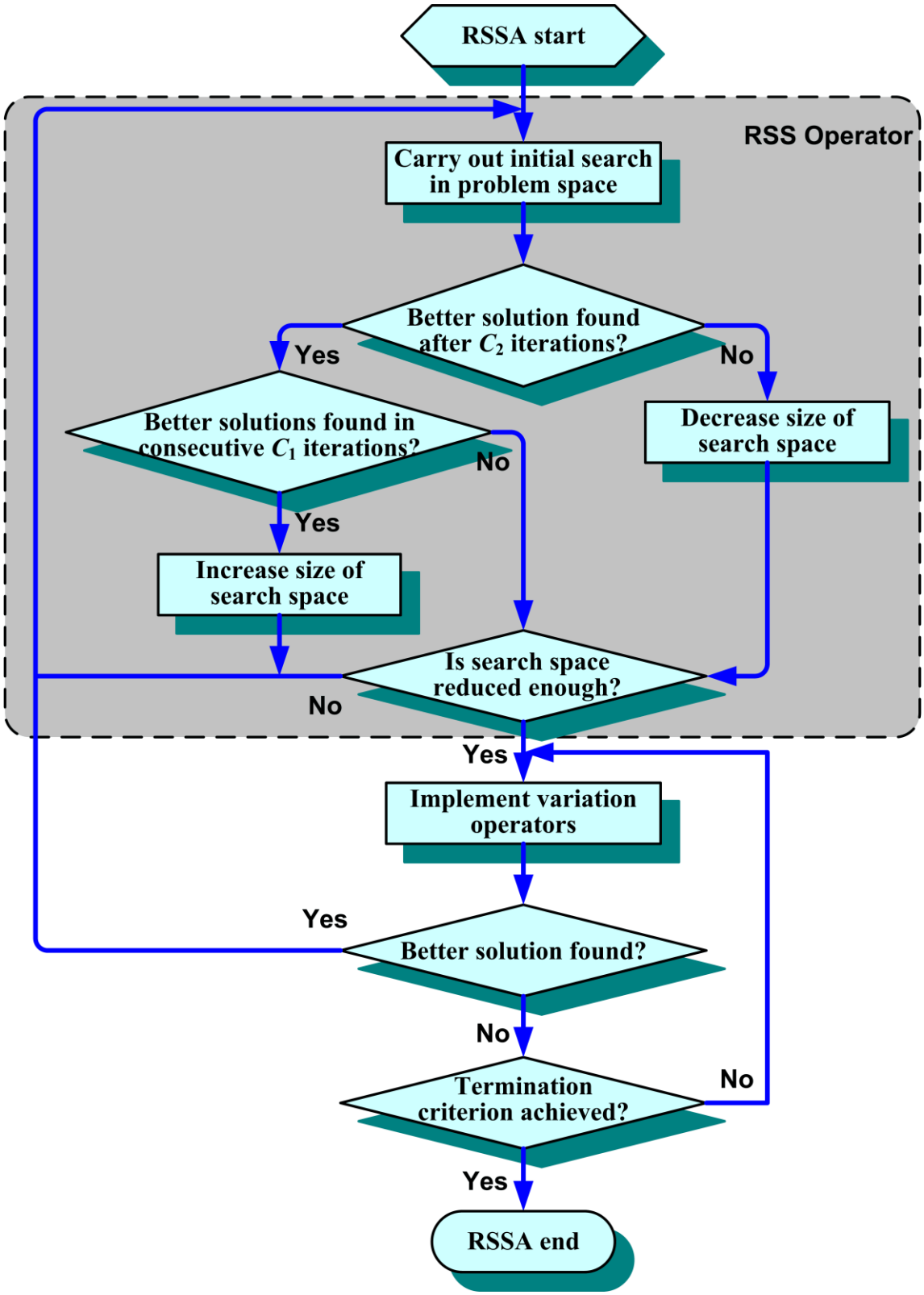


Figure 3. Flow chart of the RSSA algorithm.

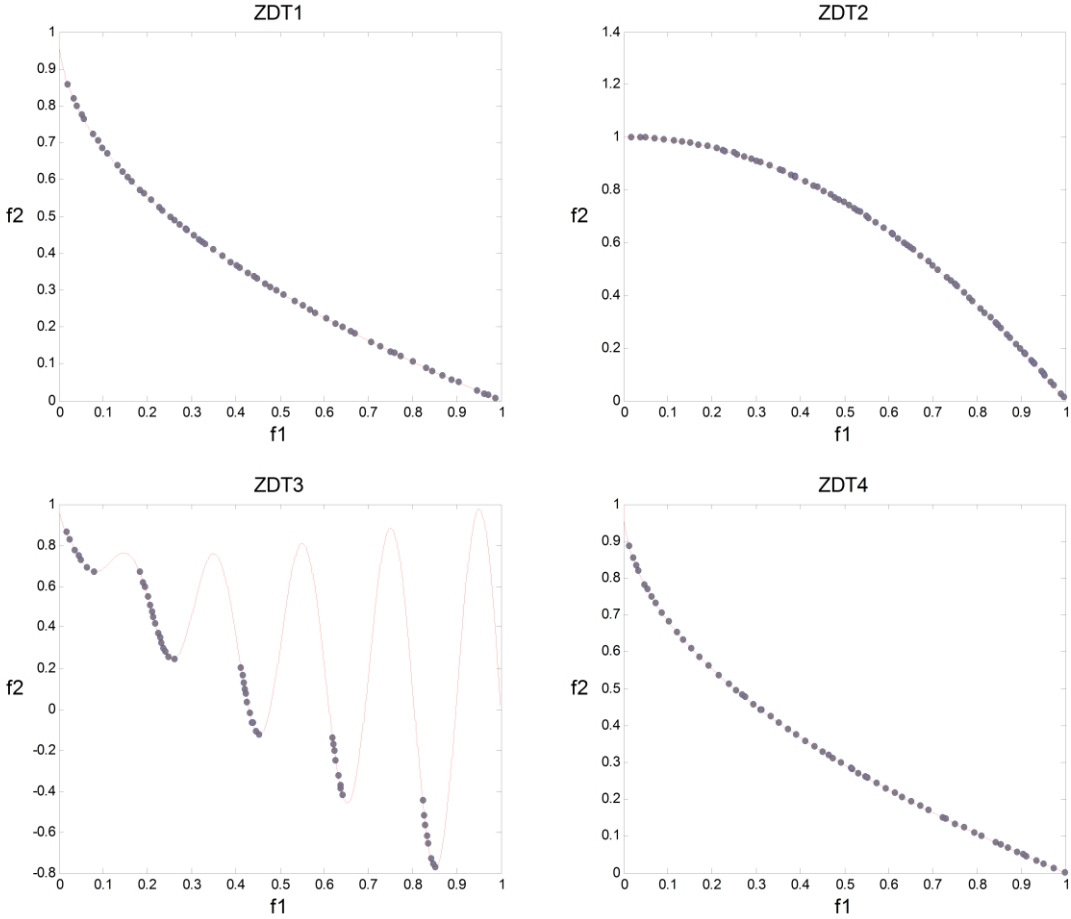


Figure 4. Pareto fronts obtained by MO-RSSA based on ZDT series problems.

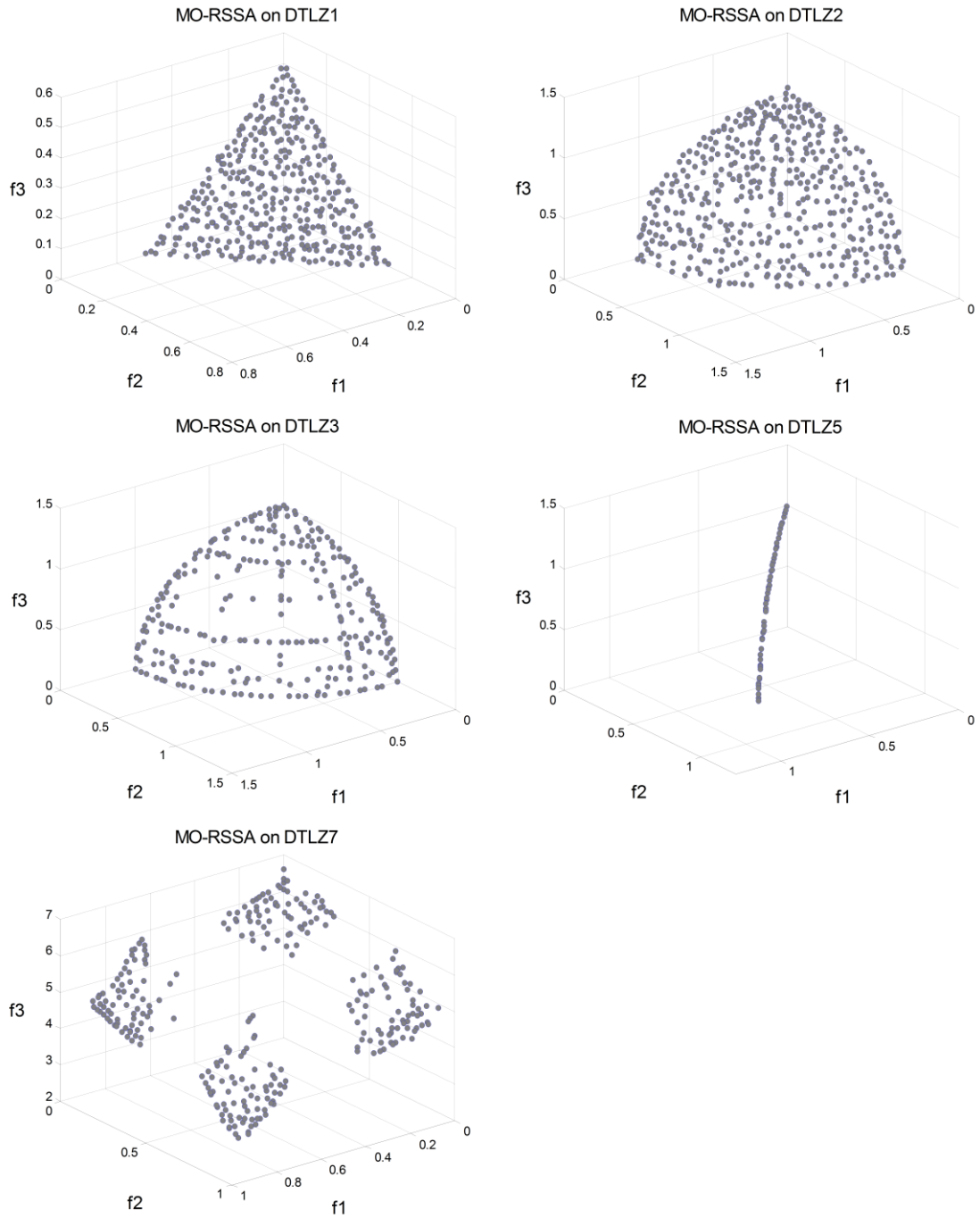


Figure 5. Pareto fronts obtained by MO-RSSA based on DTLZ series problems.

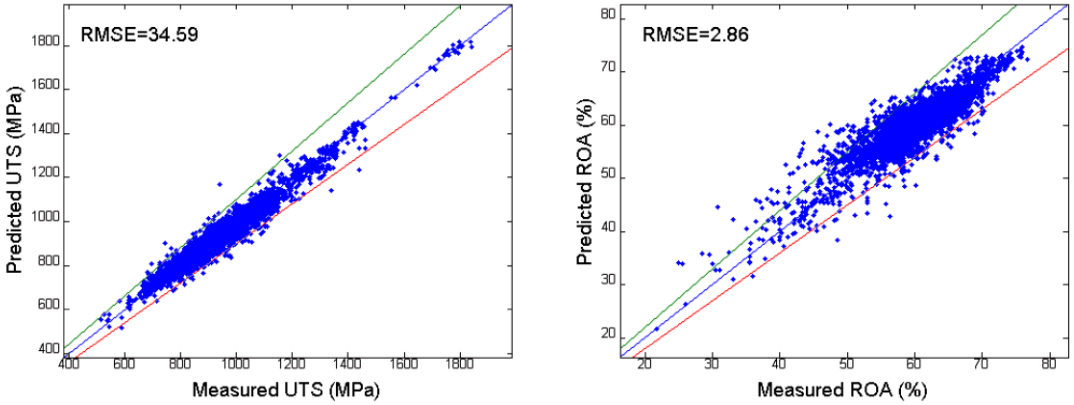


Figure 6. The prediction performance of the UTS model and the ROA models used in this chapter; the red and green lines delimit the +10% and -10% error bands respectively.

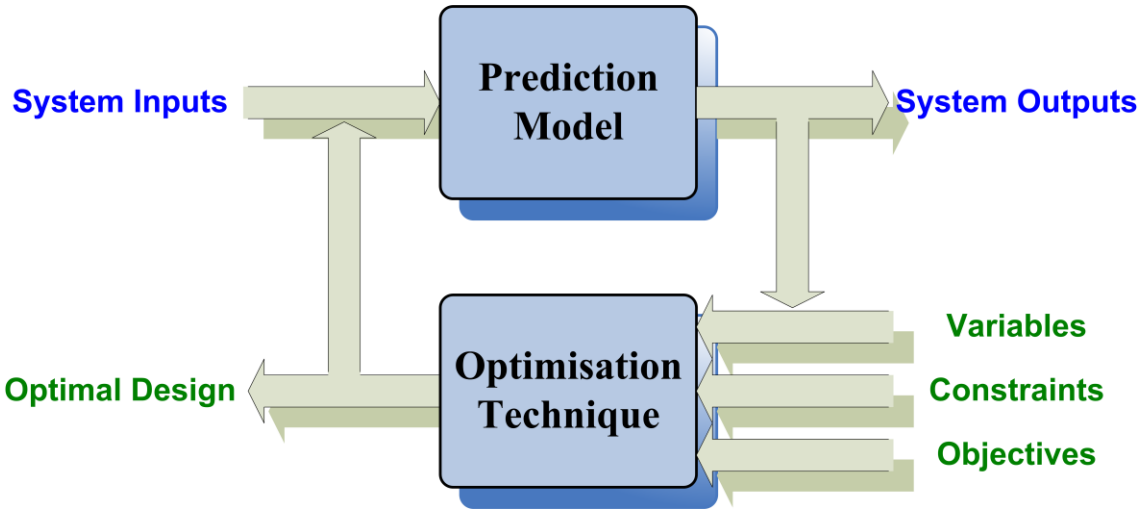


Figure 7. Optimal system design via reverse-engineering.

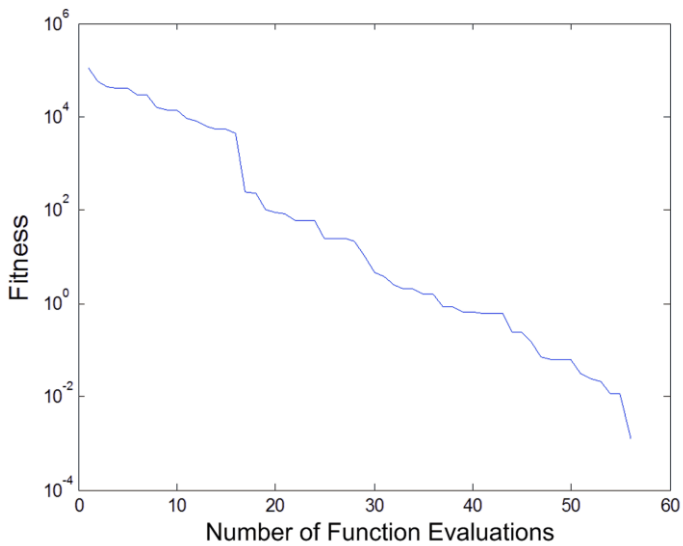


Figure 8. Average fitness of 10 runs versus function evaluation for the UTS design problem with UTS_{Target} = 900 (MPa).

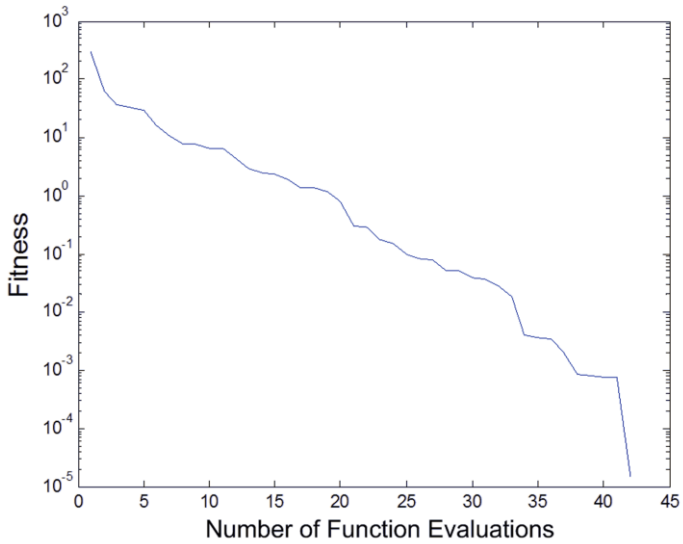


Figure 9. Average fitness of 10 runs versus function evaluation for the ROA design problem with ROA_{Target} = 60 (%).

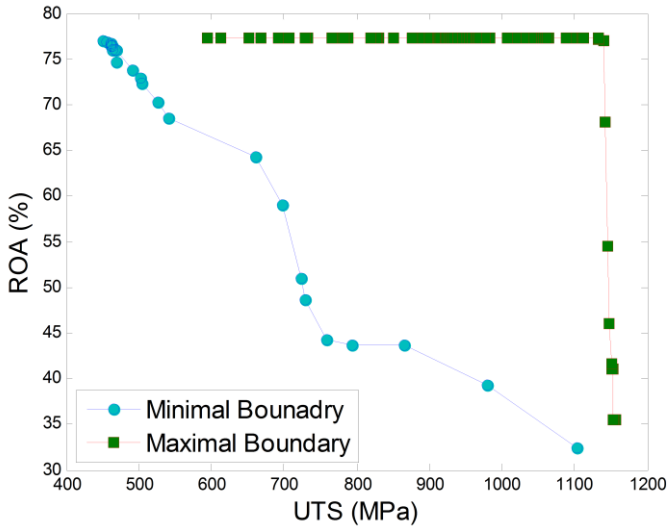
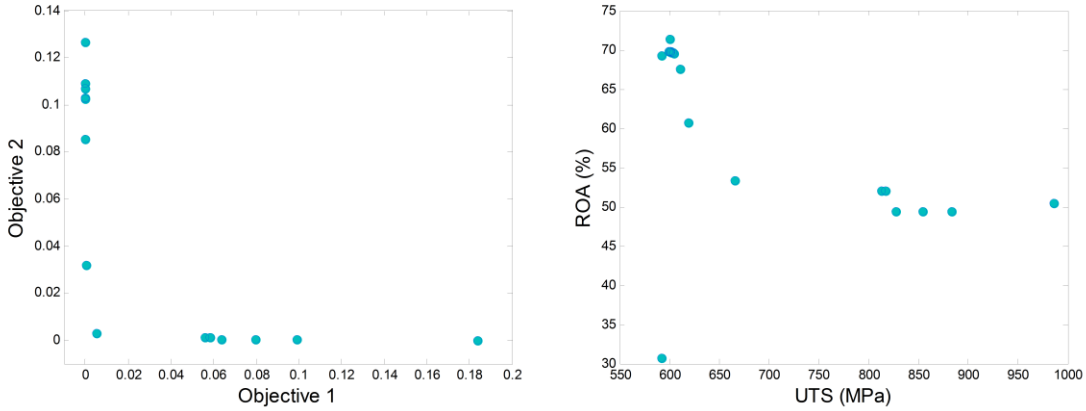
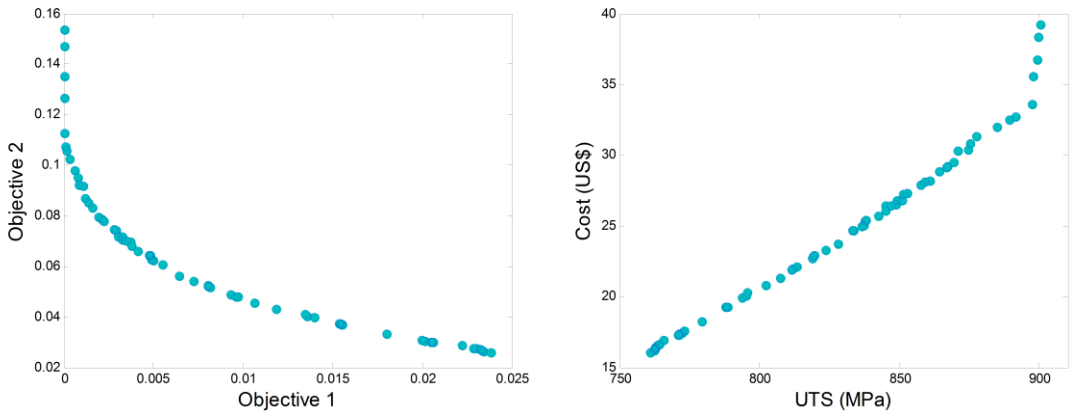


Figure 10. The maximum and minimum design boundaries for the problem of designing UTS and ROA simultaneously.



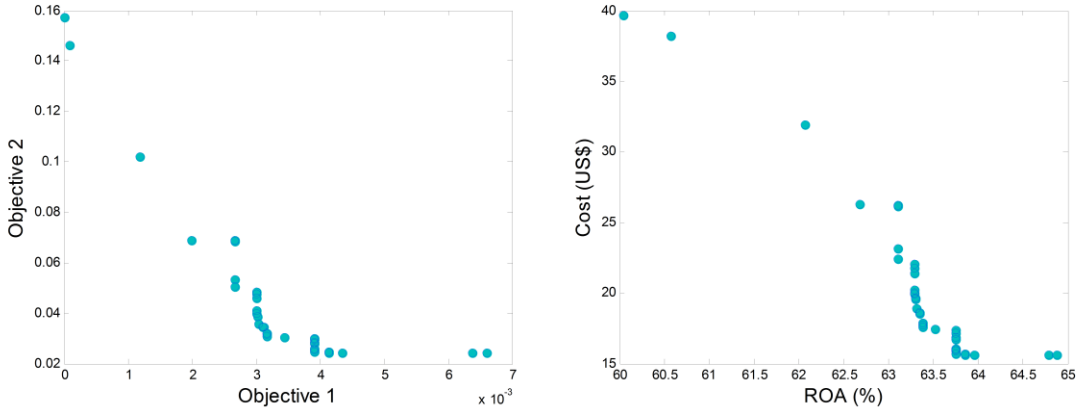
(a) (b)

Figure 11. The performance of the Pareto-optimal solutions for the design problem of UTS_{Target} = 600 (MPa) and ROA_{Target} = 50 (%) with respect to (a) the Objective 1 and the Objective 2 and (b) the UTS and the ROA.



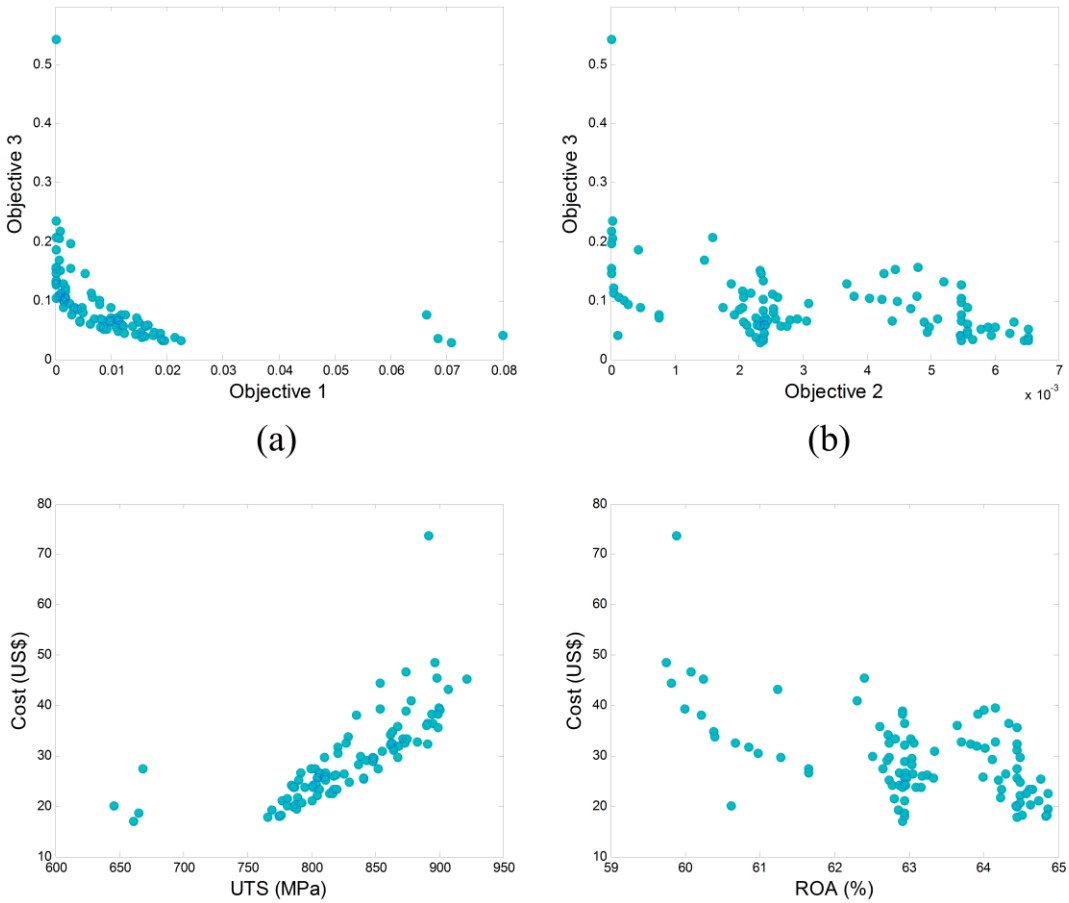
(a) (b)

Figure 12. The performance of the Pareto-optimal solutions for the design problem of UTS_{Target} = 900 (MPa) and minimising the heat treatment cost with respect to (a) Objective 1 and Objective 2; (b) UTS and Cost.



(a) (b)

Figure 13. The performance of the Pareto-optimal solutions for the design problem of ROA_{Target} = 60 (%) and minimising the heat treatment cost with respect to (a) Objective 1 and Objective 2; (b) ROA and Cost.



(a) (b)

(c) (d)

Figure 14. The performance of the Pareto-optimal solutions for the design problem of UTS_{Target} = 900 (MPa), ROA_{Target} = 60 (%) and minimising the heat treatment cost with respect to (a) Objective 1 and Objective 3; (b) Objective 2 and Objective 3; (c) UTS and Cost; and (d) ROA and Cost.