2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Calculating the Complete Pareto Front for a Special Class of Continuous Multi-Objective Optimization Problems

Xiao-Bing Hu, Ming Wang
The State Key Laboratory of Earth Surface Processes and Resource Ecology
Beijing Normal University
Beijing, China

Xiao-Bing Hu, Mark S Leeson
School of Engineering
University of Warwick
Coventry, UK

***Abstract*—Existing methods for multi-objective optimization usually provide only an approximation of a Pareto front, and there is little theoretical guarantee of finding the real Pareto front. This paper is concerned with the possibility of fully determining the true Pareto front for those continuous multi-objective optimization problems for which there are a finite number of local optima in terms of each single objective function and there is an effective method to find all such local optima. To this end, some generalized theoretical conditions are firstly given to guarantee a complete cover of the actual Pareto front for both discrete and continuous problems. Then based on such conditions, an effective search procedure inspired by the rising sea level phenomenon is proposed particularly for continuous problems of the concerned class. Even for general continuous problems to which not all local optima are available, the new method may still work well to approximate the true Pareto front. The good practicability of the proposed method is especially underpinned by multi-optima evolutionary algorithms. The advantages of the proposed method in terms of both solution quality and computational efficiency are illustrated by the simulation results.**

Keywords—*Index Terms*—Continuous Problem, Local Optima, Pareto Front, Multi-Objective Optimization, Evolutionary algorithm.

I. INTRODUCTION

Multi-objective optimization has a very wide set of realistic applications including, *inter alia*, product and process design, finance, aircraft design, the oil and gas industry, and automobile design [1], [2]. In multi-objective optimization problems, decisions have to be made in order to achieve the best trade-offs between two or more conflicting objectives. As is well known, it is in general impossible to identify a single solution that simultaneously optimizes each objective for nontrivial multi-objective problems. In such problems, improving one objective often makes other objectives suffer as a result. Therefore, Pareto-optimality becomes a crucial concept for solving multi-objective problems. A tentative solution is called non-dominated, Pareto optimal, or Pareto efficient if it cannot be eliminated from consideration by replacing it with another solution which improves one objective without worsening another one [3]-[5]. Therefore the goal of multi-objective optimization is to find such non-dominated solutions and quantify the trade-offs in satisfying the different objectives. All the Pareto-optimal solutions comprise the Pareto-optimal set, and the projection of this set in the objective space is called the Pareto front.

Basically, there are three main categories of methods to find the Pareto front for a multi-objective problem. The first category is referred to as aggregate objective function based methods, because their core idea is to integrate all original objectives into a single aggregate objective function (e.g., the weighted linear sum of the objectives) [4], [5] and thus a set of aggregate objective functions needs to be constructed. Then in terms of each aggregate objective function in the set, a single-objective optimization problem is resolved to produce a Pareto-optimal solution to the original multi-objective problem. Aggregate objective function based methods are often criticized for their subjectivity in constructing aggregate objective functions, as well as their incapability of calculating a non-convex Pareto front [6]-[10]. The second category is called constrained objective function based methods, often known as ε-constraint methods, which only attempt to minimize one single objective while treating all other objectives as extra constraints [1], [11]. If the constraints on objective functions are properly set, this category of methods may deliver good performance. The third category, probably the most widely used category, is called Pareto-compliant ranking based methods. This category of methods needs to be able to generate and operate on a pool of candidate solutions, in order to take advantage of a Pareto-compliant ranking procedure. Population-based evolutionary approaches, such as genetic algorithm (GA), particle swarm optimization and ant colony optimization, are important family members of this category [12]-[24]. A problem is, however, that due to the stochastic nature of multi-objective evolutionary approaches, the outputs of such techniques are in theory approximations of

This work was supported in part by the National Basic Research Program of China under grant 2012CB955404, the project grant 256-105570GK from the Fundamental Research Funds for the Central Universities, China, and the Seventh Framework Programme (FP7) of the European Union under Grant PIOF-GA-2011-299725.

the true Pareto front; in other words, it is likely that there is no Pareto-optimal solution at all contained in the final generation of the evolutionary process [12], [14]. This is different from the other two categories of methods, in which each output solution is usually a Pareto-optimal solution [10], as long as certain conditions are satisfied. For example, some constrained objective function based methods may theoretically guarantee to find the true Pareto front in specific circumstances [1].

In short, most existing methods for multi-objective optimization problems mainly aim to or are only able to provide an approximation of the true Pareto front, and there are very few theoretical results regarding how to guarantee the finding of the complete Pareto front [1], [6], [12], [25]. In contrast to most existing results, we have recently reported in [26] a deterministic method to calculate the complete Pareto front for a class of discrete problems for which it is possible to develop an effective method to find the *k* best solutions in terms of each single objective. Some theoretical conditions were given in [26] in order to guarantee the finding of the complete Pareto front, and then a practicable search procedure was proposed, whose effectiveness and efficiency were demonstrated by comparative experiments. The theoretical results were successfully applied to the discrete problem of multi-objective new product development [27]. This paper attempts to shed a little more light on the issue of whether it is possible, theoretically and practically, to calculate the complete Pareto front for multi-objective optimization problems. In particular, we will focus on how to extend the work in [26] from discrete problems to continuous problems. To this end, firstly we generalize the theoretical conditions in [26]. Then based on the new conditions, we propose a practicable search procedure inspired by the natural rising sea level phenomenon, in order to calculate the complete Pareto front for a special class of continuous multi-objective problems, where the number of local minima in terms of each single objective function is finite and there is an effective method to find all such local minima. The proposed method is then also extended to problems for which it is not possible to find all local minima. In this case, a good approximation of Pareto front is still achievable by the new method, and well-developed multi-optima evolutionary algorithms can provide a concrete support to the practicability of the proposed method.

II. Generalized Theoretical Conditions for Finding Pareto Front

Some theoretical conditions for finding the complete Pareto front were recently reported in [26] for either discrete problems or continuous problems (but not for both). In this section, we will generalize those conditions to guarantee a complete cover of the true Pareto front, regardless of the continuous nature or discrete nature of multi-objective optimization problems.

A general mathematical formulation of multi-objective optimization problems can be described as following:

$$\min_X [g_1(X), g_2(X), ..., g_{N_{Obj}}(X)]^T, \tag{1}$$

$$\text{subject to} \quad h_I(X) \le 0, \tag{2}$$

$$h_E(X) = 0, \tag{3}$$

$$X \in \Omega_X, \tag{4}$$

where g_i is the ith objective function of the total N_{Obj} objective functions, h_I and h_E are the inequality and equality constraints, respectively, and X is the vector of optimization or decision variables belonging to the set of Ω_X. A Pareto-optimal solution X^* to the above problem is such that there exists no x that makes

$$g_i(X) \le g_i(X^*), \text{ for all } i=1,..,N_{\text{Obj}}, \tag{5}$$

$$g_j(X) < g_j(X^*), \text{ for at least one } j \in [1,..,N_{\text{Obj}}]. \tag{6}$$

The projection of such an X^* in the objective space, i.e., the point $[g_1(X^*), g_2(X^*), ..., g_{N_{Obj}}(X^*)]$, is called a Pareto point. For the above problem, there is usually a set of Pareto-optimal solutions, and the projection of this set in the objective space is called the Pareto front.

Based on the definition of the Pareto front, we have the following statements regarding a complete cover of the true Pareto front of multi-objective optimization problems, no matter whether they are continuous or discrete.

Theorem 1: Suppose there exist $X_1, ..., X_{N_{Obj}}$ such that for any $j \in [1, ..., N_{Obj}]$,

$$g_i(X_j) \le g_i(X_i), \text{ for all } i=1,..,N_{\text{Obj}}. \tag{7}$$

Then all Pareto-optimal solutions are included in the union set

$$\Omega_{U1} = \bigcup_{i=1}^{N_{Obj}} \{X : g_i(X) \le g_i(X_i)\}. \tag{8}$$

Proof: Assume Theorem 1 is false. Therefore, there exists at least one Pareto-optimal solution, say X^*, that does not belong to the union set Ω_{U1}, which means, according to the definition of Ω_{U1} in (8), we have $g_i(X_i) < g_i(X^*)$ for all $i=1,..,N_{\text{Obj}}$. Then for any $j \in [1, ..., N_{Obj}]$, we have

$$g_i(X_j) \le g_i(X_i) < g_i(X^*), \text{ for all } i=1,..,N_{\text{Obj}}. \tag{9}$$

This means $X_1, ..., X_{N_{Obj}}$ are all more Pareto efficient than X^*. In other words, X^* is not a Pareto-optimal solution at all. Therefore, the assumption must be false, and Theorem 1 must be true.

Theorem 2: For any given set of solutions $[X_1, ..., X_{N_{GS}}]$, all Pareto-optimal solutions are included in the intersection set

$$\Omega_{IS1} = \bigcap_{j=1}^{N_{GS}} \left(\bigcup_{i=1}^{N_{Obj}} \{X : g_i(X) \le g_i(X_j)\} \right). \tag{10}$$

Proof: For any $X^+ \in \Omega_X$, all Pareto-optimal solutions are included in the union set

$$\Omega_{U2} = \bigcup_{i=1}^{N_{Obj}} \{X : g_i(X) \le g_i(X^+)\}, \tag{11}$$

Because if we $X_i = X^+$ for all i=1,..,N_{Obj}, then Condition (7) in Theorem 1 will definitely hold. Thus, for each solution in $[X_1,...,X_{N_{GS}}]$, we can work out a union set Ω_{U2}, respectively. Then, the intersection set of all such union sets, i.e., Ω_{UIS1} as defined in Eq.(10), will always cover all Pareto-optimal solutions. Therefore, Theorem 2 must be true.

One may notice that Theorem 1 is more restrictive than Theorem 2, whilst the cover given by Theorem 1 seems worse than that of Theorem 2. However, as will be shown later, Theorem 1 may have a computational efficiency from the viewpoint of practicability.

III. APPLICATION TO A SPECIAL CLASS OF CONTINUOUS PROBLEMS

Please note in Section 2, variable X may be either discrete or continuous, whilst in this section we only consider continuous multi-objective optimization problems. Therefore, we need to modify the problem formulation of Section 2 to the follows:

$$\min_{X=[x_1,...,x_{N_X}]} [g_1(X), g_2(X),..., g_{N_{Obj}}(X)]^T, \quad (12)$$

subject to Conditions (2), (3) and

$$x_i \in \Omega_{X_i}, \ i=1,...,N_X, \quad (13)$$

where X=[$x_1,...,x_{N_X}$] are the $N_X \geq 1$ optimization or decision variables of continuous real number, and Ω_{Xi} is the set of possible values for variable x_i. The set Ω_{Xi} can be further defined as a union set of $N_{RXi} \geq 1$ ranges of real number

$$\Omega_{X_i} \equiv \bigcup_{j=1}^{N_{RXi}} \{[\underline{x}_{i,j}, \overline{x}_{i,j}]\}, \quad (14)$$

where $\underline{x}_{i,j} \geq -\infty$ and $\overline{x}_{i,j} \leq \infty$ are the lower bound and upper bound of the jth range for variable x_i, respectively, and

$$\underline{x}_{i,j} \leq \overline{x}_{i,j}, \ j=1,...,N_{RXi}, \quad (15)$$

$$\underline{x}_{i,j+1} > \overline{x}_{i,j}, \ j=1,...,N_{RXi}-1. \quad (16)$$

Since variable x_i is a continuous real number, Condition (15) implies that a range for x_i can include either infinite values (when $\underline{x}_{i,j} < \overline{x}_{i,j}$) or just only one value (when $\underline{x}_{i,j} = \overline{x}_{i,j}$). Condition (16) means there is no intersection between any two ranges for the same variable x_i. In this study, we assume that every objective g_i is a continuous, differentiable function of X within all ranges. We also assume that the number of local minima in terms of each single objective g_i is finite. Let $X^*_{LM}(i,j)$ denote the X that is associated with jth local minimum in terms of g_i, where j=1,…, $N_{LM}(i)$, and $1 \leq N_{LM}(i) < \infty$ is the number of local minima in terms of g_i. $X^*_{LM}(i,j)$ has the jth smallest g_i value in all the $N_{LM}(i)$ local minima of g_i. In this study, it is assumed that no local minimum exists when $x_i=\infty$ or $x_i=-\infty$ for any i=1,...,N_X.

Suppose there is a certain effective method to find all the local minima in terms of each single objective g_i in the continuous multi-objective optimization problem defined by (12) to (16). Then based on the theoretical results in Section 2, we can calculate the complete Pareto front by starting a search from local minima. Basically, for each objective g_i, we start from its first smallest local minimum $g_i(X^*_{LM}(i,1))$ (actually the global minimum in terms of g_i). We keep increasing g_i by a predefined constant Δg_i, and then checking all solutions whose g_i is no larger than the current g_i level, until certain conditions derived from the theoretical results in Section 2 are satisfied. This is likened to the rising sea level phenomenon as illustrated in Fig.1.(a), where every time the sea level rises by Δg_i, all lands which are below the new sea level will flood. The flooded lands are the areas where we need to check in order to guarantee the finding of the complete Pareto front. Sometimes a small increase in g_i may result in a substantial change in X. To better search the landscape of the solution space, we can expand the current coast lines by a predefined constant ΔX in all feasible directions, then we increase the sea level to the minimal g_i associated with the expanded coast lines, and at last we correct the expanded coast lines to make sure that any solution in the coast lines has the same g_i. Fig.1.(b) illustrates how the sea level rises as X changes by ΔX.

Now the question is: To which level does the sea need to rise so that a complete cover of the true Pareto front is guaranteed? With Theorem 1 in Section2, we have the following rising-sea-level inspired search procedure to answer this question.

Step 1. Calculate with a certain method all local minima $X^*_{LM}(i,j)$, i=1,…,N_{Obj}, j=2,…,$N_{LM}(i)$. Predefine ΔX. For each i=1,…,N_{Obj}, let $\Omega_{CL}(i)$ denote the coast lines associated with the objective g_i, initialize $\Omega_{CL}(i)$ ={ $X^*_{LM}(i,1)$ }, and mark $X^*_{LM}(i,1)$ as explored. Set the current sea level of g_i as $L_{CSL}(i) = g_i(X^*_{LM}(i,1))$. Let the current time t=0. Let the $X_{MSL}(i) = X^*_{LM}(i,1)$ denote the solution in the coast lines $\Omega_{CL}(i)$ that has the minimal sea level of g_i.

Step 2. If $X_{MSL}(1)$,…, $X_{MSL}(N_{Obj})$ make Condition (7) in Theorem 1 hold, i.e., for any j=1,..,N_{Obj}.

$$g_i(X_{MSL}(j)) \leq g_i(X_{MSL}(i)), \text{ for all } i=1,..,N_{Obj}, \quad (17)$$

then go to Step 3. Otherwise, t=t+1, and do the follows for each i=1,…,N_{Obj}:

Step 2.1. Move from $X_{MSL}(i)$ by ΔX in all unsearched feasible directions (or well discretized directions when N_X>1), and update the cost lines $\Omega_{CL}(i)$ by adding new solutions which are found by changing $X_{MSL}(i)$.

Step 2.2. Update $X_{MSL}(i)$ with the solution which has the minimal g_i in $\Omega_{CL}(i)$, and update the current sea level $L_{CSL}(i) = g_i(X_{MSL}(i))$.

Step 2.3. For an unexplored local minimum $X^*_{LM}(i,j)$, j=2,…,$N_{LM}(i)$, if $L_{CSL}(i) \geq g_i(X^*_{LM}(i,j))$, then add $X^*_{LM}(i,j)$ to the cost lines $\Omega_{CL}(i)$, and mark $X^*_{LM}(i,j)$ as explored..

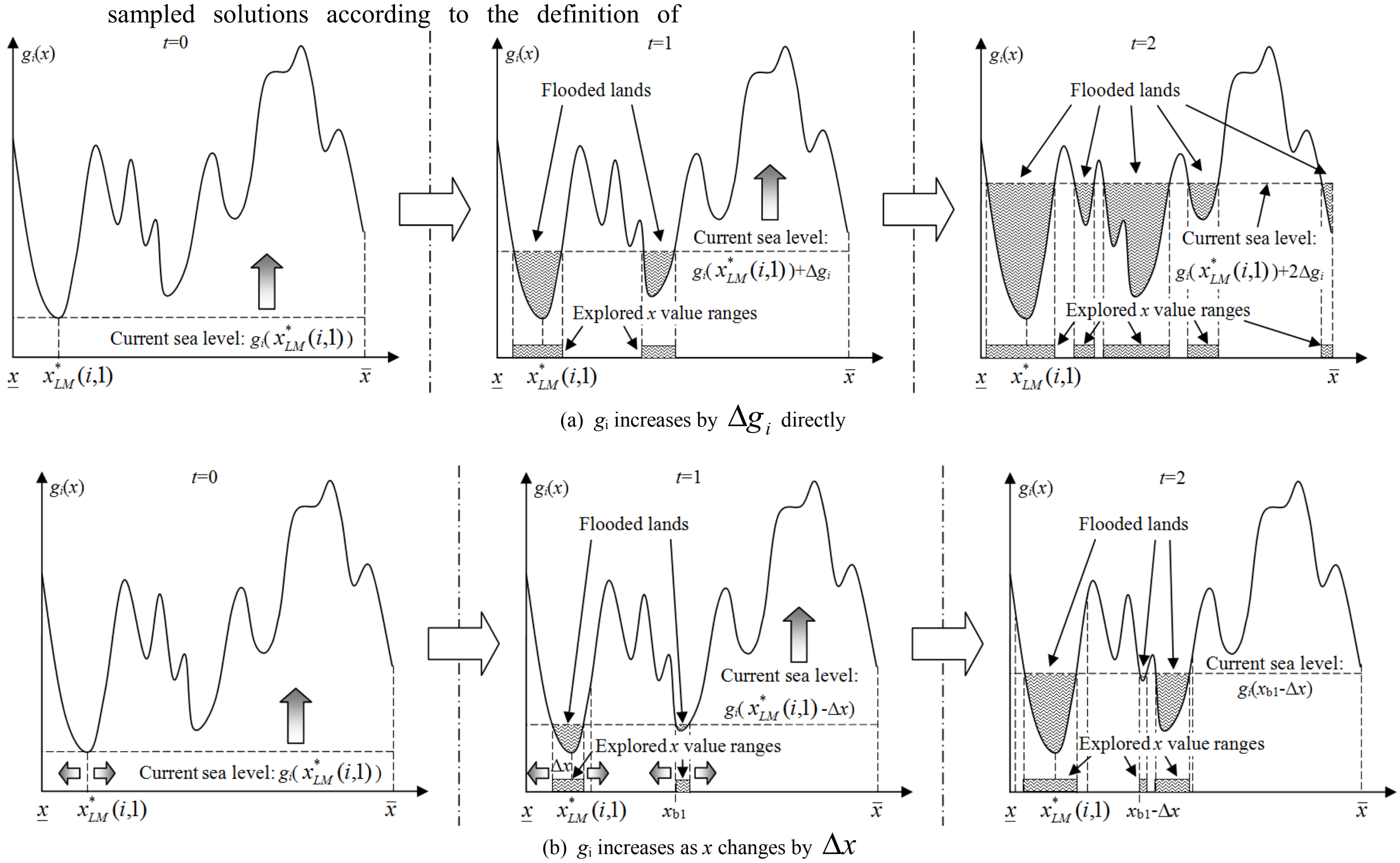
Step 3. Well sample those solutions in all flooded areas, and calculate the Pareto front by crosschecking the sampled solutions according to the definition of Pareto optimality given by (5) and (6).

One can see that in Step 2, the result of Theorem 1 is used as the termination criteria for stopping the search procedure. We can also integrate other theoretical results of Section 2 into the search procedure as criteria for terminating the simulation of sea level rising. For example, for a single given solution X_{GS}, to apply Theorem 2, we simply need to change Step 2 above as following:

Step 2. Go to Step 3 if

$$L_{CSL}(i) \geq g_i(X_{GS}), \text{ for all } i=1,..,N_{Obj}. \tag{18}$$

Otherwise, t=t+1, and for any i=1,…,N_{Obj} with $L_{CSL}(i) < g_i(X_{GS})$, do Step 2.1 to Step 2.3.



(a) g_i increases by Δg_i directly

(b) g_i increases as x changes by Δx

Fig.1 Exploration of the solution space by simulating rising sea leve

It should be noted that in the above search procedure, in each time instant, we do not expand all the coast lines simultaneously, and we do not require that all solutions in $\Omega_{CL}(i)$ have the same sea level, either. This is different from the natural sea level rising phenomenon, but from the viewpoint of practicability, they are all necessary designs. This is because it is usually much easier to calculate g_i based on X rather than to calculate X based on g_i. Since the $\Omega_{CL}(i)$ does not share the same sea level, then expanding the entire coast lines makes no sense, and instead, expanding the point which has the minimal sea level in $\Omega_{CL}(i)$ step by step is more practicable and will eventually give the same result as expanding the entire coast lines.

It is clear from the above that in the proposed search procedure, a brute-force search strategy is actually employed within all explored x ranges. Fortunately, the size of the explored x ranges is much smaller than the entire search space, as will be illustrated in the simulation section.

The guarantee that the proposed search procedure finds the true Pareto front requires an effective method to find all local minima to each of the N_{Obj} single-objective problems. Even if this is not possible, the proposed method may still work well to approximate the true Pareto front, just as most existing methods do. There are many well-developed methods to search for multiple optima, such as multi-optima evolutionary algorithms [28]-[30]. Although such methods might not guarantee to find all local optima, they can often find most of them even for very complicated problems. With these methods for multi-optima problems at hand, the proposed search procedure will have a good practicability.

IV. Simulation Results

In this section, we conduct numerical simulations to demonstrate the practicability and the effectiveness of the proposed methodology to determine the complete Pareto front for continuous problems, and also to verify the theoretical results given in the above sections.

The following continuous optimization problem model with two objectives is employed:

$$\min_{x} [g_1(x), g_2(x)]^T , \tag{19}$$

subject to

$$\begin{aligned} g_1(x) = {} & (\alpha_1 + 0.01)\sin((\alpha_2 + 0.01)x + 3.14\alpha_3) \\ & \times | \alpha_4 + 2.01 - \sin((\alpha_5 + 0.01)x + 3.14\alpha_6) | \\ & + (\alpha_7 + 0.01)\sin((\alpha_8 + 0.01)x + 3.14\alpha_9), \end{aligned} \tag{20}$$

$$\begin{aligned} g_2(x) = {} & (\alpha_{10} + 0.01)\sin((\alpha_{11} + 0.01)x + 3.14\alpha_{12}) \\ & \times | \alpha_{13} + 2.01 - \sin((\alpha_{14} + 0.01)x + 3.14\alpha_{15}) | \\ & + (\alpha_{16} + 0.01)\sin((\alpha_{17} + 0.01)x + 3.14\alpha_{18}), \end{aligned} \tag{21}$$

and

$$x \in [0,100], \tag{22}$$

where $0 \le \alpha_i \le 1$, i=1,...,18 are randomly generated coefficients. A problem framed in this model has a finite number of local minima in terms of each of the two objective functions. It is fairly easy to check the complete Pareto front once the coefficients $[\alpha_1,...,\alpha_{18}]$ are given, which makes it possible to verify the correctness of the theoretical conditions in Section 2 as well as to assess the effectiveness of the proposed method in Section 3.

In the simulations, we used the above model to construct 200 different bi-objective optimization problems by randomly generating 200 sets of coefficients $[\alpha_1,...,\alpha_{18}]$. Then we applied the proposed method as well as other relevant methods to search for the Pareto front of each bi-objective problem. Since the proposed method is inspired by the rising sea level phenomenon, it is hereafter denoted as RSL. There are two versions of RSL used in the experiment, one based on Theorem 1 (denoted as RSL-T1), and other based on Theorem 2 (RSL-T2). The expanding step was set to $\Delta x = 0.01$. In addition to RSL, we also used two well-known multi-objective optimization methods for the sake of comparison. One was the constrained objective function based method in [1] (denoted as COF hereafter). The reason COF was used was because it is one of a few existing method that may theoretically guarantee (naturally under certain conditions) finding the true Pareto front. Therefore, comparing RSL with COF is a must-do task in order to verify the theoretical results in Section 2. The other method used was the well-established NSGA-II in [16] as a representative of multi-objective evolutionary algorithms, and a comparison with NSGA-II helps to assess how well the proposed RSL may perform even if there is no effective method to find all local minima. Aggregate objective function based methods were not considered in this study, because, as illustrated in Fig.2, the Pareto front to a problem defined by (19)-(22) is often not globally convex, which makes it difficult to apply aggregate objective function based methods. In the experiment, all methods were coded and all tests were conducted in a Matlab environment on a personal computer with a 2.6GHz CPU, 4GB memory and the Windows 7 operating system.

A. Calculation of Complete Pareto Front

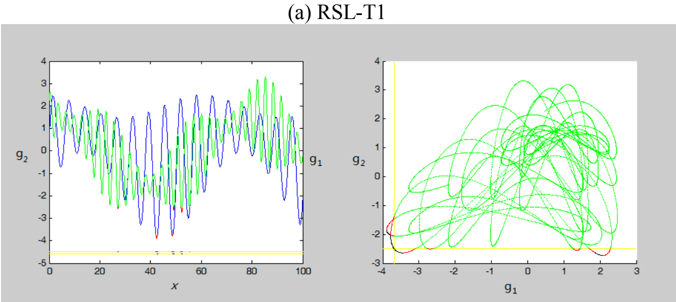
In this subsection, we aim to experimentally verify the correctness of the theoretical conditions in Section 2, and also to test whether the proposed RSL has the capability to find the complete Pareto front, and how good the capability is. To this end, two versions of RSL are compared COF. As is well known, in COF, a multi-objective problem is treated as a single-objective problem by imposing constraints on all other objectives. For example, when applying COF in this experiment, we restricted the value of g_2 (or g_1), and then actually only minimized g_1 (or g_2); In other words, g_1 (or g_2) was minimized with g_2 (or g_1) within a constrained range $[\underline{g}_2, \overline{g}_2]$ (or $[\underline{g}_1, \overline{g}_1]$). Then we moved the constrained range by a small increment, and minimized g_1 (or g_2) again. This was repeated until some termination criteria were satisfied. There are two conditions for COF to find the complete Pareto front: (i) The change in the constrained range must be small enough each time; (ii) Every constrained objective function must have moved through a sufficient value range. In this study, the change step for moving the constrained range is set as 0.01, and the total range for a constrained objective are determined by $[g_1(x^*_{LM}(1,1)), g_1(x^*_{LM}(2,1))]$ or $[g_2(x^*_{LM}(2,1)), g_2(x^*_{LM}(1,1))]$.

Fig.2 gives the results of a test case. The subplots on the left-hand side show the landscape of the solution space, as well as the explored x value ranges by different methods (red parts) and the Pareto front associated x value range (black parts). The subplots on the right-hand side give the projections of all solutions in the objective space, the areas explored by different methods (red parts plus black parts separated from green parts by a horizontal yellow line and a vertical yellow line), and the true Pareto front (black parts). Fig.2.(a) gives the results of RSL-T1, Fig.2.(b) shows those of RSL-T2, where the given solution X_{GS} for the termination criteria (18) is randomly chosen, and Fig.2.(c) is associated with COF with g_2 as the constrained objective function within the range $[g_2(x^*_{LM}(2,1)), g_2(x^*_{LM}(1,1))]$. From Fig.2, one can see intuitively that (i) both RSL and COF can find the complete Pareto front, and (ii) RSL-T1 has the best search efficiency, followed by COF, whilst RSL-T2 is the worst.

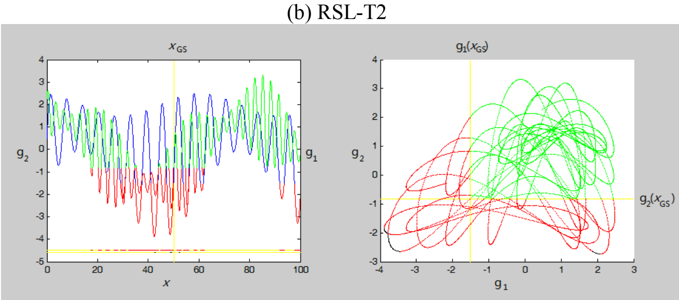
The average results of all 200 test cases are given in Table 1, where R_{PFXR} is a ratio by dividing the Pareto front associated x

value range by the length of the solution space, N_{FCPF} indicates in how many test cases a method has found the complete Pareto front, R_{EXR} is a ratio by dividing the explored x value range of a method by the length of the solution space, N_{TLM} is the total number of local minima, N_{ELM} is the number of local minima explored by a method, and CT stands for computational time. From Table 1 we have the following observations:

(a) RSL-T1



(b) RSL-T2



(c) COF

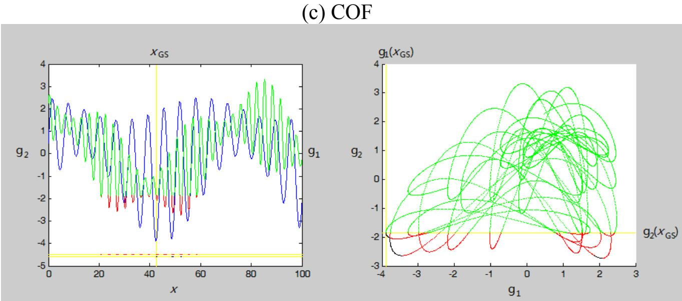


Fig.2 Results of a test case in finding the complete Pareto front.

- All methods have found the complete Pareto front in all the 200 test cases, which proves experimentally the correctness of the Theoretical results in Section 2.
- Among the three methods, RSL-T1 is the best in terms of either R_{EXR}, N_{ELM}, or CT. In other words, compared with RSL-T2 and COF, RSL-T1 explores about 3 times less x value ranges and roughly 10% less local minima, and saves about 30% computational time to find the Pareto front. Put simply, RSL-T1 can find the complete Pareto front by exploring the smallest proportion of solution space and consuming the least computing resources.
- Actually, COF can be viewed as a special case of RSL-T2 where the given solution X_{GS} is always chosen as either $x^*_{LM}(1,1)$ or $x^*_{LM}(2,1)$. Compared with a RSL-T2 with a randomly chosen X_{GS}, it is then not a surprise to see that COF has a better performance than RSL-T2.
- The obvious advantage of RSL-T1 against RSL-T2 and COF implies that the condition (7) in Theorem 1, although it seems more restrictive, is practically much more effective than the condition (10) in Theorem 2.
- R_{EXR} for RSL-T1 is about 6 times R_{PFXR}, which means the x value ranges explored by RSL-T1 is a fairly small cover of the complete Pareto front associated x value ranges, further demonstrating the search efficiency of RSL-T1.
- Comparing N_{ELM} for RSL-T1 with N_{TLM}, shoes that it is usually not necessary to have all local minima. RSL-T1 can find the complete Pareto front with fewer than half of them but the local minimum used should be among those smallest to guarantee finding the complete Pareto front.

TABLE I
AVERAGE RESULTS TO FIND THE COMPLETE PARETO FRONT

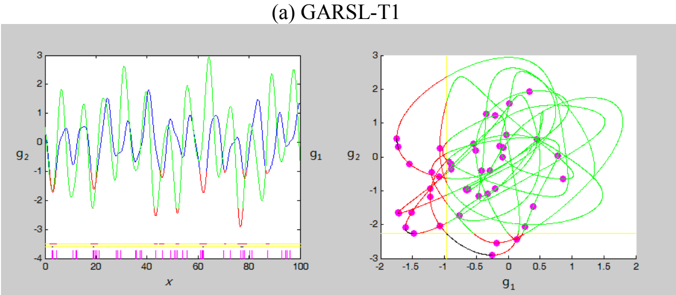
(R_{PFXR}=0.04; N_{TLM}=21.05)	RSL-T1	RSL-T2	COF
N_{FCPF}	**200.00**	**200.00**	**200.00**
R_{EXR}	**0.23**	0.74	0.52
N_{ELM}	**9.36**	12.85	10.37
CT (second)	**0.19**	0.33	0.26

B. Approximation of Pareto Front

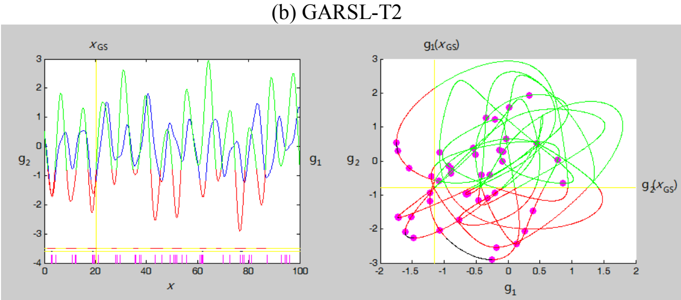
The ability of the proposed method in Section 3 to find the complete Pareto front relies, in theory, on the assumption that a continuous problem has a finite number of local minima in terms of each single objective function, and that there is an effective method to find all these local minima. This assumption is however unrealistic in many real-world multi-objective optimization applications, as it is often uncertain whether the number of local minima is finite, or whether a method could find all such local minima (sometimes even the first best solution to a single-objective problem cannot be guaranteed). Therefore, existing methods such as multi-objective evolutionary algorithms focusing on approximating the true Pareto front seem still to be practically more desirable. Here we investigate whether the proposed method is still applicable to problems with infinite local minima and/or without an effective method to find all local minima. Moreover, if it is applicable, then how good will its performance be when compared with existing methods? We show by simulation in this subsection that the proposed method can also be used to approximate the true Pareto front effectively when there is no guarantee of the availability of all local minima.

To approximate the true Pareto front, we actually do not need to modify the proposed method. We just feed the method with a set of randomly generated assumed local minima to start the search procedure, and as will be demonstrated later, the method will perform almost as well as it is fed with all true local minima. Regarding how to generate a set of assumed local minima, there are many methods for multi-optima problems, which are particularly good at finding multiple (not necessarily all) local optima. Multi-optima evolutionary algorithms are some examples of this kind [28]-[30]. In this subsection, we will use the multi-optima evolutionary algorithm in [28] to generate a set of assumed local minima in terms of each single objective function of (20) and (21), which will then be used to start RSL-T1 and RST-2, in order to approximate the Pareto front. As revealed in Table 1, for the problem defined by (19)-(22), RSL needs approximately 10 local minima on average to finish the search job. Therefore, a relatively small population size for the multi-optima evolutionary algorithm may be enough to find the first 10 smallest local minima or similar solutions. Here we set the population size to 50. The multi-optima evolutionary algorithm is run twice, one time for g_1 and the other time for g_2. Once the evolutionary algorithm stops, we choose the first 25 best solutions in terms of g_1 and also the first 25 best solutions in terms of g_2 in the final generations, and feed them to RSL as assumed local minima. To distinguish from RSL-T1 and RSL-T2 in the previous subsection, hereafter we denote multi-optima evolutionary algorithm based RSL as GARSL-T1 and GARSL-T2.

(a) GARSL-T1



(b) GARSL-T2



(c) NSGA-II

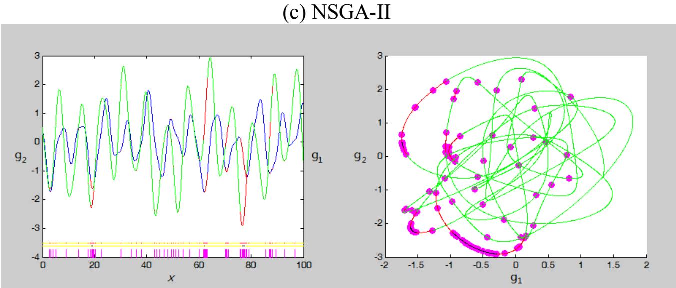


Fig.3 Results of a test case in approximating the true Pareto front.

TABLE II
AVERAGE RESULTS TO APPROXIMATE THE TRUE PARETO FRONT

(R_{PFXR}=0.04)	GARSL-T1	GARSL-T2	NSGA-II
N_{FCPF}	**191**	184	167
R_{EXR}	0.31	0.82	**0.25**
CT (second)	**4.28**	4.61	11.67

To better assess the performance of RSL in approximating the Pareto front, we compare it with the most acknowledged multi-objective evolutionary algorithm, NSGA-II in [16]. NSGA-II needs a relatively large population size so that in the final generation we might have a reasonably large number of Pareto-optimal (or near Pareto-optimal) solutions to effectively approximate the Pareto front of continuous problems. In this study, we set the population size to 100 for NSGA-II.

For both evolutionary algorithms used, i.e., the multi-optima evolutionary algorithm for RSL to start, and NSGA-II for the sake of comparison, the number of generations for evolving the population is 200, the mutation probability is set as 0.1, the crossover probability is set as 0.5, and the first 10 best solutions in a generation will be directly copied into the next generation.

Fig.3 compares the results of GARSL-T1, GARSL-T2 and NSGA-II in a test case, where for GARSL-T1, GARSL-T2, the pink vertical short lines and dots indicate the assumed local minima generated by the multi-optima evolutionary algorithm for starting the search procedure in RSL, while for NSGA-II, the pink vertical short lines and dots are associated with all solutions in the last generation of NSGA-II. In Fig.3, one can see that, even with no guarantee of a set of all local minima, both GARSL-T1 and GARSL-T2 can still work out a complete cover of the true Pareto front. The projection of the final generation of NSGA-II in the objective space has a good distribution along the true Pareto front.

Table 2 summarizes the results of GARSL-T1, GARSL-T2 and NSGA-II in all 200 test cases. For any two solutions in the final generation of NSGA-II, say x_1 and x_2, whose distance is smaller than a threshold of 0.5, i.e., $|x_1 - x_2|<0.5$, we view it as if NSGA-II has explored the whole range of $[x_1,x_2]$. The expanding step Δx for RSL is just 0.01, the threshold of 0.5 is then 50 times Δx. Therefore, it should be fair enough to use the threshold of 0.5 to judge whether NSGA-II has found a complete cover of the true Pareto front. It should be noted that the R_{EXR} of NSGA-II is not based on the x value ranges where NSGA-II has actually explored from the first generation to the last generation. Instead, it is only based on the 100 solutions in the last generation. Therefore, the R_{EXR} of NSGA-II is not comparable to those of GARSL-T1 and GARSL-T2. From Table 2, one can see that:

- In most test cases, GATSL-T1 and GATSL-T2 can still find a complete cover of the true Pareto front, but they do fail in a few of the 200 test cases. By checking the details of such failed test cases, it is found that if the multi-optima evolutionary algorithm fails to find some key local minima (e.g., the first few smallest local minima which are far away from the point that triggers the termination criteria), then it is possible that the associated x value ranges will be missed by GATSL-T1 and GATSL-T2. Therefore, if the given set of assumed local minima is reasonably close to those key true local minima, then the finding of the complete Pareto front can still be guaranteed.
- Basically, GARSL-T1 is better than GARSL-T2, which is in line with the results reported in the previous subsection, and further demonstrates the advantage of Theorem 1 against Theorem 2.
- In general, NSGA-II is worse than GARSL-T1 and

GARSL-T2 in terms of both solution quality and computational efficiency. NSGA-II seems good at finding a large continuous piece of Pareto front. However, if there is a tiny fraction of Pareto front separated far away from other parts of Pareto front, then NSGA-II often struggles to find it.

- From Table 2, one can see that a lack of the availability of all local minima will not jeopardize the practicability of the proposed method. Instead, when compared with NSGA-II, the new method still exhibits a satisfactory performance in the approximation of Pareto front.

V. CONCLUSION

This paper proposes an effective method that can guarantee finding the complete Pareto front for a special class of continuous multi-objective optimization problems. Theoretically, as long as the number of local optima in terms of each single objective function is finite and there exists an approach to find all of such local optima, the proposed method can work properly. Those well developed methods for multiple optima problems, such as multi-optima evolutionary algorithms, provide a concrete foundation to the practicability of the proposed method, the effectiveness and efficiency of which are demonstrated in a comparative experiment. Future research work may include: extending and modifying the theoretical conditions to suit more other kinds of continuous problems; investigating the possibility of developing some more effective search procedures which can identify the real Pareto front by searching as a small portion of the solution space as possible; testing the reported theoretical work in some real-world complex multi-objective problems.

REFERENCES

[1] J. Figueira, and S. Greco, *Multiple criteria decision analysis: state of the art surveys*. Matthias Ehrgottc, 2005.

[2] L. Raamesh and G.V. Uma, "Data mining based optimization of test cases to enhance reliability of the testing," in *Advances in Computing and Information Technology*, edited by D.W. Wyld, M. Wozniak, N. Chaki, N. Meghanathan, and D. Nagamalai. Springer Paddyfield, 2011, pp. 89-98.

[3] N. Barr, *Economics of the welfare state*. New York, Oxford University Press (USA), 2004.

[4] R.E. Steuer, *Multiple Criteria Optimization: Theory, Computations, and Application*. New York: John Wiley & Sons, Inc., 1986.

[5] Y. Sawaragi, H. Nakayama, and T. Tanino, *Theory of Multiobjective Optimization (vol. 176 of Mathematics in Science and Engineering)*. Orlando, FL: Academic Press Inc., 1985.

[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, pp. 631–657, 1998.

[7] I. Das and J. Dennis, "Normal-Boundary Intersection: An Alternate Method For Generating Pareto Optimal Points In Multicriteria Optimization Problems," *NASA Contractor Report 201616*, *ICASE Report No. 96-62,* 1996.

[8] A. Messac, A. Ismail-Yahaya, and C.A. Mattson, "The normalized normal constraint method for generating the Pareto front," *Structural and multidisciplinary optimization*, vol. 25, no. 2, pp. 86–98, 2003.

[9] A. Messac and C. A. Mattson, "Normal constraint method with guarantee of even representation of complete Pareto front," *AIAA journal*, vol. 42, no. 10, pp. 2101–2111, 2004.

[10] T. Erfani and S.V. Utyuzhnikov, "Directed Search Domain: A Method for Even Generation of Pareto Front in Multiobjective Optimization," *Engineering Optimization*, vol. 43, no. 5, pp. 467-484, 2011.

[11] R.T.Marler and J.S.Arora, "Survey of multi-objective optimization methods for engineering," *Struct Multidisc Optim*, vol.26, pp. 369-395, 2004.

[12] A. Konak, D.W. Coit, and A.E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering and System Safety*, vol. 91, no.9, pp. 992–1007, 2006.

[13] M. Delgado, M.P. Cuellar, and M.C. Pegalajar, "Multiobjective Hybrid Optimization and Training of Recurrent Neural Networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B,* vol. 38, no. 2, pp, 381-403, 2008.

[14] D.F. Jones, S.K. Mirrazavi, and M. Tamiz, "Multiobjective meta-heuristics: an overview of the current state-of-the-art," *Eur J Oper Res*, vol. 137, no. 1 pp. 1–9, 2002.

[15] N. Srinivas, and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–48, 1994.

[16] K. Deb, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation,* vol. 6, no. 2, pp. 182-197, 2002.

[17] W.F. Leong and G.G. Yen, "PSO-Based Multiobjective Optimization With Dynamic Population Size and Adaptive Local Archives," *IEEE Transactions on Systems, Man, and Cybernetics, Part B,* vol. 38, no. 5, pp. 1270-1293, 2008.

[18] J.D. Knowles and D.W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–72, 2000.

[19] S. Bandyopadhyay, S.K. Pal, and B. Aruna, "Multiobjective GAs, Quantitative Indices, and Pattern Classification," *IEEE Transactions on Systems, Man and Cybernetics, Part B,* vol. 34, no. 5, pp. 2088-2099, 2004.

[20] X.J. Lei and Z.K. Shi, "Overview of multi-objective optimization methods," *J Syst Eng Electron*, vol. 15, no. 2, pp. 142–6, 2004.

[21] K. Deb, Multi-objective optimization using evolutionary algorithms. New York: Wiley, 2001.

[22] D.A. Van Veldhuizen, and G.B. Lamont, "Multiobjective evolutionary algorithms: analyzing the state-of-the-art," *Evolutionary Computation*, vol. 8, no. 2, pp. 125–47, 2000.

[23] X.F. Zou, Y. Chen, M.Z. Liu, and L.S Kang, "A New Evolutionary Algorithm for Solving Many-Objective Optimization Problems," *IEEE Trans on Systems, Man, and Cybernetics, Part B*, vol. 38, no. 5, pp. 1402-1412, 2008.

[24] E, Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–95, 2000.

[25] D. Craft, T. Halabi, H. Shih, and T. Bortfeld, "Approximating convex Pareto surfaces in multiobjective radiotherapy planning," *Medical Physics*, vol. 33, no. 9, pp. 3399–3407, 2006.

[26] X.B. Hu, M. Wang, and E. Di Paolo, "Calculating Complete and Exact Pareto Front for Multi-Objective Optimization: A New Deterministic Approach for Discrete Problems," *IEEE Trans on Systems, Man, and Cybernetics, Part B*, vol.43, no.3, pp. 1088-1101, 2013.

[27] X.B. Hu, M. Wang, Q. Ye, Z.G. Han, and M.S. Leeson, "Multi-Objective New Product Development by Complete Pareto Front and Ripple-Spreading Algorithm", Neurocomputing, under revision, 2014

[28] R.E. Dorsey and W.J. Mayer, "Genetic Algorithms for Estimation Problems With Multiple Optima, Nondifferentiability, and Other Irregular Features," *Journal of Business & Economic Statistics*, vol.13, no.1, pp. 53-66, 1995.

[29] A.B. Hadj-Alouane and J.C. Bean, "A Genetic Algorithm for the Multiple-Choice Integer Program," *Operation Research*, vol. 45, no.1, pp. 92-101, 1997.

[30] S. Chen, "Locust Swarms – A New Multi-Optima Search Technique", *in Proceedings of the IEEE Congress on Evolutionary Computation*, 2009, pp. 1745–1752.