Hybrid Line Search for Multiobjective Optimization

Crina Grosan[1,2] and Ajith Abraham[1]

[1]Faculty of Information Technology, Mathematics and Electrical Engineering
Centre for Quantifiable Quality of Service in Communication Systems
Centre of Excellence
Norwegian University of Science and Technology
Trondheim, Norway
[2] Department of Computer Science
Babes-Bolyai University Cluj-Napoca, Romania
cgrosan@cs.ubbcluj.ro, ajith.abraham@ieee.org

Abstract. The aggregation of objectives in multiple criteria programming is one of the simplest and widely used approach. But it is well known that these techniques sometimes fail in different aspects for determining the Pareto frontier. This paper proposes a new line search based approach for multicriteria optimization. The objectives are aggregated and the problem is transformed into a single objective optimization problem. Then the line search method is applied and an approximate efficient point is lacated. Once the first Pareto solution is obtained, a simplified version of the former one is used in the context of Pareto dominance to obtain a set of efficient points, which will assure a thorough distribution of solutions on the Pareto frontier. In the current form, the proposed technique is well suitable for problems having multiple objectives (it is not limited to bi-objective problems). the functions to be optimized must be continuous twice differentiable. In order to assess the effectiveness of this approach, some experiments were performed and compared with two recent well known population-based metaheuristics ParEGO [8] and NSGA II [2]. When compared to ParEGO and NSGA II, the proposed approach not only assures a better convergence to the Pareto frontier but also illustrates a good distribution of solutions. From a computational point of view, of the line search converge within a short time (average about 150 milliseconds) and the generation of well distributed solutions on the Pareto frontier is also very fast (about 20 milliseconds). Apart from this, the proposed technique is very simple, easy to implement and use to solve multiobjective problems.

1 Introduction

The field of multicriteria programming abounds in methods for dealing with different kind of problems. Nevertheless, there is still space for new approaches, which can better deal with some of the difficulties encountered by the previous approaches. There are two main classes of approaches suitable for multiobjective optimization: scalarization methods and nonscalarizing methods. These

R. Perrott et al. (Eds.): HPCC 2007, LNCS 4782, pp. 62–73, 2007.

approaches convert the Multiobjective Optimization Problem (MOP) into a Single Objective Optimization Problem (SOP), a sequence of SOPs, or into another MOP. There are several scalarization methods reported in the literature: weighted sum approach, weighted t-th power approach, weighted quadratic approach, ε-constraint approach, elastic constraint approach, Benson approach, etc. are some of them [5]. Since the standard weighted sum encounters some difficulties, several other methods have been proposed to overcome the major drawbacks of this method. These include: Compromise Programming [4], Physical Programming [9], Normal Boundary Intersection (NBI) [1], and the Normal Constraint (NC) [9] methods. There is also a huge amount of work reported on population-based mataheuristics for MOP [5].

In this paper, we propose a new approach which uses a scalarization of the objectives in a way similar to the weighted t-th power approach (where t is 2 and the coefficients values are 1).

A line search based technique is used to obtain an efficient solution. Starting with this solution, a set of efficient points are further generated, which are widely distributed along the Pareto frontier using again a line search based method but involving Pareto dominance relationship.

Empirical and graphical results and illustrations obtained by the proposed approach are compared with two well known population based metaheuristics namely ParEGO [8] and NSGA II [2].

2 Line Search Generator of Pareto Frontier

The line search [6] is a standard and well established optimization technique. The standard line search technique is modified so that it is able to generate the set of non-dominated solutions for a MOP. The approach proposed is called **L**ine search **G**enerator of **P**areto frontier (LGP) and it comprises of two phases: first, the problem is transformed into a SOP and a solution is found using a line search based approach. This is called as *convergence phase*. Second, a set of Pareto solutions are generated starting with the solution obtained at the end of convergence phase. This is called as *spreading phase*. The *convergence* and *spreading* phases are described below.

Consider the MOP formulated as follows:

Let $\Re^m$ and $\Re^n$ be Euclidean vector spaces referred to as the decision space and the objective space. Let $X \subset \Re^m$ be a feasible set and let f be a vector-valued objective function f: $\Re^m \rightarrow \Re^n$ composed of n real-valued objective functions $f=(f_1, f_2,\ldots, f_n)$, where f_k: $\Re^m \rightarrow \Re$, for k=1,2,…, n. A MOP is given by:

$$\min\ (f_1(x),\ f_2(x),\ldots,\ f_n(x)),$$

$$\text{subject to } x \in X.$$

2.1 Convergence Phase

The MOP is transformed into a SOP by aggregating the objectives using an approach similar to the weighted t-th power approach. We consider t= 2 and the values of weights equal to 1. The obtained SOP is:

$$\min F = \sum_{i=1}^{n} f_i^2(x)$$
$$\text{subject to } x \in X.$$

As an important note, we would like to mention that the value 2 for t works fine if the objective functions are positive (which is the case of our experiments). But if at least one objective function is negative, then an odd value (for instance 3) must me used for t. Any of these values (2 or 3) works fine for our examples and will not influence the results.

A modified line search method is used to find the optimum of this problem. The modification proposed in this paper for the standard line search technique refers to direction and step setting and also the incorporation of a re-start procedure. To fine tune the performance, the first partial derivatives of the function to optimize are also made use of. The proposed modifications refer to:

- the setting of the direction and step
- the re-starting of the line search method

After a given number of iterations, the process is restarted by reconsidering other arbitrary starting point which is generated by taking into account the result obtained at the end of previous set of iterations.

Direction and step setting. Initially, several experiments were performed in order to set an adequate value for the direction. The standard value +1 or -1 was used and for some functions the value -1 was favorable to obtain good performance. Some experiments were also performed by setting the direction value as being a random number between 0 and 1. It was found that the usage of random number helped to obtain overall very good performance for the entire considered test functions. But usage of the value -1 for direction, obtains almost the same performance similar to that obtained with a random value. So, either of these values (the random one and the value -1) may be used for better performance.

The step is set as follows:

α_k=2+$\frac{3}{2^{2k}+1}$ (1)

where k refers to the iteration number.

The modified *line search* technique is summarized as follows:

Line_search()

Set k=1 (Number of iterations)
Repeat
  for i=1 to No of variables
    p_k=random; //or p=-1;

α_k=2+$\frac{3}{2^{2k}+1}$
$x_i^{k+1} = x_i^k + p_k \cdot \alpha_k$
endfor
if $F(x_{k+1})$>$F(x_k)$ then x^{k+1}=x_k.
k=k+1
Until k=Number of iterations (a priori known).

Remarks

(i) The condition:
if $F(x_{k+1})$>$F(x_k)$ then x_{k+1}=x_k
allows to move to the new generated point only if there is an improvement in the quality of the function.

(ii) Number of iterations for which line search is applied is apriori known and is usually a small number. For the experiments reported in this paper, the number of these iterations was set to 10.

(iii) When restarting the line search method (after the insertion of the re-start technique) the value of the iterations number starts again from 1 (this should not be related to the value of α after the first set of iterations (and after each of the following iterations)).

Several experiments were attempted to set a value for the step, starting with random values (until a point is reached for which the objective function achieves a better value); using a starting value for the step and generating random numbers with Gaussian distribution around this number, etc. As a result of the initial experiments performed, it was decided to use equation (1) to compute the step size. But, of course, there are also several other ways to set this.

Incorporation of re-start procedure. In order to restart the algorithm the result obtained in the previous set of iterations (denote it by x) is taken into account and the steps given below are followed:

For each dimension $i$of the point x, the first partial derivative with respect to this dimension is calculated. This means the gradient of the objective function is calculated which is denoted by g. Taking this into account, the bounds of the definition domain for each dimension are re-calculated as follows:

if $g_i = \frac{\partial F}{\partial x_i} > 0$ then *upper bound* $=x_i$;
if $g_i = \frac{\partial F}{\partial x_i} < 0$ then *lower bound* $=x_i$

The search process is re-started by re-initializing a new arbitrary point between the newly obtained boundaries.

2.2 Spreading Phase

At the end of the convergence phase, a solution is obtained. This solution is considered as an efficient (or Pareto) solution. During this phase and taking into

account of the existing solution, more efficient solutions are to be generated so as to have a thorough distribution of all several good solutions along the Pareto frontier. In this respect, the line search technique is made use of to generate one solution at the end of each set of iterations. This procedure is applied several times in order to obtain a larger set of non-dominated solutions. The following steps are repeated in order to obtain one non-dominated solution:

Step 1. A set of nondominated solutions found so far is archived. Let us denote it by *NonS*. Initially, this set will have the size one and will only contain the solution obtained at the end of *convergence phase*.

Step 2. We apply line search for one solution and one dimension of this solution at one time. For this:

Step 2.1. A random number i between one and |NonS| ($|\cdot|$ denotes the cardinal) is generated. Denote the corresponding solution by $nonS_i$.

Step 2.2. A random number j between one and the number of dimensions (the number of decision variables) is generated. Denote this by $nonS_{ij}$.

Step 3. Line search is applied for $nonS_{ij}$.

Step 3.1. Set p=1 (the random value also works fine).

Step 3.2. Set α (which depends on the problem, on the number of total non-dominated solutions which are to be generated, etc.).

Step 3.3. The new obtained solution *new_sol* is identical to $nonS_i$ in all dimensions except dimension j which is:
$new\_sol_j = nonS_{ij} + \alpha \cdot p$

Step 3.4. if ($new\_sol_j >$ *upper bound*) or ($new\_sol_j <$ *lower bound*)
then new_sol_j = *lower bound* + random · (*upper bound* – *lower bound*).

Step 4. if $F(new\_sol) > F(nonS_1)$
then discard *new_sol*
else if *new_sol* is nondominated with respect to the set *NonS*
then add *new_sol* to *NonS* and increase the size on *NonS* by 1.
Go to step 2.

Step 5. Stop

These steps are repeated until a set on nondominated solutions of a required size is obtained. In our experiments the size of this set is 100.

Note that this procedure it very fast and it takes less than 20 milliseconds to obtain 100 non-dominated solutions.

3 Experiments and Comparisons

In order to assess the performance of LGP, some experiments were performed using some well known bi-objective and three-objective test functions, which are adapted from [3], [7]. These test functions were also used by the authors of ParEGO [8] and NSGA II [2], which are well known in the computational intelligence community as very efficient techniques for multiobjective optimization.

Table 1. Parameters used in experiments by ParEGO and NSGA II. d denotes the number of decision parameter dimensions.

ParEGO		**NSGA II**	
Parameter	*Value*	*Parameter*	*Value*
Initial population in latin hypercube	$11d - 1$	Population size	20
Total maximum evaluations	250	Maximum generations	13
Number of scalarizing vectors	11 for 2 objectives 15 for 3 objectives	Crossover probability	0.9
Scalarizing function	Augmented Tchebycheff	Real value mutation probability	$1/d$
Internal genetic algorithm evaluations per iteration	200,000	Real value SBX parameter	10
Crossover probability	0.2	Real value mutation parameter	50
Real value mutation probability	$1/d$		
Real value SBX parameter	10		
Real value mutation parameter	50		

Details about implementation of these two techniques may obtained from [2] and [8]. Parameters used by ParEGO and NSGA II (given in Table 1) and the results obtained by these two techniques are adapted from [8].

A set of 100 non-dominated solutions obtained by LGP, ParEGO, NSGA II is compared in terms of dominance and convergence to the Pareto set. For the first comparison, two indices were computed for each set of two comparisons: number of solution obtained by the first technique which dominate solutions obtained by the second technique and number of solutions obtained by the first technique which are dominated by the solutions obtained by the second technique.

For two sets of A and B of solutions, which are compared, indices are denoted by *Dominate(A, B)* and *Dominated(A, B)* respectively. Visualization plots are used to illustrate the distribution of solutions on the Pareto frontier.

LGP uses only three parameters:

number of re-starts: 20;

number of iteration per each re-start: 10;

$\alpha$for the spreading phase (which is set independent for each test function).

3.1 Test Function DTLZ1a

The test function DTLZ1a is a two objective test function and has 6 variables [8]. It is given by:

$$minimize\ f_1 = \frac{1}{2}x_1(1+g)$$
$$minimize\ f_2 = \frac{1}{2}(1-x_1)(1+g)$$

$$g = 100\left[5 + \sum_{i=2}^{6}\left((x_i - 0.5)^2 - \cos\left(2\pi\left(x_i - 0.5\right)\right)\right)\right]$$
$x_i \in [0, 1]$, i=1, ..., n, n=6.

The Pareto set for this function consists of all solutions where all by the first decision variables are equal to 0.5 and the first decision variable may take any value between 0 and 1.

For this test function, the value of α for the spreading phase is set to 0.01. The convergence to the Pareto frontier and the distribution of solutions obtained by LGP, ParEGO and NSGA II for the test function DTLZ1a is depicted in Figure 1. Different sizes of the objective space are illustrated in order to incorporate all solutions obtained by all techniques. It is obvious that LGP assure a very good convergence and distribution for this function. From the results presented in Table 2 it can be observed that none of the solutions obtained by LGP are dominated neither by ParEGO or by NSGA II, while solutions obtained by LGP dominate all 100 solutions obtained by ParEGO and NSGA II. 88 of the solutions obtained by NSGA II are dominated by solutions obtained by ParEGO while 69 of the solutions obtained by ParEGO are dominated by solutions obtained by NSGA II.

Table 2. The dominance between solutions obtained by LGP, ParEGO and NSGA II for test function DTLZ1a

Dominate	ParEGO	NSGA II	***Dominate***	LGP	NSGA II	***Dominate***	LGP	ParEGO
LGP	100	100	ParEGO	0	69	NSGA II	0	88
Dominated	ParEGO	NSGA II	***Dominated***	LGP	NSGA II	***Dominated***	LGP	ParEGO
LGP	0	0	ParEGO	100	88	NSGA II	100	69

3.2 Test Function DTLZ4a

Test function DTLZ4a has three objective functions and 8 decision variables and is given by:

$$minimize\ f_1 = (1+g)\cos\left(\frac{x_1^{100}\pi}{2}\right)\cos\left(\frac{x_2^{100}\pi}{2}\right)$$
$$minimize\ f_2 = (1+g)\cos\left(\frac{x_1^{100}\pi}{2}\right)\sin\left(\frac{x_2^{100}\pi}{2}\right)$$
$$minimize\ f_3 = (1+g)\sin\left(\frac{x_1^{100}\pi}{2}\right)$$
$$g = \sum_{i=3}^{8}\left(x_i - 0.5\right)^2$$
$x_i \in [0, 1]$, i=1, ..., n, n=8.

The Pareto front is 1/8 of the unit sphere centered in origin. The Pareto optimal set consist of all solutions but the first two decision variables are equal to 0.5 and the first two decision variables may take any value between 0 and 1.

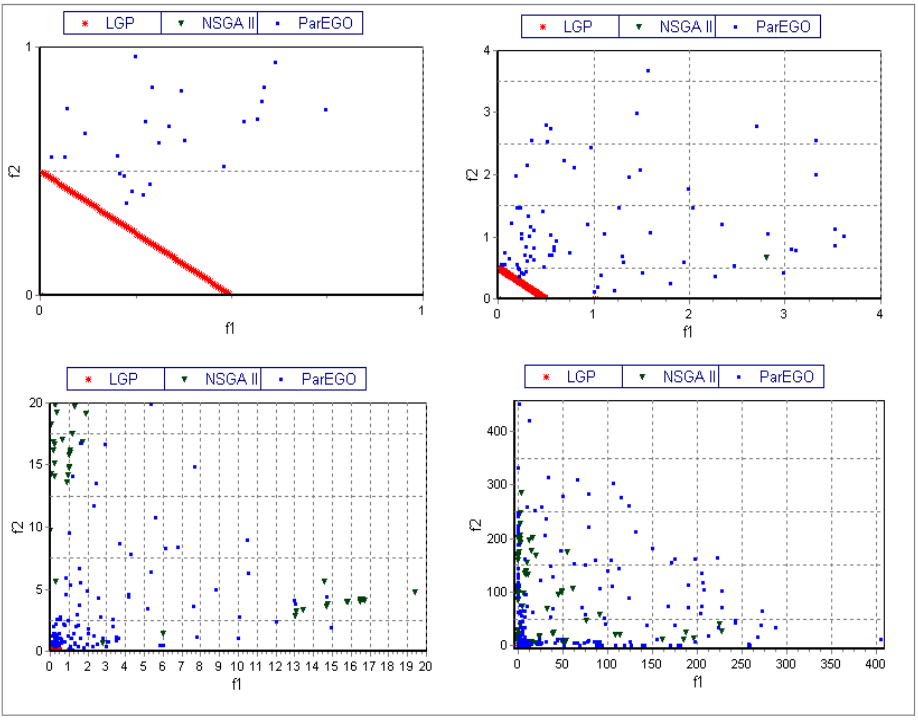

Fig. 1. Distribution of solutions on the Pareto frontier obtained by LGP, ParEGO and NSGA II for test function DTLZ1a

For test function DTLZ4a the value of α is set to 0.2. The distribution of solutions on the Pareto frontier and the convergence to the Pareto frontier for all the three algorithms is depicted in Figure 2. For test function DTLZ4a the value of α is set to 0.2.

From Figure 2 it can be observed that, compared to ParEGO and NSGA II, LGP is assuring a very good convergence. The latter two approaches are not converging very well with the parameters used.

As evident from Table 3 none of the solutions obtained by LGP are dominated neighter by ParEGO or by NSGA II while solutions obtained by LGP dominate all 100 solutions obtained by ParEGO and NSGA II. 87 of the solutions obtained by NSGA II are dominated by solutions obtained by ParEGO while 54 of the solutions obtained by ParEGO are dominated by solutions obtained by NSGA II.

3.3 Test Function DTLZ7a

This test function has 3 objectives and 8 decision variables and it is given by:

$$minimize\ f_1 = x_1$$
$$minimize\ f_2 = x_2$$
$$minimize\ f_3 = (1+g)h$$

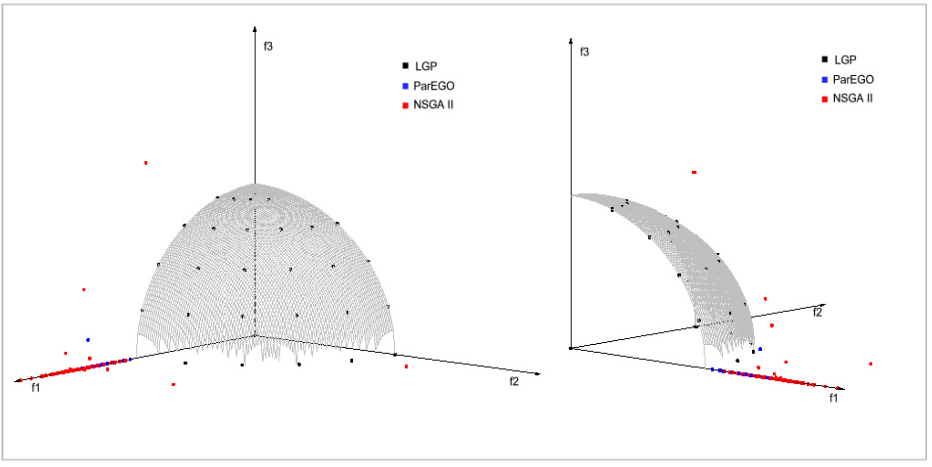

Fig. 2. Convergence to the Pareto frontier and distribution of solutions obtained by LGP, ParEGO and NSGA II on the Pareto frontier for test function DTLZ4a (view from different angles)

Table 3. The dominance between solutions obtained by LGP, ParEGO and NSGA II for test function DTLZ4a

Dominate	ParEGO	NSGA II	***Dominate***	LGP	NSGA II	***Dominate***	LGP	ParEGO
LGP	100	100	ParEGO	0	87	NSGA II	0	54
Dominated	ParEGO	NSGA II	***Dominated***	LGP	NSGA II	***Dominated***	LGP	ParEGO
LGP	0	0	ParEGO	100	54	NSGA II	100	87

$$g = 1 + \frac{9}{6}\sum_{i=3}^{8} x_i$$
$$h = 3 - \sum_{i=1}^{2}\left[\frac{f_i}{1+g}\left(1 + \sin\left(3\pi f_i\right)\right)\right]$$
$$x_i \in [0, 1],\ i{=}1, \ldots, n,\ n{=}8.$$

The Pareto front has four discontinuous regions and the Pareto set consists of all solutions where all by the first two decision variables are equal to 0.

The test function DTLZ7a has 4 discontinuous Pareto regions. LGP is able to converge very well and it is able to spread into the all four disconnected Pareto regions from a single starting point. The value of α used is 1, but there is not much difference between different values of α. As evident from Figure 3, both ParEGO and NSGA II ar far from the Pareto front in terms of convergence. Also, none of the solutions obtained by LGP is dominated by neither ParEGO or NSGA II. 17 solutions obtained by ParEGO are dominated by solutions obtained

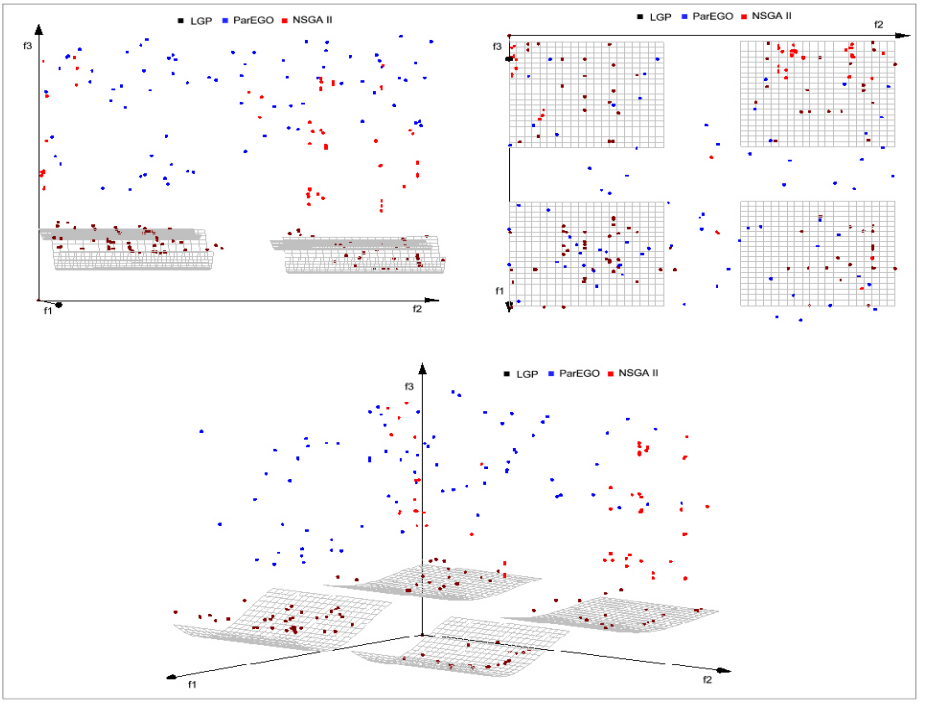

Fig. 3. Convergence to the Pareto frontier and distribution of solutions obtained by LGP, ParEGO and NSGA II on the Pareto frontier for test function DTLZ7a (view from different angles)

Table 4. The dominance between solutions obtained by LGP, ParEGO and NSGA II for test function DTLZ7a

Dominate	ParEGO	NSGA II	***Dominate***	LGP	NSGA II	***Dominate***	LGP	ParEGO
LGP	100	100	ParEGO	0	80	NSGA II	0	17
Dominated	ParEGO	NSGA II	***Dominated***	LGP	NSGA II	***Dominated***	LGP	ParEGO
LGP	0	0	ParEGO	100	17	NSGA II	100	80

by NSGA II while 80 of the solutions obtained by NSGA II are dominated by solutions obtained by ParEGO.

4 Conclusions

The paper proposes a new approach for multiobjective optimization which uses an aggregation of objectives and transforms the MOP into a SOP. A line search

based technique is applied in order to obtain one solution. Starting from this solution a simplified version of the initial line search is used in order to generate solutions with a well distribution on the Pareto frontier. Numerical experiments performed show that the proposed approach is able to converge very fast and provide a very good distribution (even for discontinuous Pareto frontier) while compared with state of the art population based metaheuristics such as ParEGO and NSGA II.

Compared to NSGA II and ParEGO, LGP has only few parameters to adjust. It is computationally inexpensive, taking less than 200 milliseconds to generate a set of nondominated solutions well distributed on the Pareto frontier.

The only inconvenience is that LGP involves first partial derivatives which makes it be restricted to a class of problems which are continuous twice differentiable. But almost all practical engineering design problems are continuous differentiable.

One of the further work ideas is to find a better way to set the value of α. In this paper, we considered different α values until we achieved a satisfactory distribution. Also, we would like to extend LGP to deal with constraint multiobjective optimization problems.

References

1. Das, I., Dennis, J.: A Closer Look at Drawbacks of Minimizing Weighted Sums of Objective for Pareto Set Generation in Multicriteria Optimization Problems. Structural Optimization 14, 63–69 (1997)
2. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transaction on Evolutionary Computation 6(2), 181–197 (2002)
3. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings of the Congress on Evolutionary Computation (CEC-2002), USA, pp. 825–830 (2002)
4. Chen, W., Wiecek, M.M., Zhang, J.: Quality Utility – A Compromise Programming Approach to Robust Design. Journal of Mechanical Design 121, 179–187 (1999)
5. Ehrgott, M., Wiecek, M.M.: Mutiobjective Programming, In Multiple Criteria Decision Analisys: State of the art surveys. In: Figueira, J., et al. (eds.) International Series in Operations Research & Management Science, vol. 78, Springer, New York (2005)
6. Gould, N.: An introduction to algorithms for continuous optimization, Oxford University Computing Laboratory Notes (2006)
7. Huband, S., Hingston, P., Barone, L., While, L.: A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. IEEE Transactions on Evolutionary Computation 10(5), 477–506 (2006)
8. Knowles, J.: ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Transactions on Evolutionary Computation 10(1), 50–66 (2006)

9. Messac, A., Mattson, C.A.: Generating Well-Distributed Sets of Pareto Points for Engineering Design using Physical Programming. Optimization and Engineering 3, 431–450 (2002)
10. Messac, A., Ismail-Yahaya, A., Mattson, C.A.: The Normalized Normal Constraint Method for Generating the Pareto Frontier. Structural and Multidisciplinary Optimization 25(2), 86–98 (2003)