Evolutionary Algorithm for Multiple Objective Convex Integer Problems

Krasimira Genova[1] and Vassil Guliashki[2]

Abstract – **An interactive evolutionary algorithm is proposed to solve multiple objective convex integer problems. It combines two different strategies helping the Decision Maker (DM) to choose good preference direction and to focus the search process around a desired part of Pareto optimal front. These strategies lead to drastic reducing the number of iterations for moving the population to the set of Pareto-optimal points. An illustrative example is presented.**

Keywords – **Multi-objective optimization, Evolutionary algorithms, Reference direction, Pareto-optimal front.**

I. Introduction

In this paper is considered the multiple objective convex integer optimization problem, which can be stated in the following general form:

$$\text{Minimize} \quad \boldsymbol{f}(\boldsymbol{x}) = [\, f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})]^T \tag{1}$$

$$\text{subject to:} \quad \boldsymbol{g_j}(\boldsymbol{x}) \le 0, \qquad j = 1,2,..., m; \tag{2}$$

$$\boldsymbol{x_i}^{(L)} \le \boldsymbol{x_i} \le \boldsymbol{x_i}^{(U)}, \quad i = 1,2,\ldots, n; \tag{3}$$

$$\boldsymbol{x} \in \boldsymbol{Z^n}, \tag{4}$$

where $\boldsymbol{g_j}(\boldsymbol{x})$, $j = 1,2,\ldots, m$; are convex functions and $f_i(\boldsymbol{x})$, $i = 1,2,\ldots, k$; are nonlinear functions.

Below in the text of the paper we consider the term “solution” as a variable vector and the term “point” as a corresponding objective vector.

A solution $\boldsymbol{x} \in \mathbf{Z}^{\boldsymbol{n}}$ is a vector of n decision variables: $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$. The value $x_i^{(L)}$ is the known lower bound and the value $x_i^{(U}$ is correspondingly the upper bound of variable x_i. The solutions satisfying the constraints (2)-(4) constitute a feasible decision variable space $\boldsymbol{V} \subset \boldsymbol{Z^n}$. The objective functions (1) constitute a k-dimensional space, called objective space $\boldsymbol{F} \subset \boldsymbol{R}^k$. For each solution $\boldsymbol{x}$ in the decision variable space, there exists a point $\boldsymbol{f} \in \mathbf{R}^k$ in the objective space, denoted by $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f} = (f_1, f_2, \ldots, f_k)^T$.

The problem (1-4) does not posses an analytically defined optimal solution. To solve it vectors, containing values of each objective are compared during an optimization procedure. In this case the usual concepts are those proposed by Pareto [12].

The domination between two solutions is defined as follows (see [1, 3, 10]):

Definition 1. *A solution* $\mathbf{x}^{(1)}$ *is said to dominate the other solution* $\mathbf{x}^{(2)}$, *if both the following conditions are true*:

1. *The solution* $\boldsymbol{x}^{(1)}$ *is no worse(say the operator* $\prec$ *denotes worse and the operator* $\succ$ *denotes better) than* $\boldsymbol{x}^{(2)}$ *in all objectives*, or $f_j(\boldsymbol{x}^{(1)}) \not\prec f_j(\boldsymbol{x}^{(2)})$ for $j = 1,2,..., k$;.

2. *The solution* $\boldsymbol{x}^{(1)}$ *is strictly better than* $\boldsymbol{x}^{(2)}$ *in at least one objective*, or $f_j(\boldsymbol{x}^{(1)}) \succ f_j(\boldsymbol{x}^{(2)})$ for at least one $j \in \{1,2,..., k\}$.

All points which are not dominated by any other point $\boldsymbol{f} \in \boldsymbol{F}$ are called the non-dominated points of class one, or simply the non-dominated points. Usually the non-dominated points together make up a front in the objective space and are often visualized to represent a non-domination front. The points lying on the non-domination front, by definition, do not get dominated by any other point in the objective space, hence they are Pareto-optimal points (together they constitute the Pareto-optimal front), and the corresponding variable vectors are called Pareto-optimal solutions.

There are two ideal goals in the multi-objective optimization:

1. Find a set of solutions which are diverse enough to represent the entire range of the Pareto-optimal front, and

2. Find a set of Pareto-optimal solutions, which satisfy in the best way the DM’s preferences.

In this paper is proposed an interactive algorithm to solve the problem (1-4). The interactive algorithms are the most popular in solving multi-objective optimization problems. They consist of two alternate phases: 1. Interaction (dialogue) with the DM and 2. Generating solutions. Usually an appropriate single objective convex integer optimization problem is solved during the second phase. Such problems belong to the class of NP-hard problems (see for example [5, 11]). There does not exist an exact algorithm, which can solve these problems in time depending polynomially on the problem input data length or on the problem size. For this reason many researchers investigate approximate algorithms with polynomial computational complexity, which solve such kind optimization problems. For the past 20 years evolutionary multiple objective optimization (EMOO) methodologies have demonstrated their usefulness in finding a set of near Pareto-optimal solutions [1, 2, 4]. As a sequence many source codes – both commercial and free have been created and the EMO algorithms obtained wide application. During the World Congress of Computational Intelligence (WCCI) in Vancouver 2006, the EMOO has been evaluated as one of the three fastest growing fields of research and application among all computational intelligence topics.

In principle the evolutionary optimization (EO) algorithms use a population-based approach, in which the iterations are performed on a set of solutions (called population) and more than one solution is generated at each iteration. The main positive features making popular the EO algorithms are the following: (i) They do not require any derivative information;

[1] Krasimira Genova is with the Institute of Information Technologies, “Acad. G. Bonchev” str. Bl. 29A, 1113 Sofia, Bulgaria, E-mail: kgenova@iinf.bas.bg

[2] Vassil Guliashki is with the Institute of Information Technologies, “Acad. G. Bonchev” str. Bl. 29A, 1113 Sofia, Bulgaria, E-mail: vggul@yahoo.com

(ii) EO algorithms are relatively simple to implement; (iii) EO algorithms are flexible and robust, i.e. they perform very well on a wide spectrum of problems (see [6]); The use of a population in EO algorithms has a number of advantages (see [1]): 1) it provides an EO procedure with a parallel processing power, 2) it allows EO procedures to find multiple optimal solutions, thereby facilitating the solution of multi-modal and multi-objective optimization problems, and 3) it provides an EO algorithm with the ability to normalize decision variables (as well as objective and constraint functions) within an evolving population using the best minimum and maximum values in the population.

Some important disadvantages of EMOO algorithms are: (i) their convergence to the Pareto-optimal front could be slow and may require large number of iterations; (ii) they face difficulty in solving problems with a large number of objectives, i.e. they could obtain difficult a well representative set of Pareto-Optimal Solutions (see [4]).

The number of objectives as a convergence factor is considered in [13]. Good approach in solving problems with large number of objectives is to use the EMOO methodologies to find a preferred and smaller set of Pareto-optimal solutions, instead of the entire front [4]. In this way the DM can concentrate to explore only the regions of Pareto-optimal front, which are of interest to her/him. An accelerating technique for population based algorithms is proposed in [7]. A technique for quickly moving the population to the Pareto-optimal front is proposed in [8]. Some hybrid EMOO algorithms have been recently proposed to overcome the second mentioned disadvantage (see [4, 9]). They expand the use of classical multi-objective optimization procedures (see [10]) like reference point-, reference direction- and other type methods, proposing new approaches and hybrid techniques.

We propose two strategies for quickly moving the whole population to the Pareto-optimal front. They are included in the proposed algorithm for multiple objective convex integer problems (MOCIP-algorithm) and lead to drastic reduction of iteration number for convergence the search process to Pareto-optimal front. At the same time they deal multiple preference conditions simultaneously and have robust performance for problems with large number of objectives. A specific form of fitness function is proposed for performing a small number of genetic iterations if DM considers that this may help to find better compromise solution. At the final phase of the search process we combine our approach with the reference point approach suggested by Wierzbicki [14].

II. Proposed Reference Point Strategies

The main idea of both proposed strategies is for given reference point provided by the DM to construct direction in the variable space, which is used to move quickly the whole population of solutions to the desired part of Pareto-optimal front – as close as possible to a given reference point.

A. Worst point reflection strategy (WPRS)

Let the reference point *fr* is known and let we have a population *P* of *p* solutions. Then perform the following procedure:

Step 1. Calculate the Euclidean distance measure in the criteria space between each point $fi \in P$, for $i = 1,\dots,p$; and the reference point *fr*:

$$di = \sqrt{\sum_{j=1}^{k} \left(fi_j - fr_j \right)^2} \qquad \textbf{(5)}$$

Step 2. Determine the solution $\boldsymbol{x}_{\max}$, corresponding to the point $fi_{\max}$, having maximal value $di_{\max}$. Set *icount* = 1. Set *itlim* = *const*. Here $const \le n+1$;

Step 3. Calculate the weight center **c** of all other solutions $\boldsymbol{xi} \in P$, for $i = 1,\dots,p-1$; and $i \neq \max$:

$$\boldsymbol{c} = \frac{\sum_{i=1}^{p-1} x_i}{p-1} \qquad \textbf{(6)}$$

Step 4. Construct a direction in the variable space for moving the whole population to the reference point using reflection of solution $\boldsymbol{x}_{\max}$ towards the weight center *c*:

$$\boldsymbol{y} = \boldsymbol{c} - \boldsymbol{x}_{\max} \qquad \textbf{(7)}$$

Step 5. Move each solution $xi \in P$, for $i = 1,\dots,p$; in the following manner:

1.) $\boldsymbol{xi}_{\text{new}} = \boldsymbol{xi} + \boldsymbol{y}$. In case the obtained $\boldsymbol{xi}_{\text{new}}$ violates some constraint from the system **(2-4)** change ***y*** by: $\boldsymbol{y} = 0.9\boldsymbol{y}$ and repeat 1.)

2.) Round off the obtained solution $\boldsymbol{xi}_{\text{new}}$ to the nearest integer feasible solution and include it in the new population P_{new}.

At this step some newly generated solutions may coincide. In this case the number of population members is reduced by 1.

Step 6. Set *icount* = *icount* + 1 and $P = P_{\text{new}}$. If *icount* > *itlim*, then STOP (the procedure terminates), else go to Step 3.

B. Improving direction strategy (IDS)

Let the reference point *fr* is known and let we have a population *P* of *p* solutions. Then perform the following procedure:

Step 1. Calculate the Euclidean distance measure in the criteria space between each point $fi \in P$, for $i = 1,\dots,p$; and the reference point *fr* according **(5)**. Arrange all solutions $xi \in P$, for $i = 1,\dots,p$; in a list *D* according their *di* distance measures in an ascendant order.

Step 2. Calculate the weight center of the first three solutions $\boldsymbol{xi} \in D$, for $i = 1, 2, 3$;:

$$\boldsymbol{cd} = \frac{\sum_{i \in D, i=1}^{3} x_i}{3} \qquad \textbf{(8)}$$

Calculate the weight center ***c*** of all other solutions $\boldsymbol{xi} \in D$, for $i = 4,\dots,p$;

Step 3. Construct an improving direction in the variable space for moving the whole population to the reference point as follows:

$$\boldsymbol{y} = \boldsymbol{cd} - \boldsymbol{c} \qquad \textbf{(9)}$$

Step 4. Move each solution $xi \in P$, for $i = 1,\dots,p$; in the following manner:

1.) $\boldsymbol{xi}_{\text{new}} = \boldsymbol{xi} + \alpha\boldsymbol{y}$, where α is a positive scalar. In case the obtained $\boldsymbol{xi}_{\text{new}}$ violates some constraint from the system **(2-4)** change α by: α =0.9α and repeat 1.)

2.) In case the corresponding di-value is decreased repeat 1.), otherwise go to 3.)

3.) Round off the obtained solution $\boldsymbol{xi}_{\text{new}}$ to the nearest integer feasible solution and include it in the new population P_{new}.

Step 5. Find all different non-dominated solutions $\boldsymbol{xi} \in P_{\text{new}}$ and arrange them in an ascending order of their Euclidean distances to the reference point fr. Then include them in the final population P_{fin}. STOP.

III. New evolutionary MOCIP-algorithm

The considered problem has a closed feasible domain because there are given lower and upper bounds for each variable (see constraint **(3)**). The Tchebycheff center of the feasible domain can be calculated and can be rounded off to the closest integer feasible point $\boldsymbol{x^{ch}}$, called below rounded Tchebycheff center.

The DM can give feasible or infeasible reference point(s). In both cases the proposed new algorithm finds a set of (near) Pareto-optimal solutions, which are located close to the supplied reference point(s). The found non-dominated points and their corresponding solutions during the search process are stored in an array *LND*, which is updated at each generation of new point.

The proposed MOCIP-algorithm contains the following steps:

Step1. Calculate the points f^*_1, f^*_2, …, f^*_k; and their corresponding solutions x^*i, i=1,…,k; , where f^*_i is the optimal point for the i-th objective. To obtain these points solve k single objective problems with the same constraint system **(2-4).**

Step 2. Create an initial population Po of size p, generating p uniformly distributed integer feasible solutions around the rounded Tchebycheff center $\boldsymbol{x^{ch}}$. Here p could vary according the number of objectives k in the current problem. For example: p=2k, when k≥50, p=3k for $k \in [33 \div 50)$, p=4k for $k \in [25 \div 33)$, p=5k for $k \in [10 \div 25)$, etc.

Step 3. Ask the DM to specify a reference point fr.

Step 4. Perform *WPRS*.

Step 5. Ask the DM to specify the most acceptable, as well the most non acceptable obtained point (corresponding xb and xw) Based on the direction $v = xb - xw$ propose new reference point to DM, or ask her/him to specify a new reference point fr.

Step 6. Perform *IDS*. Include all obtained non-dominated solutions in the population P_{fin}.

Step 7. Show the obtained non-dominated points, corresponding to the non-dominated solutions $xi \in P_{fin}$, for i = 1,…,p_{fin}; to the DM.

Step 8. Ask the DM if she/he wishes to perform small number genetic iterations using P_{fin}. If yes calculate for each point from P_{fin} utility coefficients $\eta_i = \dfrac{fi(x) - fi^*}{fi_{\max} - fi^*}$, i=1,…k; where $fi_{\max}$ is the obtained maximal value of i-th objective during the search process. Calculate fitness values for the solutions in P_{fin}, and for the new generated offspring solutions, minimizing the following fitness function:

$$\boldsymbol{F}_i = \eta_i.di.(nd_i+1) \qquad \textbf{(11)}$$

where di are the Euclidean distance measures in the criteria space between the point $fi \in P_{fin}$, for $i = 1,\dots,p_{fin}$; and the reference point $\boldsymbol{fr}$. The value nd_i corresponds to the number of points, which eventually dominate the point fi.

Step 9. Show the solutions and their corresponding points in the population P_{fin} to the DM. Ask DM to enter a last reference point $\boldsymbol{fr}$, close located to the non-dominated solution $\bar{x} \in P_{fin}$.

Step 10. Perform the procedure, suggested by Wierzbicki [14], to create k new reference points around $\boldsymbol{fr}$ as follows:

$$\boldsymbol{fj} = \boldsymbol{fr} + (\bar{x} - \boldsymbol{fr}).\,\mathrm{e}^{(j)}, j=1,\dots,k; \qquad \textbf{(12)}$$

where $\mathrm{e}^{(j)}$ is the j-th coordinate direction vector.

Step 11. Perform simple search procedure for a better point along the direction ($\bar{x}$ - $\boldsymbol{fr}$), starting with each created reference point $\boldsymbol{fj}$ j=1,…,k;.

Step 12. Ask the DM to choose final solution among the non-dominated solutions found during the search process, stored in *LND*. STOP.

IV. Illustrative example

The performance of MOCIP-algorithm is illustrated on the following test example:

$$\textbf{Min } \boldsymbol{f}(x) = [\boldsymbol{f1}(x), \boldsymbol{f2}(x), \boldsymbol{f3}(x)]^T,$$

$$\boldsymbol{f1}(x) = x_1^2 + 3x_2^2;\ \boldsymbol{f2}(x) = 5x_1^2 + x_2^2;\ \boldsymbol{f3}(x) = 2x_1^2 - x_2;$$

subject to:

$$- x_1 - x_2 + 11 \le 0$$

$$0 \le x_1 \le 16,$$

$$0 \le x_2 \le 16,$$

$$x_1, x_2 \text{ – integer;}$$

At Step 1. The points f^*_1, f^*_2, f^*_3; are: f^*_1 = (91, 329,125), $x1^*$ = (8,3). f^*_2 = (247,101,-1), $x2^*$=(2,9), f^*_3 = (768,256,-16), $x3^*$ = (0,16).

At Step 2. The rounded Tchebycheff center $\boldsymbol{x^{ch}}$ = (10,10). The following initial population is created: $x^{(1)}$ = (5,12), $x^{(2)}$ = (8,12), $x^{(3)}$ = (11,12), $x^{(4)}$ = (14,12), $x^{(5)}$ = (7,10), $x^{(6)}$ = (10,10), $x^{(7)}$ = (13,10), $x^{(8)}$ = (5,8), $x^{(9)}$ = (8,8), $x^{(10)}$ = (11,8), $x^{(11)}$ = (14,8);

At Step 3. The DM specifies the reference point $\boldsymbol{fr}$ = (100, 150, 45).

At Step 4. The worst point in the population is $x^{(4)}$ = (14,12), $f^{(4)}$ = (628,1124,380) with Euclidean distance measure d4=1157.4476. The weight center **c** = (9.2, 9.8). The direction for moving the whole population is **y** = (-4.8, -2.2). The obtained new population is: $x^{(1)}$ = (1,10), $x^{(2)}$ = (3,10), $x^{(3)}$ = (6,10), $x^{(4)}$ = (9,10), $x^{(5)}$ = (3,8), $x^{(6)}$ = (5,8), $x^{(7)}$ = (8,8), $x^{(8)}$ = (4,7), $x^{(9)}$ = (5,6), $x^{(10)}$ = (6,6), $x^{(11)}$ = (9,6); The worst point in this population is $x^{(4)}$ = (9,10), $f^{(4)}$ = (381,505,152) with

Euclidean distance measure $d4$=465.2258. The weight center **c** = (5, 7.9). The direction for moving the whole population is **y** = (-4, -2.1). The obtained new population is: $x^{(1)}$ = (1,10), $x^{(2)}$ = (2,9), $x^{(3)}$ = (3,8), $x^{(4)}$ = (5,8), $x^{(5)}$ = (4,7), $x^{(6)}$ = (5,6), $x^{(7)}$ = (6,5);

At Step 5. The DM specifies as most acceptable the solution $x^{(7)}$ = (6,5) with $f^{(5)}$ = (111,205,67); The most non acceptable point is $x^{(1)}$ = (1,10) with $f^{(1)}$ = (301,105,-8); The improvement direction between them is y = (5, -5). The new proposed reference point is ***fr*** = (100, 500, 200) with corresponding solution x = (10,0),. The weight center ***cd*** = (5,6). The weight center ***c*** = (2.75, 8.75). The direction for moving the whole population is **y** = (2.25, -2.75). The parameter α =1. The obtained new population is: $x^{(1)}$ = (3,8), $x^{(2)}$ = (4,7), $x^{(3)}$ = (5,6), $x^{(4)}$ = (7,6), $x^{(5)}$ = (6,5), $x^{(6)}$ = (7,4), $x^{(7)}$ = (8,3); During the repeated step in the same direction also the non-dominated solutions $x^{(8)}$ = (9,2), $x^{(9)}$ = (10,1) have been generated.

At Step 7. The obtained non-dominated solutions are: $x^{(1)}$ = (1,10), $x^{(2)}$ = (2,9), $x^{(3)}$ = (3,8), $x^{(4)}$ = (4,7), $x^{(5)}$ = (5,6), $x^{(6)}$ = (6,5), $x^{(7)}$ = (7,4), $x^{(8)}$ = (8,3), $x^{(9)}$ = (9,2), $x^{(10)}$ = (10,1);

At Step 8. the DM does not want to perform a genetic search procedure.

At Step 9. the DM refuses to enter a last reference point ***fr***.

At Step 10. and Step 11. don’t have been generated any better points.

At Step 12. the DM chooses the solution $x^{(7)}$ = (7,4) with $f^{(7)}$ = (97,261,94) as the best compromise solution.

In case DM wishes, she/he can start the algorithm specifying another reference point. In this way DM could explore different parts of Pareto-optimal front.

V. Conclusion

The proposed MOCIP-algorithm has the following basic characteristics:

- It is designed to find a preferred set of solutions instead of the entire Pareto-optimal set.
- It can quickly converge to the desired part of Pareto-optimal front.
- It is indifferent to the shape of Pareto-optimal front.
- It is applicable to problems with large number of objectives and large number of variables.
- It does not put great demands to the DM.
- It is an interactive evolutionary method and could generate a number of solutions in the region of interest, so that the DM would be able to find without great efforts the satisfactory non-dominated solution among them.

Acknowledgement

The authors gratefully acknowledge the support of Bulgarian National Science Fund, Grant No DTK02/71 “Web-Based Interactive System, Supporting the Building Models and Solving Optimization and Decision Making Problems” and of European Social Fund and Bulgarian Ministry of Education, Youth and Science under Operative Program “Human Resources Development”, Grant BG051PO001-3.3.04/40.

References

[1] Branke J., Deb K., Miettinen K., Słowiński R., „*Multiobjective Optimization. Interactive and Evolutionary Approaches*“, Springer-Verlag Berlin Heidelberg, 2008

[2] Coello C. A. C., D. A. Van Veldhuizen, and G. Lamont *Evolutionary Algorithms for Solving Multi-Objective Problems*, Boston, MA: Kluwer Academic Publishers, 2002.

[3] Deb K., „*Multi-Objective Optimization Using Evolutionary Algorithms*”, Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester, 2001

[4] Deb K., J. Sundar, U. B. Rao N., S. Chaudhuri, “Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms”, *International Journal of Computational Intelligence Research*, Vol. 2, No. 3 (2006), pp. 273-286.

[5] Garey M. R. and Johnson D. S. *Computers Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco 1979.

[6] Goldberg, D. E. Genetic Algorithms in Search, Optimization and Machine Learning, Reading Mass, Addison Wesley, 1989.

[7] Guliashki V., (2008) „An Accelerating Technique for Population Based Algorithms”, Proceedings of Papers of IX Internatrional Conference on Computer Systems and Technologies CompSysTech’08, Gabrovo, Bulgaria, 12-13 June, 2008, pp. -IIIB.11-1- ÷ -IIIB.11-6-.

[8] Guliashki V., L. Kirilov, “SPEA-Based Method for MCDM Convex Integer Problems”, *Cybernetics and Information Technologies*, Vol. 9, No. 4 (2009), pp. 93-101.

[9] Ivanova J, “An Interactive Hybrid Algorithm of Integer Multiobjective Optimization”, International Conference Automatics and Informatics’08, In: Proceedings “School for young Scientists Business Process management”, (Ed. P. Ruskov), 2008, pp. 45-48.

[10] Miettinen K., P. Neittaanmäki, M. M. Mäkelä, J. Périaux, “*Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming and Industrial Applications*”, John Wiley & Sons, Chichester, 1999

[11] Nemhauser G.L. and Wolsey L.A. *Integer and Combinatorial Optimization*, Wiley, New York 1988.

[12] Pareto V. “*Cours d’Economie Politique*”, Rouge, Lausanne, Schwitzerland, 1896.

[13] Praditwong, K.; Xin Yao, „How well do multi-objective evolutionary algorithms scale to large problems”, *Evolutionary Computation*, 2007. CEC 2007. *IEEE Congress on*, 25-28 Sept. 2007, pp. 3959 – 3966.

[14] Wierzbicki A. P., “The use of reference objectives in multiobjective optimization”. In G. Fandel and T. Gal, editors, Multiple Criteria Decision Making Theory and Applications, Berlin: Springer-Verlag, pp. 468-486.