An Evolutionary Approach for Multiobjective Optimization using Adaptive Representation of Solutions

Crina Groşan

Department of Computer Science
Babeş-Bolyai University, Kogălniceanu 1
Cluj-Napoca, 3400, Romania.
`cgrosan@cs.ubbcluj.ro`

Abstract. Many algorithms for multiobjective optimization have been proposed in the last years. In the recent past a great importance have the MOEAs able to solve problems with more than two objectives and with a large number of decision vectors (space dimensions). The difficulties occur when problems with more than three objectives (higher dimensional problems) are considered. In this paper, a new algorithm for multiobjective optimization called Multiobjective Adaptive Representation Evolutionary Algorithm (MAREA) is proposed. MAREA combines an evolution strategy and an steady-state algorithm. The performance of the MAREA algorithm is assessed by using several well-known test functions having more than two objectives. MAREA is compared with the best present day algorithms: SPEA2, PESA and NSGA II. Results show that MAREA has a very good convergence.

1 Introduction

In the recent years a number of Multiobjective Optimization Evolutionary Algorithms (MOEAs) have been proposed. The interest is now focused on finding the Pareto front for functions having more than two objectives.

A new algorithm called Multiobjective Adaptive Representation Evolutionary Algorithm (MAREA) is proposed in this paper. This algorithm uses a solution representation similar to the representation used by Adaptive Representation Evolutionary Algorithm (AREA) introduced in [5]. Used operators are the same as those used by AREA.

For assessing the performances of the MAREA algorithm a comparison with some recent algorithms for Multiobjective Optimization is performed. The algorithms used in this comparison are: SPEA2 introduced by Zitzler and al. in [8], NSGA II introduced by Deb in [2] and PESA introduced by Knowles in [1]. Four difficult test functions are considered for comparison purposes. The number of objectives varies between 2 and 8.

The paper is structured as follows: Section 2 describes AREA technique: solution representation and the operators used. In Section 3 the newly algorithm proposed – MAREA – is presented. In Section 4 the test function used for comparison are described. Two performance metrics for compare the results obtained by these algorithms are presented in Section 5. Numerical experiments with the algorithms SPEA2, NSGA II and PESA are performed in Section 6 of the paper. A set of conclusions are mark out in Section 7 of the paper.

2 AREA Technique

The main idea of this technique is to allow each solution to be encoded over a different alphabet. Moreover, the representation of a particular solution is not fixed. Representation is adaptive and may be changed during the search process as effect of the mutation operator. AREA relies mainly on Dynamic Representation (DR) proposed in [7].

2.1 Solution representation

Within AREA each solution has its own encoding alphabet and can be represented as a pair(x, B) where B is an integer number, $B \geq 2$ and x is a string of symbols from the alphabet $\{0, 1, \ldots, B\text{-}1\}$. If $B = 2$, the standard binary encoding is obtained. The alphabet over which x is encoded may change during the search process by applying an operator similar with mutation operator and called *transmutation*. If no ambiguity arises we will use B to denote the alphabet $B = \{0, 1, \ldots B\text{-}1\}$.

2.2 Mutation

Each gene is affected by mutation with a fixed mutation probability. The mutated gene is a symbol randomly chosen from the same alphabet.

Consider the chromosome C represented over the alphabet $B = 8$: $C = (631751, 8)$. Consider a mutation occurs on the 3^{rd} position of x. Let be 4 the value of mutated gene. Then the mutated chromosome is: $C_1 = (634751, 8)$.

2.3 Transmutation

The transmutation operator can modify only the value of the alphabet over that the chromosome is represented. The new value of the alphabet is a randomly chosen integer value.

When position giving the alphabet B is changed, then the object variables will be represented using symbols over the new alphabet, corresponding to the mutated value of B.

Consider a transmutation occurs on the last position of chromosome C and the mutated value is $B_2 = 10$. Then the mutated chromosome is:

$C_2 = (209897, 10)$.

C and C_2 encode the same value over two different alphabets ($B = 8$, $B_2 = 10$).

Remark A mutation generating an offspring worse than its parent is called a *harmful mutation*.

2.4 Offspring acceptance

If the offspring obtained by mutation is better than its parent than the parent is removed from the population and the offspring enters the new population. Otherwise, a new mutation of the parent is considered.

A mutation generating an offspring worse than its parent is called a harmful mutation. If the number of successive harmful mutations exceeds a prescribed threshold (*Max_Harmful_Mutations*) the individual representation is changed and with this new representation it enters in the new population.

3 MAREA Algorithm

Two stages are distinguished to determinate the real Pareto front. At the first stage the effort is focused on finding the Pareto front. At the second stage the effort is focused on spreading the individuals along the Pareto front.

At stage I, MAREA uses a population containing few individuals (usually one individual). These individuals are mutated until no improvement of solution's quality occurs for a specified number of iterations. That means that this solution is near Pareto front (the real Pareto front or a local Pareto front). Stage II begins at this moment. At this stage the spread of solution on the Pareto front is realized. The aim of this stage is to obtain a good distribution along Pareto front. An enlarged population is permitted at this stage in order to obtain a good coverage of Pareto front. A detailed description of the MAREA's stages is given bellow.

3.1 Stage I – Convergence to the Pareto front

MAREA uses a single population of individuals that are evolved using two variations operators: mutation and transmutation. The evolutionary model used here is similar to (1+1) ES.

At this stage the population has only one individual. Initial population (initial individual) is randomly generated. Mutation is the unique variation operator. The offspring and parent are compared. Dominance relation guides the survival.

If the offspring dominates the parent then the offspring replaces the parent. If the parent dominates the offspring obtained for Max_Harmful_Mutations successive mutations then another alphabet is chosen and the parent is represented in symbols over this alphabet.

A parameter called *Max_Steps_To_Front* is used on order to find when this stage is over. If no improvement occurs for *Max_Steps_To_Front* iterations it is likely that this solution is near Pareto (global or very good local) front. As the search process gets near Pareto front the first stage stops here and at this moment the second stage starts.

3.2 Stage II – Dispersion on the Pareto front.

After first stage the population contains one individual only. This individual is near Pareto (local or global) front. The evolutionary model used here is similar with Steady-State.

During this stage the population size is variable and can increase and decrease also. Mutation operator is applied to each solution from population. *Max_Harmful_Mutation* parameter is used to determinate when the alphabet representation for a solution is changed. At each step a randomly chosen individual is mutated. The offspring is added to the population. All dominated solutions are removed. If all solutions are nondominated and the population size exceeds a prescribed treshold then some solutions are removed. For computing which solutions will be removed from the population the following mechanism is used: the nearest solutions are chosen from population. One of them is removed.

The size of final population can be smaller than maximum size allowed. This situation is likely if the Pareto front consist of a very small number of points.

4 Test Functions

Test functions used in these experiments were introduced in [3]. Each of them is described in the Table 1. These test functions are M – objective problems.

5 Performance metrics

In this study a convergence metric proposed by Deb and Jain in [4] are used to determinate the algorithms performances.

This convergence metric represents the distance between the set of converged non-dominated solutions and the global PO front.

Let P^* be the reference or target set of points on the PO front and let F be the final non-dominated set obtained by a multiobjective evolutionary algorithm.

Then from each point i in F the smallest normalized Euclidian distance to P^* will be:

$$d_i = \min_{j=1}^{|P*|} \sqrt{\sum_{k=1}^{M} \left(\frac{f_k(i) - f_k(j)}{f_k^{\max} - f_k^{\min}} \right)^2}$$

where $f_k^{\max}$ and $f_k^{\min}$ are the maximum and minimum function values of k-th objective function in P^* and M represent the number of objective functions.

Remark

Lower values of convergence metric represent good convergence ability.

Table 1. Test functions.

Test function		Domain	Global minimum
DTLZ1	$f_1(\mathrm{x}) = \frac{1}{2}x_1x_2\dots x_{M-1}(1+g(\mathrm{x}_M))$, $f_2(\mathrm{x}) = \frac{1}{2}x_1x_2\dots(1-x_{M-1})(1+g(\mathrm{x}_M))$, $\vdots$ $f_{M-1}(\mathrm{x}) = \frac{1}{2}x_1(1-x_2)(1+g(\mathrm{x}_M))$, $f_M(\mathrm{x}) = \frac{1}{2}(1-x_1)(1+g(\mathrm{x}_M))$, *where* $g(\mathrm{x}_M) = 100\left(\lvert\mathrm{x}_M\rvert + \sum_{x_i\in\mathrm{x}_{\mathrm{M}}(x_i-0.5)^2-\cos(20\pi(x_i-0.5))}\right).$	$0 \leq \quad x_i \quad \leq 1$ for i = 1,2,…,n	$x_i{}^* = 0.5$ $(x_i{}^* \in \mathrm{x}_M)$ $\sum_{m=1}^{M} f_m^* = 0.5$
DTLZ2	$f_1(\mathrm{x}) = (1+g(\mathrm{x}_M))\cos(x_1\pi/2)\dots\cos(x_{M-1}\pi/2)$, $f_1(\mathrm{x}) = (1+g(\mathrm{x}_M))\cos(x_1\pi/2)\dots\sin(x_{M-1}\pi/2)$, $f_M(\mathrm{x}) = (1+g(\mathrm{x}_M))\sin(x_1\pi/2)$, *where* $g(\mathrm{x}_M) = \sum_{x_i\in\mathrm{x}_M}(x_i-0.5)^2$.	$0 \leq \quad x_i \quad \leq 1$ for i = 1,2,…,n	$x_i{}^* = 0.5$ $(x_i{}^* \in \mathrm{x}_M)$ $\sum_{m=1}^{M} (f_m^*)^2 = 1$
DTLZ3	The test functions are the same that test functions used in DTLZ2 and g function is one used in DTLZ1.	$0 \leq \quad x_i \quad \leq 1$ for i = 1,2,…,n	$x_i{}^* = 0.5$ $(x_i{}^* \in \mathrm{x}_M)$ $\sum_{m=1}^{M} (f_m^*)^2 = 1$

6 Numerical experiments

In order to mark out MAREA performances some numerical experiment are performed. Three algorithms are used for comparisons: SPEA2, NSGA II and PESA.

To make a comparison fair the same number of function evaluations is used by all considered algorithms.

The parameters used by each of the considered algorithms are presented in the Table 2.

The values of parameters used by PESA, SPEA2 and NSGA II have taken from [6].

Table 2. The values of parameters used by MAREA, SPEA2, NSGA II and PESA. The parameters for SPEA2, NSGA II and PESA are taken from [6].

Parameter	**MAREA**	**PESA**	**SPEA2**	**NSGA II**
Crossover probability	-	0.8	0.7	0.7
Distribution Index (DI) for SBX	-	15	15	15
Mutation Probability (if n = number of variables)	$1/n$	$1/n$	$1/n$	$1/n$
DI for polynomial mutation	-	15	15	20
Ratio of internal population size to archive size	-	1:1	1:1	1:1
Number of grids per dimension (PESA)	-	10	-	-
MAX_HARMFUL_ MUTATIONS (MAREA)	3000	-	-	-

The values corresponding to population size for each considered objectives numbers are presented in Table 3.

The number of generations and the number of functions evaluations corresponding to objectives number for each test functions are presented in Tables 4 and 5.

The results obtained by applying convergence metric are presented in Table 6. Results are averaged over 30 runs. Resutls obtained by NSGA II, SPEA2 and PESA are taken from [6].

We can see from this table MAREA outperforms the considered algorithms for comparison in almost all situations. PESA is better than NSGA II and SPEA 2. MAREA strongly outperforms NSGA II and SPEA 2 for all considered test functions. For a small number of objectives PESA and MAREA give closed

Table 3. Population size corresponding to objectives number. The values are taken from [6].

Number of objectives	Population size
2	20
3	50
4	100
6	250
8	400

Table 4. Number of generations used in experiments for each test functions. The values are taken from [6].

Test function	**Number of generations**	
	For 2, 3 and 4 objectives	**For 6 and 8 objectives**
DTLZ1	300	600
DTLZ2	300	600
DTLZ3	500	1,000

Table 5. Number of functions evaluations used for all considered algorithms. The values are taken from [6].

Test function	**Number of functions evaluations**				
	2 obj	**3 obj**	**4 obj**	**6 obj**	**8 obj**
DTLZ1	6,000	15,000	30,000	150,000	360,000
DTLZ2	6,000	15,000	30,000	150,000	360,000
DTLZ3	10,000	25,000	50,000	250,000	600,000

Table 6. Results obtained by applying convergence metric. Results are averaged over 30 runs.

Test function		**Convergence metric**			
		MAREA	**PESA**	**SPEA2**	**NSGA II**
DTLZ1	2 objectives	0.57292	2.8694	3.088	2.2766
	3 objectives	0.04299	0.0441	0.0484	0.3836
	4 objectives	0.01546	0.0231	0.2992	3.1028
	6 objectives	0.001604	0.0017	5.999	120.191
	8 objectives	0.00391	0.004	498.27	465.301
DTLZ2	2 objectives	0.000091	0.00008	0.00026	0.00180
	3 objectives	0.00014	0.00035	0.00663	0.010003
	4 objectives	0.001304	0.00170	0.03369	0.04529
	6 objectives	0.00195	0.00301	2.00216	1.67564
	8 objectives	0.008084	0.00689	2.35258	2.30766
DTLZ3	2 objectives	0.12605	0.79397	0.77622	0.6369
	3 objectives	0.1591	0.20528	0.29271	0.2451
	4 objectives	0.0293	3.60430	5.07137	6.3261
	6 objectives	0.1822	5.30454	10.5368	9.4875
	8 objectives	0.1734	6.32247	10.6293	10.273

results. But for a greater number of objectives MAREA is able to converge to the global Pareto front while PESA not in all situations.

7 Conclusions and further work

In this paper a new evolutionary algorithm for multiobjective optimization was proposed. This algorithm uses solution representation introduced by AREA technique. The particularity of this algorithm is his structure. Initially only one individual is considered. The algorithm tries to situate this individual near Pareto front. Mutation and transmutation operators are used in order to improve the individual. If no improvement appear after a specified number of mutations means that this solution in near Pareto front. (Sometimes this can be a local Pareto front). When this unique solution is near Pareto front the population can be extended by considering the others nondominates solutions. These solutions are obtained by applying mutation and transmutation operator also over these new solutions considered. After a specified number of iteration the last population obtained is considered the final population.

The two parameters for determining when the alphabet over which a solution is represented is changed and the other parameter determining when the first stage is finish (i.e. determining when population size can increase) are very important. By considering different values of these parameters the final result is different.

An adaptation of these parameters for each problem is the work of the other paper.

References

1. Corne, D., Knowles, J., Oates, M. The Pareto-Envelope based Selection Algorithm for Multiobjective Optimization. In Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature, Springer-Verlag, Berlin, (2000) 839-848.
2. K. Deb, S. Agrawal, A Pratap and T. Meyarivan, A fast elitist non – dominated sorting genetic algorithm for multi-objective optimization: NSGA II. In M. S. et al. (Ed), *Parallel Problem Solving From Nature – PPSN VI*, Springer-Verlag, Berlin (2000) 849-858.
3. K. Deb, L. Thiele, M. Laumanns and E. Zitzler. Scalable Multi-Objective Optimization Test Problems. Proceeding of IEEE Congress on Evolutionary Computation, Hawaii, (2002).
4. Deb, S. Jain, Running performance metrics for evolutionary multi-objective optimization, KanGAL Report 2002004, Indian Institute of Technology, Kanpur, India, (2002).
5. Grosan, C., Oltean, M. Adaptive Representation Evolutionary Algorithm - a new technique for single objective optimization. In Proceedings of First Balcanic Conference on Informatics (BCI), Thessaloniki, Greece, (2003) 345-355.
6. V. Khare, X. Yao, K. Deb. Performance Scaling on Multi-objective Evolutionary Algorithms. Technical Report 2002009, Kanpur Genetic Algorithm Laboratory (KanGAL), Indian Institute of Technology Kanpur, India (2002).
7. 12.Kingdon J, Dekker L. The Shape of Space, Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA '95) IEE, London, (1995) 543-548.
8. Zitzler, E., Marco Laumanns and Thiele, L., SPEA 2: Improving the Strength Pareto Evolutionary Algorithm, TIK Report 103, Computer Engineering and Networks Laboratory (TIK), Departament of Electrical Engineering Swiss federal Institute of Technology (ETH) Zurich, (2001).