SOLVING MULTIOBJECTIVE OPTIMIZATION PROBLEMS USING A NON-PARETO BASED FITNESS ASSIGNMENT METHOD

Corina ROTAR[1), Laszlo Barna IANTOVICS[2)
1)"1 Decembrie 1918" Univ. of Alba Iulia, Romania, 2)Petru Maior University, Tg. Mures, Romania
crotar@uab.ro

Abstract: Evolutionary algorithms for multi-criteria optimization fall into two major categories: Pareto and non-Pareto based techniques. Pareto techniques explicitly make use of the Pareto optimality concept in the assessment of individuals' quality. In this regard, the extracting of the non-dominated solutions is a costly procedure in terms of time, but with remarkable results. Non-Pareto techniques avoid the drawback of the Pareto ranking techniques but having weaker performance. We believe that the non-Pareto fitness assignment in evolutionary multi-criteria optimization may be advanced. Therefore, the paper proposes a new procedure for fitness evaluation that accompanies a simple evolutionary algorithm for solving multiobjective optimization. The proposed evaluation procedure determines the potential of each solution to be non-dominated and does not identify the non-dominated solutions, thus, the algorithmic efficiency in terms of time becomes superior to the Pareto-ranking based algorithms. The performance of a solution is given by the degree it may contribute to the improvement of the population and does not rely on its Pareto rank. Comparative analysis shows that the proposed evolutionary algorithm has lower complexity and provides better results than the state of the art algorithm.
Keywords: *Evolutionary Computation, multiobjective optimization, Fitness assignment.*

I. INTRODUCTION

Solving multi-criteria optimization from the perspective of evolutionary algorithms is one of the most prolific research directions. In recent decades, many evolutionary methods have been developed and successfully solved various multi-objective optimization problems. As a proof of increased interest in the topic stands the huge web repository for evolutionary multiobjective optimization [21]. From the beginning, the multi-criteria optimization evolutionary algorithms challenged the problem of evaluating possible solutions in multiobjective context. As for the single-criterion optimization, fitness function may be built on the sole objective's base, whereas for the multi-criteria optimization, researchers struggled to establish a proper strategy for assessing the solutions in relation with the multiple objectives. Among the first well-known evolutionary techniques that have gained popularity and remained as reference works, we firstly remark Vector Evaluate Genetic Algorithm[13] which implies that appropriate portions of the population, or subpopulations, are selected from the current generation according to each of the objectives, separately. Also, several evolutionary techniques were developed on the basis of scalarizing [15], aggregation techniques [1],[1],[2], lexicographic method [3], [15], constraint approach. Non-Pareto based EA recently include algorithms as decomposition based: MOEA/D [17] and indicator based evolutionary algorithms: IBEA [18]. MOEA/D is an aggregation-based evolutionary algorithm based on the idea of decomposing the original multi-objective problem into several single-objective sub-problems by means of well-defined scalarizing functions. IBEA can be adapted to the user's preferences and does not require any additional diversity preservation.

Goldberg's proposal to incorporate the concept of Pareto optimality in the evaluation of candidate solutions represent a turning point in researching the subject. Currently, most evolutionary multi-objective optimization (EMO) algorithms apply Pareto-based ranking schemes. MOGA [19], NPGA[20], NSGA[8], SPEA [9], NSGA-II[10], SPEA2 [11], PAES [12]. In a comprehensive review of the state-of-art evolutionary techniques for multiobjective optimization [14], it is stated that Pareto-based approaches are the most popular.

Regarding the Pareto evaluation mechanism, there are several issues that should be mentioned. Firstly, Pareto ranking procedure, or those procedures which decide if the solutions of the current population are Pareto non-dominated, are costly in terms of the consumed time. For an optimization problem with m objectives, using a population of n individuals, popular algorithm, NSGA, is reported as an efficient technique for MOP but it has O(m·n3) complexity for Pareto ranking procedure [[8]]. The improved NSGA-II is a faster procedure which alleviates the main criticism regarding the computational complexity, having a reduced complexity to $O(m \cdot n^2)$. The authors note that although the time complexity has reduced to $O(m \cdot n^2)$, the storage requirement has increased to $O(n^2)$. Therefore, a less time complexity is attained with the cost of an increased space. Later, Jensen [7] proposed a faster variant of non-dominated ranking in time $O(n \cdot log(m-1) \cdot n)$. Another popular algorithm SPEA is later improved by its successor SPEA2 by focusing on the fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method. The improvement does change the overall complexity of the algorithms from $O(n^3)$ to $O(n^2 \cdot logn)$. Many papers discuss the complexity issue of the MOEA's and developing the faster MOEAs is one of the desiderata. In [4], [6],

Manuscript received July 31, 2018; revised October 30, 2018

comprehensive analysis is given, suggesting the researchers' effort to reduce the computational complexity. The popularity of Pareto-based evolutionary algorithms is attained rather by their performances, but their straightforwardness: for instance, comparing them with the uncomplicated aggregation techniques, it is obvious that they are more sophisticated and include auxiliary procedures to stress some feature (selection mechanism, diversity preservation, crowding measurement, and so on) and to advance the overall performance. The present paper propose a plain non-Pareto based evolutionary algorithm, which is comparable, regarding the efficiency and performance with the Pareto-ranking based algorithms. The rest of the paper is organized as follows: section 2 describes the proposed techniques, section 3 reports experimental results and, finally, section 4 discusses the main results and suggests further research directions.

II. DESCRIPTION OF THE PROPOSED ALGORITHM

Fitness assignment procedure: Deciding whether a solution of the current population is Pareto non-dominated is costly, as it requires direct comparisons with the other solutions from the population. Therefore, fitness assignment based on the Pareto ranking is an expensive procedure. An alternative to this expensive procedure would be the straightforward aggregation techniques, where the weighted sum of the objectives' values gives the performance of the solution. When using an aggregation technique for fitness assignment, another issue arises: how to properly choose the weights. Thus, the following algorithm for fitness assignment comprises the straightforwardness of an aggregation technique and the efficiency of a Pareto-ranking technique. *Notation*:

-pop– population of n solutions: pop(i), $i=\{1,\dots n\}$

-b (m, n) - matrix of scores (integers), where m- number of objectives, n- population size.

For the i^{th} solution of the current population, pop(i), the score b(k,i) represents the number of other individuals pop(j), $j \neq i$ that are strictly weaker than pop(i) regarding the k^{th} objective.

$$b(k,i) = \mathbf{count}(pop(j), j \neq i \ \mathbf{if}\ pop(i).f(k) < pop(j).f(k))$$

Each solutions pop(i) has m scores b(.,i), corresponding to the m objectives involved. The higher the scores are, the more qualified solutions are. For instance, if pop(i) is a unique Pareto non-dominated solution in a population of size n, it would gain maximum scores for each objective, respectively b(k,i)=(n-1), for $k=\{1,\dots,m\}$.

The Figure 1 depicts the evaluation procedure's results for a bi-objective minimization problems (m=2). We consider 8 solutions, plotted in the objective space (left). The obtained scores are shown as data labels (right).

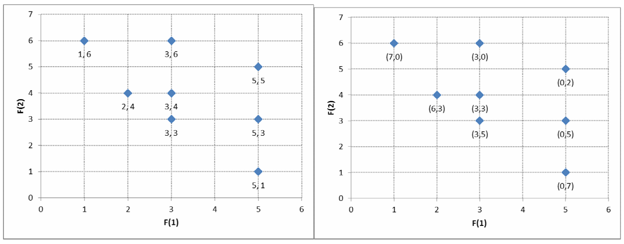

Figure 1. Left: solutions plotted in objectives' space. Right: computed scores for the solutions.

Solutions (1,6) and (5,1) have the scores (7,0) and (0,7), representing that both minimize only one objective. Solutions (2,4) and (3,3) having the scores (6,3) and (3,5) should be the favorite compromise solutions, as they gain, overall, high scores in both objectives. The weakest solutions are (5,5) and (3,6) having the scores (0,2) and (3,0); the lowest scores suggest that both solutions are defeated by many other candidates of the current population. The overall score of a candidate solution is computed as the aggregation of scores for all objectives. Therefore, for the given scenario, the performances of the solutions are:

TABLE 1 Scenario 1: Overall score corresponding to objectives

Solution	Objectives (F(1),F(2))	Scores (b(1,.),b(2,.))	Overall score (performance)
pop(1)	(1,6)	(7,0)	**7**
pop(2)	(2,4)	(6,3)	**9**
pop(3)	(3,3)	(3,5)	**8**
pop(4)	(5,1)	(0,7)	**7**
pop(5)	(3,6)	(3,0)	3
pop(6)	(3,4)	(3,3)	6
pop(7)	(5,3)	(0,5)	5
pop(8)	(5,5)	(0,2)	2

Scores of the solutions are integers between 0 and (n-1). We observed that genuine, non-dominated Pareto solutions correspond to those solutions which have overall scores higher or equall to the $m*(n-1)$, where n is the population size and m-the number of objectives. This observation leads us to the simple replacement procedure which will keep the best solutions (according to the overall scores) and will replace the solutions with an overall score below a specific computable value.

Considering the following example:

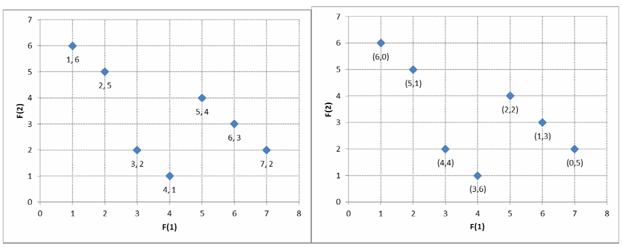

Figure 2. Left: solutions plotted in objectives' space. Right: computed scores for the solutions.

TABLE 2. Scenario 2: Overall score corresponding to objectives

Solution	Objectives (F(1),F(2))	Scores (b(1,.),b(2,.))	Overall score (performance)
pop(1)	(5,4)	(2,2)	4
pop(2)	(6,3)	(1,3)	4
pop(3)	(7,2)	(0,5)	5
pop(4)	(1,6)	(6,0)	**6**
pop(5)	(2,5)	(5,1)	**6**
pop(6)	(4,1)	(3,6)	**9**
pop(7)	(3,2)	(4,4)	**8**

As in the previous scenario, all non-Pareto solutions (*pop(4), pop(5), pop(6), pop(7)*) correspond to an overall score above *(n-1)*.

For bi-objective space, the sum of the scores offers a hint of the solution's quality. The sum of the scores also represents the distance (*p-norm, where p=m-1*) to reference/minimum point (0,0,…0).

Intuitivelly, for multi-objective space, of order *m*, the distance between the i^{th} vector of scores *(b(1,i),b(2,i),…,b(m,i))* to reference point (0,0,…0) is given by the Minkowski distance of order *p=m-1*:

$$MinkowskiDistance(pop(i)) = \left(\sum_{k=1}^{m} b(k,i)^{(m-1)}\right)^{\frac{1}{m-1}}$$

Considering a multiobjective optimization problem with m objectives $f_1,f_2,\dots f_m$, and *n* solutions pop(i), i={1,…n}, the evaluation procedure computes the objectives' value for each individual, then computes the values of the scores b(k,i), k=1,…,m, i=1,…,n and, further, establishes the quality of each individual according to the *MinkowskiDistance* of order *m-1*.

The fitness of an individual is computed as the Minkowski distance of order p=m-1, between the scores and reference point (0,0,…,0) :

$$fitness(pop(i)) = MinkowskiDistance(pop(i))$$

where *m* - number of objectives.

For different values of *m*, Minkowski distance corresponds to: Manhattan Distance (m=2, p=1), Euclidian distance (m=3, p=2).

```
Evaluation procedure – naive algorithm
Initalization b(k,l)=0, where k={1…m} and l={1…n}
Compute objectives' values for each individual pop(i), where i={1…n}
for each obj=1 to m //compute the scores b(k,.)
        for i=1 to n
                for j=i+1 to n
if pop(i).f(obj) < pop(j).f(obj) then
                                b(obj, i) = b(obj, i) + 1
                                end if
                                if pop(i).f(obj) > pop(j).f(obj)
        then
                        b(obj, j) = b(obj, j) + 1
                                end if
                end for
        end for
end for
for each i=1 to n
    pop(i).fitness= MinkowskiDistane b(k,i)), k={1,…m}
end for
End
```

```
Evaluation procedure – optimized algorithm
Initalization b(k,l)=0, where k={1…m} and l={1…n}
Compute objectives' values for each individual pop(i), where i={1…n}
for each obj=1 to m //compute the scores b(k,.)
        Sort population pop by f_obj //time complexity O(n*logn)
        For i=1 to n
        count=number of individuals with the same value for f(obj) in sorted population
                b(obj,i)=n-i-count+1
        end for
end for
for each i=1 to n
    pop(i).fitness= Minkowski Distane (b(k,i)), k={1,…m}
end for
End
```

The Score-Based Fitness Assignment algorithm's time complexity is $O(m*n^2)$, for the naive variant. Optimized algorithm has a complexity of $\boldsymbol{O(m*n*logn)}$. Space complexity is *O(n)*. Comparing with NSGA-II, where the time complexity for ranking the candidates is the same $O(m\cdot n^2)$ but the space complexity is increased to $O(n^2)$, the proposed procedure is less expensive.

Elitism: Starting with the observation that Pareto solutions among the current population get, during evaluation procedure, an overall score higher than a specific value, we propose the replacement mechanism that will keep the qualified solutions and will replace only those underqualified solutions with smaller scores.

Initially we have established a *threshold* which splits the population in two: qualified and non-qualified solutions. The threshold is computed by the following formula:

$$Threshold = \sqrt[m-1]{\sum_{k=1}^{m}\left(\frac{1}{n}\sum_{i=1}^{n} b(k,i)^{(m-1)}\right)}$$

For instance, the threshold's value for 2 objectives represents the sum of the average scores per each objective of the population. Further, the new generation will replace only those solutions from the current generation which have a fitness value below the given *Threshold*. For a population of constant size *n*, the *threshold* is a constant computable parameter of the algorithm.

```
Replacement procedure
for i=1 to n
  if pop(i).fitness<Threshold then
  replace pop(i) with pop_new(i)
  end if
end for
```

The proposed algorithm for multiobjective optimization is a standard genetic algorithm where the fitness is given by the *Minkowsky distance* of order m-1 to origin of the individuals' scores. We used in our implementation: binary encoding with vectors of length *30 x NoOfVariables*, binary tournament selection, uniform crossover, strong mutation with probability *1/(NoOfVariable*30)* and elitism.

```
Score-based multiobjective Genetic Algorithm
Randomly generate initial population pop()of
size n
while (termination_condition*)
        call Evaluation
        Compute Threshold
        for i=1 to n //crossover
        ind1=binarytournamentselect(pop)
        ind2=binarytournamentselect(pop)
        pop_new(i)=    uniformcrossover(ind1,
ind2)
                q=rnd()
                if q<p_mut then //mutation
                        pop_new(i)=
                binarymutation(i)
                end if
        end for
for i=1 to n //elitism
if fitness(pop(i))<Threshold then
pop(i)=pop_new(i)
                end if
        end for
end while
End
```

Remark: the Score-based Multiobjective GA does not include an extra diversity preservation mechanism. The results presented in the following paragraphs show that the score-based fitness assignment correlated with elitist replacement procedure and binary tournament selection maintain the distribution in Pareto fronts, comparable, for bi-objective cases, to algorithms which have extra diversity preservation mechanism.

III. EXPERIMENTS

In order to illustrate the performance of the proposed Score-based Multiobjective Genetic Algorithm, we used several test problems proposed in [22], three metrics: Spacing [15], Coverage [5] and S-metric (hyper-volume metric) [5] and, for comparisons, NSGAII is considered.

For a fair comparison, each algorithm runs for 10000 fitness evaluation. NSGAII settings are: 100 individuals, 100 iterations, mutation probability 1/(NoOfVariable*30), SBX crossover. SBMGA settings are: binary encoding with 30 binary values per each variables, 100 individuals, mutation probability 1/(NoOfVariable*30), uniform crossover. The algorithms run for 30 times and the *hypervolume*, *spacing* and *coverage* metrics were computed.

Results:

S-metric (hyper-volume metric) is a metric which is widely used in performance assessment of the MOEAs. Hyper-volume represents the volume of the *n*-dimensional space that is contained by an *n*-dimensional set of points. The dominated hyper-volume, or S-metric [5] is computed relative to a reference point and corresponds to the size of the objective space which contains the solutions which are dominated by at least one of the members of the set. Despite its computational cost, S-metric is one of the preferred performances metric as it measures both convergence and diversity. The metric value is to be maximized.

TABLE 3. Average Hypervolume values for ZDT1,2,3,4, for 30 runs

Test problem	Hypervolume	
	NSGA2	SBMGA
ZDT1	*0.775539*	**0.8402**
ZDT2	*0.515311*	**0.644929**
ZDT3	***0.579934***	0.536055
ZDT4	***0.513386***	0.507048

The results suggest that SBMGA performs significally better than NSGAII for the first two test problems. For ZDT3 problem, NSGAII performs better and for the ZDT4 problem, even the average hypervolume is higher for NSGAII, the statistical test did not confirm that NSGAII performs better than SBGMA. As the hypervolume metric measure both: the closeness to the Pareto front and the distribution across the approximated Pareto front, we considered that the smaller hypervolume value computed for SBMGA in ZDT3 case is given by the solutions' distribution across the Pareto front. Spacing and Coverage metric confirms that, for ZDT3, SBMGA offers closer solutions to the real Pareto front, but with a weaker distribution than NSGAII.

For a close comparison of two non-dominated sets resulted by using different algorithms, *Coverage* metric [5] is used. Coverage computes the fraction of solutions in one set of non-dominated solutions (found by one algorithm) that are dominated by those obtained by the other algorithm. The results presented in TABLE 4 shows that, except for the fourth problem (ZDT4), SBMGA converge closer to the Pareto front than NSGAII.

TABLE 4. Average Coverage metric's values for ZDT1,2,3,4, for 30 runs

Test problem	Coverage	
ZDT1	C(SBMGA,NSGAII)	**0.9992**
	C(NSGAII,SBMGA)	0.5038
ZDT2	C(SBMGA,NSGAII)	**0.9832**
	C(NSGAII,SBMGA)	0.5919
ZDT3	C(SBMGA,NSGAII)	**0.7638**
	C(NSGAII,SBMGA)	0.66465
ZDT4	C(SBMGA,NSGAII)	0.3514
	C(NSGAII,SBMGA)	**0.6015**

Spacing metric is designed to measure the distribution of the vectors among the non-dominated vectors found so far. Schott [16] proposed such a metric measuring the range (distance) variance of the neighbouring vectors in the non-dominated vectors found. *Spacing* metric measures the variation distance between neighbour vectors of the approximated Pareto front.

Regarding the distribution along the Pareto front, NSGAII performs significally better for ZDT2 and ZDT3, equivalentlly for ZD1 and worse for ZDT4.

TABLE 5. Average Spacing values for ZDT1,2,3,4, for 30 runs

Spacing	NSGAII	SbMGA
ZDT1	**0.007517**	0.00889
ZDT2	**0.002748**	0.016571
ZDT3	**0.008331**	0.029442
ZDT4	0.143162	**0.016236**

Overall, the SBGMA's comparison with NSGAII is presented in TABLE 6.

TABLE 6. SBMGA's comparisons with NSGAII: + (better), -(worse), = (equivalent)

	Hypervolume	Coverage	Spacing
ZDT1	+	+	=
ZDT2	+	+	-
ZDT3	-	+	-
ZDT4	=	-	+

We have tested the algorithm for DTLZ1 and DTLZ2 test problems with 3, 4 and 5 criteria. Except for DTLZ1, the convergence of SBMGA is promising, but due to the lack of explicit diversity preservation mechanism, the solutions in the approximation fronts in higher dimensions are not well-distributed.

IV. CONCLUSIONS

We propose a simple genetic algorithm of O(m*n**log*n) computational complexity for multiobjective optimization. The popular Pareto-ranking procedure, which offers good approximations of the Pareto fronts for many problems is one of the most expensive procedure in evolutionary algorithms for MOP. Also, the scalarization techniques, for instance aggregation techniques, have low computational cost but weaker performance. Therefore, we investigated the possibility to design a simple genetic algorithm with a low-complexity fitness assignment procedure for multiobjective optimization. The resulted algorithm, SBMGA, is based on the observation that aggregation of the scores per each objective gives a hint of the solution's quality in multiobjective space. For a minimization/maximization problem, the individual's score for an objective represents the number of other individuals from the current population which have lower/higher corresponding objective's value. The goal is to maximize the fitness of the individuals, which is computed as an aggregation of the scores.

As for a bi-objective problem, the sum of the scores represents the distance (p-norm, where p=m-1) to reference/minimum point (0,0,…0), for multi-objective space, of order *m*, we propose as fitness function, the generalized distance between the vector of scores to the reference point (0,0,…0), respectively, the Minkowski distance of order p=m-1.

For accelerating the convergence, we have included an elitistic replacement procedure that allows the high-qualified individuals to survive in the next generation. The fitness threshold is a computable value at each generation, and it is not an extra parameter of the algorithm.

Experimental results show that, for bi-objective problems, SBMGA performs, somewhat better than a Pareto-ranking based algorithm, NSGAII. For more than two objectives, even the SBMGA converges toward the true Pareto front, the lack of diversity preservation mechanism prevents a good distribution along the front. Nevertheless, score-based fitness assignment procedure has a lower computational complexity than Pareto-ranking procedures. SBMGA is unsophisticated and less expensive: the main advantages are its lower computational complexity in comparison with state-of-art MOEA's and the absence of any extra parameter. The conducted experiments strongly suggest that mapping the objective space into the scores' space is a promising research direction. As further research, we propose to investigate the SBMGA's performance for many objective optimization problems and to incorporate a diversity preservation mechanism.

ACKNOWLEDGEMENT

„This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI -UEFISCDI, project number PN-III-P2-2.1-PED-2016-1835, within PNCDI III".

REFERENCES

[1] W Cohon, Jared L. Multiobjective programming and planning. Courier Corporation, 2013.

[2] R. E. Steuer. Multiple Criterion Function Goal Programming Applied to Managerial Compensation Planning. Computers & Operations Research, 10(4):299-309, 1983.

[3] Koski, Juhani. "Defectiveness of weighting method in multicriterion optimization of structures." International Journal for Numerical Methods in Biomedical Engineering 1.6 (1985): 333-337.

[4] Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems.: Springer, 2007.

[5] Zitzler, Eckart, and Lothar Thiele. "Multiobjective optimization using evolutionary algorithms—a comparative case study." In: International Conference on Parallel Problem Solving from Nature. Springer Berlin Heidelberg (1998): 292-301

[6] Buzdalov, Maxim, and Anatoly Shalyto. "A provably asymptotically fast version of the generalized Jensen algorithm for non-dominated sorting." International Conference on Parallel Problem Solving from Nature. Springer International Publishing, 2014.

[7] Fortin, Félix-Antoine, Simon Grenier, and Marc Parizeau. "Generalizing the improved run-time complexity algorithm for non-dominated sorting." Proceedings of the 15th annual conference on Genetic and evolutionary computation. ACM, 2013.

[8] Jensen, Mikkel T. "Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms." IEEE Transactions on Evolutionary Computation 7.5 (2003): 503-515.

[9] Srinivas, Nidamarthi, and Kalyanmoy Deb. "Muiltiobjective optimization using nondominated sorting in genetic algorithms." Evolutionary computation 2.3 (1994): 221-248.

[10] Zitzler, Eckart, and Lothar Thiele. "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach." IEEE transactions on Evolutionary Computation 3.4 (1999): 257-271.

[11] Deb, Kalyanmoy, et al. "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II." International Conference on Parallel Problem Solving From Nature. Springer Berlin Heidelberg, 2000.
[12] Zitzler, Eckart, M. Laumanns, and Lothar Thiele. “SPEA2: Improving the Strength Pareto Evolutionary Algorithm”, TIK Report 103, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, 2001.
[13] Knowles, Joshua D., and David W. Corne. "Approximating the nondominated front using the Pareto archived evolution strategy." Evolutionary computation 8.2 (2000): 149-172.
[14] Schaffer, J. David. "Multiple objective optimization with vector evaluated genetic algorithms." Proceedings of the 1st international Conference on Genetic Algorithms. L. Erlbaum Associates Inc., 1985.
[15] Van Veldhuizen, D A and Lamont G B. Multiobjective evolutionary algorithms: analyzing state-of-the-art. Evolutionary Computing, 2000, MIT Press, Cambridge, MA, 8(2): 125-147.
[16] Schott, J. R. (1995), ‘Fault tolerant design using single and multicriteria genetic algorithm optimization’, M.S. thesis, Department Aeronautics and Astronautics, Massachusetts Inst. Technology, Cambridge, MA.
[17] Van Veldhuizen, David A. Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. No. AFIT/DS/ENG/99-01. AIR FORCE INST OF TECH WRIGHT-PATTERSONAFB OH SCHOOL OF ENGINEERING, 1999.
[18] Zhang, Qingfu, and Hui Li. "MOEA/D: A multiobjective evolutionary algorithm based on decomposition." IEEE Transactions on evolutionary computation 11.6 (2007): 712-731.
[19] Zitzler, Eckart, and Simon Künzli. "Indicator-based selection in multiobjective search." International Conference on Parallel Problem Solving from Nature. Springer Berlin Heidelberg, 2004.
[20] Fonseca, Carlos M., and Peter J. Fleming. "Multiobjective genetic algorithms made easy: selection sharing and mating restriction." (1995): 45-52.
[21] Horn, Jeffrey, Nicholas Nafpliotis, and David E. Goldberg. "A niched Pareto genetic algorithm for multiobjective optimization." Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on. Ieee, 1994.
[22] Coello, C. A. C., List of References on Evolutionary Multiobjective Optimization. Available: http://delta.cs.cinvestav.mx/~ccoello/EMOO/EMOObib.html
[23] Zitzler, Eckart, Kalyanmoy Deb, and Lothar Thiele. "Comparison of multiobjective evolutionary algorithms: Empirical results." Evolutionary computation 8.2 (2000): 173-195.