BULETINUL Universităţii Petrol – Gaze din Ploieşti	Vol. LIX No. 1/2007	19 - 28	Seria Matematică - Informatică - Fizică

Performance Measures for Multi-objective Optimization Algorithms

Elena Simona Nicoară

Universitatea Petrol-Gaze din Ploieşti, Bd. Bucureşti 39, Ploieşti, Catedra de Informatică
e-mail: snicoara @upg-ploiesti.ro

Abstract

In the real world, the most optimization problems are multi-objective. For the complex ones we do not often have polynomial algorithms which return the exact solution(s) in practical time. For this reason, we use approximation algorithms which find solution(s) (near) optimal in a practical time. Alongside the aspects related to comparing two multi-objective solutions, there are also aspects related to measuring the performance of an algorithm. In the paper we present the most important performance measures for multi-objective optimization algorithms. As a practical example, we have compared three genetic algorithms on the bi-objective JSSP test-problem ft10, and the results showed the necessity to simultaneously consider many performance measures for the multi-objective algorithms.

Key words: *multi-objective optimization, Pareto dominance, Pareto optimal solution, performance measure, approximation of the Pareto optimal set*

Multi-objective Optimization Problems

A *uni-objective optimization problem* consists in searching a solution $x \in X$, so that the objective function, $f(x)$, have a maximum or minimum value. In this case, to compare two candidate-solutions, $x^{(1)}$ and $x^{(2)}$, means to compare their objective values, $f(x^{(1)})$ and $f(x^{(2)})$.

Yet many real problems involve both the simultaneous optimization of many objectives, often in conflict one with another, and a complex large search space. Ordinarily, this space is very difficult to explore through enumerative methods, because it contains multiple possible solutions, placed in hardly accessible regions or having properties which slow down finding the solutions. In these cases, we search many solutions of compromise.

Though there are procedures to aggregate the objectives in a single one and to treat the problem as a uni-objective problem, in the case of complex problems the disadvantages of this approach have to be considered.

A *multi-objective optimization problem* is defined by a set of *n* parameters (decision variables), a set of *k* objective functions and a set of *m* constraints. The objective functions and the constraints are functions of the decision variables. The aim of optimization is to

$$\text{minimize} \quad y = f(x) = (f_1(x), f_2(x),\ldots, f_k(x)),$$

$$\text{satisfying the constraints } e(x) = (e_1(x), e_2(x),\ldots, e_m(x)) \le 0, \tag{1}$$

where $x = (x_1, x_2, \ldots, x_n) \in X$, $y = (y_1, y_2, \ldots, y_k) \in Y$.

Here x is named the *decision vector*, y is named the *objective vector*, X the *decision space* and $Y_f = f(X_f)$ the *objective space*. The *feasible set* X_f is the set of decision vectors x which satisfy the constraints $e(x) \leq 0$.

In the uni-objective optimization problems, the feasible set is totally ordered, with respect to the objective function f, while in the multi-objective optimization problems, X_f is, generally, partially ordered, with respect to the objective function f. [8] To compare two candidate-solutions in the multi-objective optimization problems there have been defined the concepts of *Pareto dominance* and *Pareto optimal solutions*. Based on the Pareto dominance we can also introduce the optimality criterion for the problem to be solved.

Without restricting the generality, we can consider that all the partial objective functions are to be minimized. The following definitions refer to such optimization problems.

Definition 1. $x^{(1)}$ dominates $x^{(2)}$ if the both following conditions are satisfied [2]:

$x^{(1)}$ is not weaker than $x^{(2)}$ in every objective, namely ${x^{(1)}}_i \prec {x^{(2)}}_i$, $\forall\, i \in \{1,2,...,k\}$;
$x^{(1)}$ is strictly better than $x^{(2)}$ at least in one objective, namely $\exists i \in \{1,2,...,k\}$ so that ${x^{(1)}}_i \prec {x^{(2)}}_i$.

The symbol $\prec$ is generally accepted to designate the dominance relation.

If $x^{(1)}$ dominates $x^{(2)}$ we say that $x^{(2)}$ is dominated by $x^{(1)}$.

Consequently, for any two candidate-solutions, one of these situations can be true:

- one solution dominates the other (is better considering all the objectives),
- the solutions can not be compared (does not exist a dominance relation between them).

Definition 2. A decision vector $x \in X$ is nondominated relative to a set $A \subseteq X$ iff [8]:

$$\neg\exists a \in A \text{ so that } a \text{ dominate } x. \tag{2}$$

Definition 3. The decision vector x is Pareto optimal if and only if it is nondominated relative to the entire set X.

In other words, a decision vector (a candidate solution) is Pareto optimal if it can not be improved in any objective without causing a degradation in at least one another objective (Pareto, 1896) [8].

Generally speaking, it does not exist a single Pareto optimal solution of a problem, but a set of Pareto optimal solutions. None of them can be identified as better than the others, in the absence of preference information for a certain objective.

The Pareto optimal solutions form the Pareto optimal set, and the correspondent objective vectors form the Pareto optimal front.

Definition 4. Let $A \subseteq X$ be a set of decision vectors. A is a local Pareto optimal set iff [8]:

$$\forall a \in A, \neg\exists x \in X \text{ so that } x \text{ dominate } a,\ \|x - a\| < \varepsilon \text{ and } \|f(x) - f(a)\| < \delta, \tag{3}$$

where $\|\ .\ \|$ is a metrics, and $\varepsilon > 0$, $\delta > 0$ very small values.

In other words, A contains those decision vectors which are not dominated by the decision vectors in a small neighborhood, in the objective space.

Definition 5. The set A is a global Pareto optimal set iff [8]:

$$\forall a \in A, \neg \exists x \in X \text{ so that } x \text{ dominate } a. \quad (4)$$

So, A is formed by the nondominated decision vectors. All the Pareto optimal solutions are not mandatory for a global Pareto optimal set, and every global Pareto optimal set is also a local Pareto optimal set. The Pareto optimal set includes the global optimum solutions.

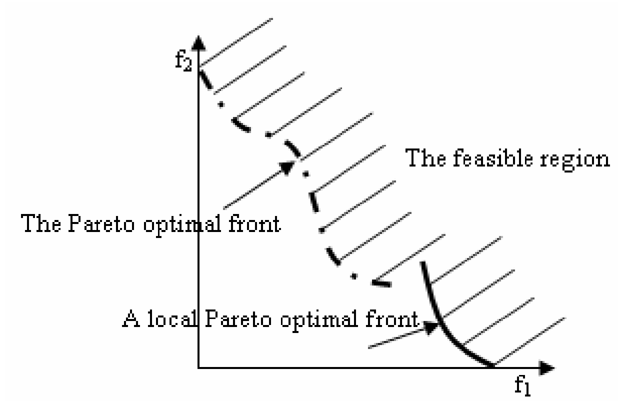

Fig. 1. Illustration of a global Pareto optimal front and a local Pareto optimal front in a bi-objective space, for a minimization problem

In the multi-objective optimization, the aim is to find as many different solutions as possible (near) Pareto optimal. A multi-objective optimization algorithm has to perform two tasks [2]:

- to guide the search towards the global Pareto optimal region and
- to maintain the population diversity (in the objective space, in the parameters space or in both of them) in the current nondominated front.

Performance Measures for Multi-objective Optimization Algorithms

In the multi-objective optimization, it is often impossible to know the Pareto optimal set. In these conditions, to measure the performance of a search algorithm is difficult enough.

The general performance criteria for the multi-objective optimization algorithms are:

- *accuracy* - how close the generated nondominated solutions are to the best known prediction;
- *coverage* - how many different nondominated solutions are generated and how well they are distributed;
- *the variance for every objective* - which is the maximum range of nondominated front, covered by the generated solutions (fraction of the maximum range of the objective in the nondominated region, covered by a nondominated set).

Starting from these criteria, there were developed various performance measures for the search algorithms, originating from various areas (statistics, computer science, biology etc.). In the following sections of the paper we present the most important performance measures, which also allow us to compare algorithms and to adjust their parameters for better results.

A Diversity Performance Measure

A comparing measure, from the solutions diversity point of view, is described by Deb et al. [3]. It is based on the successive distances of the solutions in the first front of the final population. The set of solutions in this front is compared with a uniform distribution, and then the deviation from it is determined by the formula [3]:

$$\Delta = \sum_{i=1}^{|F_0|} \frac{\left|d_i - \overline{d}\right|}{|F_0|} . \tag{5}$$

Here, d_i is the Euclidean distance between two successive solutions in the first front (F_0), in the objective space, and $\overline{d}$ is the average of these distances. In order for the distribution to take into account the entire space of the real front, we have included the frontier solutions also in the front F_0. We note that the deviation measure is computed for every run where the distinct solutions are more than three. The average of these deviations for ten runs, for example, can be a comparing measure for different algorithms. Hence, in terms of the ability to obtain more „dispersed" solutions, more uniformly distributed solutions on the first front, an algorithm with a smaller average deviation is a better option.

The Expected Number of Evaluation for Success

The performance measure ENES (Expected Number of Evaluation for Success) is determined by the following formula, which indicates the average number of evaluations to obtain the success.

$$ENES = NE / NS. \tag{6}$$

NE denotes the number of evaluations for the objective function in 20 independent runs and *NS* the number of successes in the 20 runs.

Quantitative Performance Measures If the Pareto Optimal Set Is Known

Talbi [7] submitted separate performance measures, for cases when the Pareto optimal set is known, respectively unknown. The Pareto optimal set is known when it can be identified using an enumerative method (for example Branch & Bound).

If the Pareto optimal set (*OP*) is known, Talbi defines *the absolute efficiency of the algorithm* as the proportion occupied by the Pareto solutions in the approximated Pareto optimal set (*OP**), relative to the real Pareto optimal set [7]:

$$E = \frac{\left|OP^* \cap OP\right|}{|OP|} . \tag{7}$$

A solution belonging to *OP*,* without being Pareto optimal, is not necessarily a weak solution. The smaller the distance between *OP* and *OP**, the better the algorithm.

Let $d(x,y)$ be a distance between two solutions in the objective space (the Tchebycheff norm):

$$d(x,y) = \sum_{i=1}^{n} \lambda_i \left| f_i(x) - f_i(y) \right| , \tag{8}$$

where λ_i is a parameter which allows to normalize the various criteria.

The distance between a solution in *OP** and a solution *y* can be defined by the following formula [7]:

$$d(OP^*, y) = \min d(x,y), x \in OP^* . \tag{9}$$

One must notice that we also can use other distances between two solutions in the objective space besides the Tchebycheff norm.

Based on these definitions, we can consider many *quantitative measures* for algorithms [7]:

- the longest distance between the sets OP^* and OP [7]:

$$WD = \max d(OP^*, y), y \in OP \, . \tag{10}$$

- the average distance between the sets OP^* and OP [7]:

$$MD = \frac{\sum_{y \in OP} d(OP^*, y)}{|OP|} \, . \tag{11}$$

- a measure for the uniformity of the set OP^* [7]:

$$DIV = \frac{WD}{MD} \, . \tag{12}$$

Qualitative Measures Based on Approximations of the Pareto Optimal Set

If the Pareto optimal set is not known, then we work with an approximation of it.

M.P. Hansen and A. Jaskiewicz have introduced a family of comparing relations between approximations in 1998 [7].

Definition 6. An approximation A weakly dominates an approximation B if [7]:

$$A \neq B \text{ and } ND(A \cup B) = A, \tag{13}$$

where $ND(S)$ is the set of solutions nondominated by the set S. This means that A weakly dominates B if for every solution $x_B \in B$ there is a solution $x_A \in A$ which dominates x_B or is equal to x_B and at least a solution in A is not in B (see figure 1.(a)).

Definition 7. An approximation A strongly dominates an approximation B if [7]:

$$ND(A \cup B) = A \text{ and } B - ND(A \cup B) \neq \varnothing. \tag{14}$$

In other words, for every solution $x_B \in B$ there is a solution $x_A \in A$ which dominates x_B or is equal to x_B and at least a solution $x_B \in B$ is dominated by a solution $x_A \in A$ (see figure 1.(b)).

Definition 8. An approximation A totaly dominates an approximation B if [7]:

$$ND(A \cup B) = A \text{ and } B \cap ND(A \cup B) = \varnothing. \tag{15}$$

This means that every solution $x_B \in B$ is dominated by a solution $x_A \in A$ (see figure 1.(c)).

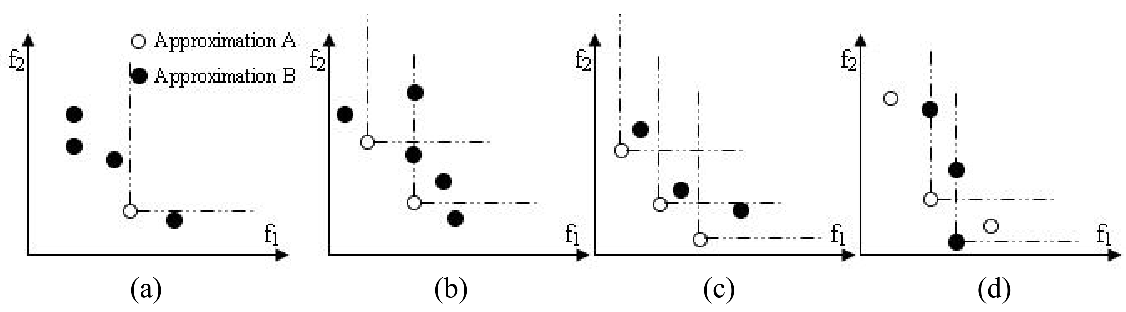

Fig. 2. The dominance relations between two approximations, for two/objective minimization: (a) weak dominance, (b) strong dominance, (c) total dominance, (d) incomparable approximations.

Figure 1.(d) illustrates two incomparable approximations – neither dominates the other. Each above relation defines a partial order in the approximations set. Moreover, these relations allow only a *qualitative comparison* between approximations.

To judge an approximation as satisfactory, attributes such as coverage, uniformity, cardinality and adjacency are important under two aspects: the rules to stop an approximation algorithm and the procedures to tune its parameters.

The Contribution of a Metaheuristic Relative to Another Metaheuristic

In [7] is also submitted to research a comparing measure for two approximation methods, two metaheuristics, M_1 and M_2, named the *relative efficiency measure*. It allows to compare the methods by counting the Pareto solutions, identified by M_1, dominated by those identified by M_2. Let OP^*_i be the Pareto optimal set identified by M_i and let C be the Pareto solution set, common to both OP^*_i. Let W_1, W_2, be the solutions sets in OP^*_1, respectively OP^*_2 which dominate the solutions in OP^*_2, respectively OP^*_1. Let L_1 and L_2 be the solutions sets in OP^*_1, respectively OP^*_2 dominated by the solutions in OP^*_2 and respectively OP^*_1.

The set of solutions in OP^*_1, OP^*_2 which have dominance relation with no solution in OP^*_2, respectively OP^*_1 will be then [7]:

$$N_i = OP_i^* - (C \cup W_i \cup L_i), i \in \{1,2\} \quad (16)$$

Let OP^* be the set of Pareto optimal solutions found by both methods [7]:

$$OP^* = C \cup W_1 \cup N_1 \cup W_2 \cup N_2 \quad (17)$$

The *contribution of the metaheuristic M_1 relative to M_2* is the ratio of the solutions in OP^* produced by M_1 [7]:

$$Cont(M_1 / M_2) = \frac{\frac{|C|}{2} + |W_1| + |N_1|}{|C| + |W_1| + |N_1| + |W_2| + |N_2|} = \frac{\frac{|C|}{2} + |W_1| + |N_1|}{|OP^*|} \quad (18)$$

We note that if the two metaheuristics provide the same solutions, then

$Cont(M_1/M_2) = Cont(M_2/M_1) = ½$, and if all the solutions produced by M_2 are dominated by the solutions produced by M_1, then $Cont(M_2/M_1) = 0$.

The Distance Between a Set of Solutions and a Reference Set

To evaluate a set of solutions S_j, Czyzack and Jaszkiewicz (1998) proposed a measure based on the distance from a reference set (the Pareto optimal set or a near Pareto optimal set); more precisely, the average distance from every reference solution to the nearest solution in S_j. This measure, entitled $D1_R$ by Knowles and Corne (2002), is defined in terms of a reference set S^*, as follows [4]:

$$D1_R(S_j) = \frac{1}{|S^*|} \sum_{y \in S^*} \min\{d_{xy} | x \in S_j\} \quad (19)$$

Here, d_{xy} is the distance between a solution x and a reference solution y in the N-dimensional objective space [4]:

$$d_{xy} = \sqrt{(f_1^*(y) - f_1^*(x))^2 + \ldots + (f_N^*(y) - f_N^*(x))^2}\,, \tag{20}$$

where $f_i^*(.)$ is the i^{th} objective, normalized using the reference set S^*.

The smaller $D1_R(S_j)$, the better the solutions set S_j. Note that $D1_R$ is not the average distance from every solution in S_j to the nearest solution in S^*; this corresponds to the general distance. While the general distance can evaluate only the proximity of the set S_j to S^*, $D1_R(S_j)$ can evaluate both proximity to S^* and the distribution of S_j.

If S_j proves to be a good approximation for the real Pareto optimal set, the solution x in S_j, chosen by the decident, can approximate the best solution x^*. In this case, the loss caused by the choice of x instead of x^* can be approximately measured by the distance between x and x^* in the objective space. If the solution x^* is not known, we can not directly measure this distance. But an expected value for this distance can be coarsely estimated by the average distance from every Pareto optimal solution to the nearest solution available. The measure $D1_R$ corresponds exactly to this approximation.

With the aim of evaluate a set of solutions S_j, let $S = S_1 \cup S_2 \cup \ldots \cup S_J$ be the union of J sets of solutions. A direct performance measure for the set S_j relative to S is the proportion of the solutions in S_j which are not dominated by any other solution in S [4]:

$$R_{NDS}(S_j) = \frac{\left|S_j - \{x \in S_j \,|\, \exists y \in S : y \prec x\}\right|}{\left|S_j\right|}\,. \tag{21}$$

$y \prec x$ denotes the dominance of y on x. The bigger is ratio $R_{NDS}(S_j)$, the better is the set S_j.

Another performance measure is the *front diversity relative to the solutions space*. Because the Pareto optimal front is defined relative to the objective space, it is natural that the performance measures based on the Pareto optimal front should also be defined relative to the objective space.

Other Performance Measures

In [1] authors define the following performance measures for multi-objective algorithms:

- The convergence order of the algorithm;
- Local performance measures, which evaluate the algorithm power to improve the population (of solutions) state from the generation T to the generation T+1;
- Global performance measures to evaluate the algorithm's behavior (behavior in infinity conditions) in the long term. These return the needed computer resources and time resources to attain the optimum.

Other comparing measures are:

- The best objective value found in X runs;
- The average duration, in 10 runs, to obtain the same solution(s);
- The consistency associated to the localization of all the optima;
- The progress rate from one generation to the next;
- The progress velocity (the average distance in the search space, made in the beneficial direction / the number of evaluations for the objective function).

Case study. Analysis of results

In order to validate the need for considering many performance measures for algorithms used to solve multi-objective problems, we tested three genetic algorithms (GA) on the bi-objective Job Shop Scheduling Problem (JSSP) ft10. It consists in scheduling 10 jobs, each formed of 10 operations, on 10 machines, being known the precedence relations and the processing times. Minimization of the makespan and minimization of the number of late jobs against the deadline 1000 units are the objectives.

Although this problem is a small-sized one, it is among the most difficult JSSP test-problems. A solution was identified only after 24 years from its enunciation, and after 2 more years it was demonstrated that the optimal solution (in the uni-objective formulation) has the makespan 930 units. The tested genetic algorithms are: an elitist genetic algorithm, NSGA_II (Non-dominated Sorting Genetic Algorithm II) designed by Deb [3] and NSGA_II_DAR (NSGA_II with Dynamic Application of genetic operators and partial population Reinitialization). The last one was proposed by the author in [6].

For the ft10 problem, the Pareto optimal set is not known. The performance measures for the tested algorithms and their values are presented in table 1 below.

Table 1. The performance measures for three algorithms solving the bi-objective ft10

PERFORMANCE MEASURE	**ALGORITHM**	**ELITIST GA**	**NSGA_II**	**NSGA_II DAR**
The best makespan solution in the final front	makespan	**1054**	1175	1096
	number of late operations	**5**	16	8
The best aggregated objective value (coefficients 0.8, 0.2)		**844.2**	943.2	878.4
The average aggregated objective values in the final front		**823.5**	934.6	890.3
The worst aggregated objective values in the final front		**893.8**	992.1	956.5
Diversity in the schedules space				
The number of different nondominated solutions, per test		4.3	5.5	**8.36**
Minimum schedule overlapping		0.87	**0.80**	0.83
Maximum schedule overlapping		**0.99**	**0.99**	**0.99**
Average schedule overlapping		0.95	**0.92**	0.94
Diversity in the objective space				
The diversity performance measure described by Deb				
	Minimum	0.07	**0.01**	**0.01**
	Average	9.30	**5.33**	6.24
The variance for the first objective (the most important)				
Range dimension for the makespan		412	**456**	213
Run time, per generation* (seconds)		3.6	**2.4**	3.2

(*) The results were obtained on a processor AMD Athlon 1600 MHz and 256 MB RAM.

In the table, the bold values indicate the best value for the current performance measure. We notice that, from the solution fitness point of view, the best algorithm is the elitist genetic algorithm. If this would be the single perspective to consider, as in the uni-objective case, we may conclude that other analysis is not necessary, and the best algorithm is the elitist GA. But the distribution of the solutions in the space is very important for the multi-objective problems.

The level of distribution was measured by many indicators, grouped in three classes: diversity in the schedules space, diversity in the objective space and the variance for the first objective (the most important).

For the permutation encoding (associated with scheduling solutions) we used to measure the diversity of found solutions the number of different nondominated solutions, per test and the indicator named schedule overlapping, defined in [5]. The last one takes into account both the order of operations in the schedule sequence and the start processing and stop processing times of the operations. It quantifies the similarity of two schedules so those identical schedules have a measure equal to 1 and two completely different schedules a measure equal to 0.

Consequently, reduced values for this measure indicate a better distribution of the solutions in the schedules space. From the diversity point of view, we note that NSGA_II is the best algorithm closely followed by NSGA_II_DAR. If the time resources are strongly limited, the user would consider also the run time needed. And from this perspective, the NSGA_II proves to be, on the average, the best option to take. As we can see, there is no one single algorithm to prove the best from all perspectives, namely: fitness and diversity of the solutions and time complexity. These data prove that we can not efficiently evaluate the quality of an algorithm form a single perspective.

In conclusion, to better assist the user to decide what algorithm to use for a certain problem at a given time, under particular circumstances, we must consider many performance measures.

Conclusion

In the multi-objective optimization, the aim is to find as many different solutions as possible (near) Pareto optimal. Because there are many types of complex problems (multimodal, deceptive etc.) and no single algorithm is the best for every type and even for every instance, many methods / algorithms have been developed in the literature to solve these problems.

The described performance indicators allow to measure the performance of an algorithm, to adjust the parameters of an algorithm to obtain better results and also to compare different algorithms.

The measures can be quantitative or qualitative. We can treat separately the performance, when the Pareto optimal set is known or unknown. These measures take into account: the solution diversity, the expected number of evaluations for attaining the success, the proportion occupied by the Pareto solutions in the approximated Pareto optimal set, the proportion of the solutions found by an algorithm in the set of solutions found by two algorithms, the distance between an identified set of solutions and a reference set of solutions, the convergence order of the algorithm, the power to improve the population from a generation to another, the run time for attaining a certain set of solutions etc.

Ordinarily, for the multi-objective optimization, we compare many sets of solutions obtained by different algorithms or obtained by the different parameters specifications of the same algorithm.

We recommend *the simultaneous use of many performance measures*, because it is impossible to evaluate all the aspects in every set of solutions using a single measure. The above presented case study is a proof for this recommendation.

References

1. Beyer, H.G., Schwefel, H.P., Wegener, I. - How to analyse evolutionary algorithms, *Theoretical Computer Science* 287, pp. 101-130, 2002
2. Deb, K. - Multi-objective genetic algorithms: Problem difficulties and construction of test problems, *Evolutionary Computation Journal*, 7(3), pp. 205-230, 1999

3. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T. - A fast elitist nondominated sorting genetic algorithm for multi-objective optimization: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6(2), pp.182–197, 2002
4. Holland, J.H. - Genetic algorithms, *Scientific American*, 267(1), pp. 44-50, 1992
5. Jensen, M.T. - *Robust and flexible scheduling with evolutionary computation*, doctoral thesis, Dissertation Series DS-01-10, Aarhus University, Denmark, 2001
6. Nicoară, E.S. - *Contribuţii privind utilizarea algoritmilor evolutivi în conducerea proceselor de fabricaţie în industria medicamentelor*, doctoral thesis, Ploieşti, 2007
7. Talbi, E.G. - Métaheuristiques pour l'optimisation combinatoire multi-objectif: Etat de l'art, *C.N.E.T (France Telecom) report*, Oct. 1999
8. Zitzler, E., Deb, K., Lothar, T. - Comparison of multiobjective evolutionary algorithms: Empirical results, *Evolutionary Computation Journal*, 8(2), pp. 125-148, 2000

Indicatori de Performanţă pentru Algoritmi de Optimizare Multiobiectiv

Rezumat

În lumea reală, majoritatea problemelor de optimizare sunt multiobiectiv. Pentru cele complexe şi de mari dimensiuni adeseori nu avem la dispoziţie algoritmi de complexitate polinomială care să redea soluţia(soluţiile) exactă(e) în timp util. De aceea, se folosesc algoritmi aproximativi care găsesc soluţii (aproape) optime în timp util. Împreună cu aspectele legate de compararea a două soluţii multiobiectiv, apar şi aspectele legate de măsurarea performanţei unui algoritm şi de comparare a doi algoritmi. În lucrare sunt prezentaţi cei mai importanţi indicatori de performanţă pentru algoritmi de optimizare multiobiectiv. Drept exemplu, am comparat trei algoritmi genetici aplicaţi pe problema-test biobiectiv JSSP ft10, iar rezultatele au demonstrat necesitatea considerării simultane a mai multor indicatori de performanţă pentru algoritmi multiobiectiv.