

Survey of Evolutionary Algorithms Used in Multiobjective Optimization

Vassil Guliashki, Hristo Toshev, Chavdar Korsemov

Institute of Information Technologies, 1113 Sofia

E-mails: vggul@yahoo.com toshv@iinf.bas.bg chkorsemov@iinf.bas.bg

I. Introduction

The evolutionary multi-objective optimization (EMOO) is a popular and useful field of research and developing algorithms to solve many real-life multiobjective problems (see for example [6, 18, 38]). During the World Congress of Computational Intelligence (WCCI) in Vancouver 2006, EMOO has been evaluated as one of the three fastest growing fields of research and application among all computational intelligence topics.

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 reasons for the popularity of EO algorithms are as follows: (i) They do not require any derivative information; (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 [23]). The use of a population in EO algorithms has a number of advantages (see [3]): (i) it provides an EO procedure with a parallel processing power, (ii) it allows EO procedures to find multiple optimal solutions, thereby facilitating the solution of multimodal and multiobjective optimization problems, and (iii) 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. The shortcoming of working with a population of solutions is the computational cost and the memory necessary for the execution of each iteration.

Some EO algorithms use an elitism operator, which combines the old population with the newly created population and chooses to keep better solutions from the combined population. Such an operation makes sure that an algorithm has a monotonically non-degrading performance. The presented survey of EO

algorithms designed to solve multi-objective optimization problems categorizes the algorithms as elitist (which use an elitism operator) and non-elitist algorithms.

The paper is organized as follows: In section II the principles of evolutionary multi-objective optimization are given. A short survey of evolutionary multi-objective optimization (EMOO) algorithms is presented in section III. Convergence properties of some EMOO algorithms are discussed in section IV. Some current and future challenges are outlined in section V. Conclusions are made in section VI.

II. EMOO principles

The multi-objective optimization problem has the following general form:

$$\begin{aligned}
(1) \quad & \text{Minimize/Maximize } f_m(\mathbf{x}), & m = 1, 2, \dots, M; \\
& \text{subject to} \\
(2) \quad & g_j(\mathbf{x}) \geq 0, & j = 1, 2, \dots, J; \\
(3) \quad & h_k(\mathbf{x}) = 0, & k = 1, 2, \dots, K; \\
(4) \quad & x_i^{(L)} \leq x_i \leq x_i^{(U)}, & i = 1, 2, \dots, n.
\end{aligned}$$

Here we consider the term “solution” as a variable vector and the term “point” as the corresponding objective vector.

A solution $\mathbf{x} \in \mathbf{R}^n$ is a vector of n decision variables: $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$. 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 $S \subset \mathbf{R}^n$. The objective functions (1) constitute a M -dimensional space, called objective space $Z \subset \mathbf{R}^M$. For each solution \mathbf{x} in the decision variable space, there exists a point $\mathbf{z} \in \mathbf{R}^M$ in the objective space, denoted by $\mathbf{f}(\mathbf{x}) = \mathbf{z} = (z_1, z_2, \dots, z_M)^T$.

The domination between two solutions is defined as follows (see [3, 18]):

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

1. The solution $\mathbf{x}^{(1)}$ is not worse than $\mathbf{x}^{(2)}$ in all objectives. Thus, the solutions are compared based on their objective function values (or location of the corresponding points $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$ on the objective space).

2. The solution $\mathbf{x}^{(1)}$ is strictly better than $\mathbf{x}^{(2)}$ in at least one objective.

All points which are not dominated by any other point $\mathbf{z} \in Z$ are called the non-dominated points of class one, or simply 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.

The Pareto-optimal solutions are defined in multiobjective optimization (see [3, 18]) as follows:

Definition 2. If for every member x in a set P there exists no solution y (in the neighborhood of x such that $\|y - x\|_\infty \leq \varepsilon$, where ε is a small positive scalar) dominating any member of the set P , then the solutions belonging to the set P constitute a local Pareto-optimal set.

In the multi-objective optimization there are two ideal goals:

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

An EMOO algorithm works with the following principle in handling multiobjective optimization problems (see [3]):

Step 1. Find multiple non-dominated points as close to the Pareto-optimal front as possible, with a wide trade-off among objectives.

Step 2. Choose one of the obtained points using higher-level information.

In the “a posteriori” MCDM approaches (also known as “generating MCDM methods”) the task of finding multiple Pareto-optimal solutions is achieved by executing many independent single-objective optimizations, each time finding a single Pareto-optimal solution (see [3]). A parametric scalarizing approach (such as the weighted-sum approach, ε -constraint approach, and others) can be used to convert multiple objectives into a parametric single-objective function. By simply varying the parameters (weight vector or ε -vector) and optimizing the scalarized function, different Pareto-optimal solutions can be found. In contrast, in an EMOO procedure, multiple Pareto-optimal solutions are attempted to be found in a single simulation by emphasizing multiple non-dominated and isolated solutions.

III. Short survey of evolutionary algorithms in the EMOO

Here we present a short chronological survey of the basic EMOO algorithms during the last two decades in terms of non-elitist and elitist algorithms.

In brief the aggregating functions algorithms will be also mentioned (see [47]), which could be used in combination either with the non-elitist or with the elitist approach.

– Aggregation functions

Each evolutionary algorithm needs a scalar fitness function to work. Then, in the multiobjective case, it is almost natural to propose a combination of all objectives into a single one using either an addition, multiplication or any other combination of arithmetical operations, devised by the developer of the algorithm. In fact, this is also the oldest mathematical programming method for multi-objective optimization, since it can be derived from the Kuhn-Tucker conditions for nondominated solutions (see [35]). An example of this approach is a sum of weights of the form:

$$(5) \quad \text{Min} \sum_{i=1}^m w_i f_i(x)$$

where $w_i \geq 0$ are the weighting coefficients representing the relative importance of the m objective functions of problem (1)-(4). The drawback here is that the weights should be selected very intelligently, and this is difficult (see [45, 46]).

III. 1. Non-Elitist Multi-Objective Evolutionary Algorithms

– Schaffer’s vector-evaluated genetic algorithm (VEGA)

The first implementation of a real multi-objective evolutionary algorithm (vector-evaluated genetic algorithm or VEGA) was suggested by David Shaffer in the year 1984 (see [45, 46]). He modified the simple three-operator genetic algorithm (with selection, crossover and mutation) by performing independent selection cycles according to each objective. The selection method is separated for each individual objective to fill up a portion of the mating pool. Then the entire population is thoroughly shuffled to apply crossover and mutation operators. This is performed to achieve the mating of individuals of different subpopulation groups. The algorithm worked efficiently for some generations but in some cases suffered from its bias towards some individuals or regions (mostly individual objective champions). This does not fulfill the second goal in multi-objective optimization, mentioned in section II.

– Multi-Objective Genetic Algorithm (MOGA)

This algorithm is suggested by Fonseca and Fleming in [22]. In MOGA, the rank of a certain individual corresponds to the number of chromosomes in the current population by which it is dominated. All non-dominated individuals are assigned the highest possible fitness value (all of them get the same fitness, such that they can be sampled at the same rate), while dominated ones are penalized according to the population density of the corresponding region to which they belong (i.e., fitness sharing is used to verify how crowded is the region surrounding each individual).

– Niche-Pareto Genetic Algorithm (NPGA)

This algorithm is suggested by Horn, Nafpliotis and Goldberg in [28]. The NPGA uses a tournament selection scheme based on Pareto dominance. The basic idea of the algorithm is: two individuals are randomly chosen and compared against a subset from the entire population (typically, around 10% of the population). If one of them is dominated (by the individuals randomly chosen from the population) and the other is not, then the non-dominated individual wins. All the other situations are considered a tie (i.e., both competitors are either dominated or non-dominated). When there is a tie, the result of the tournament is decided through fitness sharing.

– Non-dominated Sorting Genetic Algorithm (NSGA)

This algorithm is suggested by Srinivas and Deb [48] and it is known as Non-dominated Sorting Genetic Algorithm (NSGA). The NSGA is based on several layers of classifications of the individuals as suggested by Goldberg [23]. Before selection is performed, the population is ranked on the basis of non-domination: all non-dominated individuals are classified into one category (with a dummy fitness value, which is proportional to the population size, to provide an equal reproductive potential for these individuals). To maintain the diversity of the population, these classified individuals are shared with their dummy fitness values. Then this group of classified individuals is ignored and another layer of non-dominated individuals is considered. The process continues until all individuals in the population are classified. Since individuals in the first front have the maximum fitness value, they always get more copies than the rest of the population. The algorithm of the NSGA is not very efficient, because Pareto ranking has to be repeated over and over again.

– Investigations of Tanaka

Tanaka has developed the first scheme to incorporate user's preferences into an EMOO algorithm [49]. In real-world applications it is normally the case that the user does not need the entire Pareto optimal set, but only a small portion of it. Normally the user can define certain preferences that can narrow the search and that can magnify certain portions of the Pareto front.

– Comparative analysis

Making comparative analysis of the algorithms, above pointed, it is established with no doubt, that MOGA is excelling, followed by NPGA and NSGA.

The main conclusion about the implementations of this generation of GA is, that in order one EMOO algorithm to be successful, a good mechanism for the selection of the non-dominated solutions has to be combined with a good mechanism for variety support, which will guarantee the good performance of the algorithm.

III. 2. Elitist Multi-Objective Evolutionary Algorithms

The elitist EMOO methodologies include an elite-preservation mechanism in their procedures. As above mentioned the addition of elitism in an evolutionary optimization algorithm provides a monotonically non-degrading performance. The non-elitist EMOO algorithms do not use such a mechanism and usually perform worse than the elitist algorithms.

The wide development of EMOO algorithms in the recent years has begun after the works of Eckart Zitzler [56], due to it the elitism has become a standard mechanism in the development in this direction. In the context of multiobjective optimization, elitism usually (although not necessarily) refers to the use of an external population (also called secondary population) to retain the non-dominated individuals found along the evolutionary process. The main motivation for this mechanism is the fact that a solution that is non-dominated with respect to its current population is not necessarily non-dominated with respect to all the populations that are produced by an evolutionary algorithm. Thus, what we need is a way of guaranteeing that the solutions that we will report to the user are non-dominated with respect to every other solution that our algorithm has produced. Therefore, the most intuitive way of doing this is by storing in an external memory (or archive) all the non-dominated solutions found. If a solution that wishes to enter the archive is dominated by its contents, then it is not allowed to enter. Conversely, if a solution dominates anyone stored in the file, the dominated solution must be deleted.

After the theory offered by Zitzler, most of the researchers began to incorporate external populations in their EMOO algorithms and the use of this mechanism (or an alternative form of elitism) became a common practice. In fact, the use of elitism is a theoretical requirement in order to improve and guarantee convergence of an EMOO algorithm and therefore it is important.

– Non-dominated Sorting Genetic Algorithm II (NSGA-II)

This algorithm is known as Non-dominated Sorting Genetic Algorithm II (NSGA-II) is introduced by Deb and Agarwal in [17] as an improved version of the NSGA [48]. In NSGA-II, for each solution one has to determine how many solutions dominate it and the set of solutions to which it dominates. The NSGA-II estimates the density of solutions surrounding a particular solution in the population

by computing the average distance of two points on either side of this point along each of the objectives of the problem. This value is the so-called crowding distance. During selection, the NSGA-II uses a crowded-comparison operator which takes into consideration both the non-domination rank of an individual in the population and its crowding distance (i.e., non-dominated solutions are preferred over dominated solutions, but between two solutions with the same non-domination rank, the one that resides in the less crowded region is preferred). The NSGA-II does not use an external memory. Instead, the elitist mechanism of the NSGA-II consists of combining the best parents with the best offspring obtained. Its mechanism is better.

– **Strength Pareto Evolutionary Algorithm (SPEA)**

This algorithm is known as Strength Pareto Evolutionary Algorithm (SPEA) and was introduced by Zitzler and Thiele in [56]. This approach was conceived as a way of integrating different EMOO algorithms. SPEA uses an archive containing non-dominated solutions previously found (the so-called external non-dominated set). At each generation, non-dominated individuals are copied to the external non-dominated set. For each individual in this external set, a strength value is computed. This strength is similar to the ranking value of MOGA [22], since it is proportional to the number of solutions to which a certain individual dominates. In SPEA, the fitness of each member of the current population is computed according to the strengths of all external non-dominated solutions that dominate it. The fitness assignment process of SPEA considers both closeness to the true Pareto front and even distribution of solutions at the same time. Thus, instead of using niches based on distance, Pareto dominance is used to ensure that the solutions are properly distributed along the Pareto front. Although this approach does not require a niche radius, its effectiveness relies on the size of the external non-dominated set. In fact, since the external non-dominated set participates in the selection process of SPEA, if its size grows too large, it might reduce the selection pressure, thus slowing down the search. Because of this, the authors decided to adopt a technique that primes the contents of the external non-dominated set so that its size remains below a certain threshold.

– **Strength Pareto Evolutionary Algorithm 2 (SPEA 2)**

This algorithm is a second algorithm by Zitzler and Thiele, and is known as Strength Pareto Evolutionary Algorithm 2 (SPEA2) [36, 54]. It has three main differences with respect to its predecessor SPEA:

- (1) it incorporates a fine-grained fitness assignment strategy which takes into account for each individual the number of individuals that dominate it and the number of individuals by which it is dominated;
- (2) it uses a nearest neighbor density estimation technique which guides the search more efficiently;
- (3) it has an enhanced archive truncation method that guarantees the preservation of boundary solutions.

– **Pareto Archived Evolution Strategy (PAES) Algorithm**

This algorithm is introduced by Knowles and Corne in [34]. PAES consists of a 1 + 1 evolution strategy (i.e., a single parent that generates a single offspring) in combination with a historical archive that records the non-dominated solutions previously found. This archive is used as a reference set against which each mutated

individual is being compared. Such a historical archive is the elitist mechanism adopted in PAES. A special feature of this algorithm is the procedure used to maintain diversity which consists of a crowding procedure that divides objective space in a recursive manner. Each solution is placed in a certain grid location based on the values of its objectives (which are used as its “coordinates” or “geographical location”). A map of such grid is maintained, indicating the number of solutions that reside in each grid location. Since the procedure is adaptive, no extra parameters are required (except for the number of divisions of the objective space). This adaptive grid (or variations of it) has been adopted by several modern EMOO algorithms [10].

– **Pareto Envelope based Selection Algorithm (PESA)**

Corne et al. [13] suggested an algorithm known as PESA, which combines the good aspects of SPEA and PAES. Like SPEA, PESA carries two populations (a smaller EA population and a larger archive population). Non-domination and the PAES crowding concept is used to update the archive with the newly created child solutions.

In an extended version of PESA (see [12]), instead of applying the selection procedure on population members, hyperboxes in the objective space are selected based on the number of solutions residing in the hyperboxes. After hyperboxes are selected, a random solution from the chosen hyperboxes is kept. This region-based selection procedure has shown to perform better than the individual-based selection procedure of PESA. In some sense the PESA2 selection scheme is similar in concept to ε -dominance (see [36]), in which predefined ε values determine the hyperbox dimensions. Other ε -dominance based EMOO procedures (see [15]) have shown computationally faster and better distributed solutions than NSGA-II or SPEA2.

– **Bio-inspired and evolution-based heuristic algorithms**

There exist also many bio-inspired heuristics for multi-objective optimization (see [4]) and different evolution-based EMOO algorithms (see [2]). The most important among them are the particle swarm optimization and differential evolution [39], whose use has become increasingly popular in multi-objective optimization [1, 7, 8, 40]. However, other bio-inspired algorithms such as artificial immune systems and ant colony optimization have also been used to solve multi-objective optimization problems [5, 24, 25, 37].

IV. Convergence properties of some EMOO algorithms

In the context of investigations on convergence to the Pareto-optimal front, some authors (see [44, 52]) have considered the distance of a given set to the Pareto-optimal set for finitely large search spaces. Related works treating continuous search spaces are [43, 26]. The distribution was not taken into account, because the focus was not on this matter. However, in comparative studies, the distance alone is not sufficient for performance evaluation, since extremely differently distributed fronts may have the same distance to the Pareto-optimal front.

Two complementary metrics of performance were presented in [56, 57]. The second metric is in some way similar to the comparison methodology proposed in [21]. In summary, it may be said, that the performance metrics are hard to define

and it probably will not be possible to define a single metric that allows for all criteria in a meaningful way. Along with that problem, the statistical interpretation associated with a performance comparison is rather difficult and still needs to be answered, since multiple significance tests are involved, and thus, tools from analysis of variance may be required. In [55] a simulation study was performed on six EMOO algorithms: SPEA, NSGA, VEGA, HLGA, NPGA, FFGA. It was found that there is a clear performance gap between SPEA and NSGA, as well between NSGA and the remaining algorithms, but the fronts achieved by VEGA, HLGA, NPGA and FFGA are rather close together. The results indicated that VEGA might be slightly superior to the last three algorithms, while NPGA achieves fronts closer to the global optimum than FFGA. It seems that VEGA and HLGA have difficulties evolving well-distributed trade-off fronts on the nonconvex function. Another result was that the elitism is an important factor in evolutionary multi-objective optimization. On one hand SPEA clearly outperformed all other considered algorithms and this was the only method among all considered, that incorporates elitism as a central part of the algorithm. On the other hand, the performance of the other algorithms improved significantly when SPEA's elitist strategy was included. Some results indicated that NSGA with elitism equals the performance of SPEA.

However, it should be mentioned that in certain situations, e.g., when preference information is included in the fitness assignment process and the preferences change over time, elitism may have its drawbacks.

In [42] the number of objectives is considered as a convergence factor. The results showed that the performance of multiobjective evolutionary algorithms, such as NSGA-II and SPEA2, deteriorates substantially as the number of objectives increases. NSGA-II, for example, did not even converge for problems with six or more objectives.

In [41] different performance indices were investigated. The conclusion was that the researchers should be very careful in evaluating and comparing the EMOO algorithms according to performance indices only. This is particularly important, when there is little information about the shape of the true Pareto-optimal front, and this is the case in most real-world applications.

V. Current and future challenges

Probably the number of evolutionary metaheuristics designed to solve multi-objective problems will increase further in the next decade. As noted in [9, 11] a lot of significant methods have been proposed in literature since the pioneering work of Shaffer [45, 46].

Janssens [29] pointed out some current and future challenges:

- Usually the decision makers want a small set of solutions to make a choice among them. The challenge is to provide them with a set, as small as possible, that represents the whole set of choices, but to compute this set in an efficient way.

- The Pareto-curve should be presented to the decision maker, who can select a solution lying on this curve according to her or his preferences. The problem here is that the solutions number is typically exponential for discrete problems or is of infinite size for continuous problems. It is often difficult to construct the full Pareto-curve, so that an approximation of the curve is required.

– Some heuristics have been proposed to construct an approximation of the curve, but their performance and complexity remain still unexplored. Some research in this area has been done introducing the concept of ε -Pareto set, a set P_ε of solutions, which approximately dominate every other solution s . The set P_ε contains a solution s' that is within a factor $1 + \varepsilon$ of s , or better, in all the objectives. Such kind of approximation has been studied in [27, 50, 53].

– The evaluation of metaheuristics is still under discussion. Usually metaheuristics are evaluated according to two criteria: computational effort and quality of the solutions. Quantitative measures instead of graphical visualization should be used in many evolutionary algorithms to produce an evaluation. Standards for evaluation have to be developed and more theoretical work regarding quality evaluation has to be done (see [33]).

– Solving many multiobjective optimization problems it becomes clear, that it is impractical or unnecessary to store all non-dominated solutions obtained during the search process. Many algorithms are designed to store an “archive” (i.e. a subset of the discovered non-dominated points). It is interesting to study the computational complexity of maintaining the archive. It is proposed that the archive should be of a bounded and modest size (see [32]).

VI. Conclusions

This paper has presented a short review of algorithms in the fast growing area of evolutionary multiobjective optimization. The EMOO procedures are designed to achieve two goals: (i) convergence to as close to the Pareto-optimal front as possible and (ii) maintenance of a well distributed set of trade-off solutions.

The early research efforts in this area were focused on finding a set of well-converged and well distributed near-optimal trade-off solutions. At the next stage the EMOO researchers concentrated on developing better and computationally faster algorithms by means of scalable test problems [51] and adequate performance metrics to evaluate EMOO algorithms. One of the major aspects of scientific research is the efficiency, which is regarded at algorithmic level and at data structure level [30, 31]. A variety of measures for implementation quality are suggested, allowing a quantitative (rather than only qualitative), comparison of results [20, 55, 56]. Zitzler et al. [55] stated that when assessing performance of an EMOO algorithm, one was interested in measuring three things:

- Maximize the number of elements of the Pareto optimal set found.
- Minimize the distance of the Pareto front produced by the algorithm with respect to the global Pareto front (assuming we know its location).
- Maximize the spread of solutions found, so that we can have a distribution of vectors as smooth and uniform as possible.

Concurrently with the research on performance measures, other researchers were designing test functions. K. Deb proposed a methodology to design multiobjective problems that is widely used [19]. Later on, an alternative set of test functions was proposed, but this time, due to their characteristics, no enumerative process was required to generate their true Pareto front [16, 14]. These test functions are also scalable, their use has become spread. Researchers in the field

normally validate their EMOO algorithms with problems having three or more objective functions, and ten or more decision variables.

As noted in [47] some important research directions remain still unexplored. There is a need for detailed studies of the different aspects involved in the parallelization of EMOO techniques (e.g., load balancing, impact on Pareto convergence, performance issues, etc.), including new algorithms that are more suitable for parallelization than those currently in use. It will be also very interesting to study, for example, the structure of fitness landscapes in multiobjective problems. Such study could provide some insights regarding the sort of problems that are particularly difficult for evolutionary algorithms and could also provide clues with respect to the design of more powerful EMOO techniques.

References

1. A b b a s s, H. A., R. S a r k e r. The Pareto Differential Evolution Algorithm. – International Journal on Artificial Intelligence Tools, Vol. **II**, 2002, No 4, 531-552.
2. B a b u, B. V., M. M. L. J e h a n. Differential Evolution for Multiobjective Optimization. – In: Proc. of the Congress on Evolutionary Computation (CEC'2003), Canberra, Australia, Vol. **4**, December 2003, IEEE Computer Society Press, Los Alamitos, 2696-2703.
3. B r a n k e, J., K. D e b, K. M i e t t i n e n, R. S ł o w i ń s k i. Multiobjective Optimization. Interactive and Evolutionary Approaches. Berlin, Heidelberg, Springer Verlag, 2008.
4. C o e l l o, C. A. C. The EMOO Repository: A Resource for Doing Research in Evolutionary Multiobjective Optimization. – IEEE Computational Intelligence Magazine, **45**, February 2006, 37-38.
5. C o e l l o, C. A. C., N. C r u z C o r t e s. Solving Multiobjective Optimization Problems Using an Artificial Immune System. – Genetic Programming and Evolvable Machines. Vol. **2**, June 2005, 163-190.
6. C o e l l o, C. A. C., G. B. L a m o n t. Applications of Multiobjective Evolutionary Algorithms. Singapore, World Scientific, 2004.
7. C o e l l o, C. A. C., G. T o s c a n o P u l i d o, M. S a l a z a r L e c h u g a. Handling Multiple Objectives with Particle Swarm Optimization. – In: IEEE Trans. Evolutionary Computation, Vol. **8**, June 2004, No 3, 256-279.
8. C o e l l o, C. A. C., M. S. L e c h u g a. MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization. – In: Congress on Evolutionary Computation (CEC'2002), Vol. **2**, May 2002, IEEE Service Center, Piscataway, 1051-1056.
9. C o e l l o, C. A. C., D. A. V a n V e l d h u i z e n, G. B. L a m o n t. Evolutionary Algorithms for Solving Multiobjective Problems. Kluwer Academic Publishers, 2002.
10. C o e l l o, C. A. C., G. T o s c a n o P u l i d o. Multiobjective Optimization Using a Micro-Genetic Algorithm. – In Proc. of the Genetic and Evolutionary Computation Conference, (GECCO'2001), San Francisco, California, Morgan Kaufmann Publishers (Lee Spector et al., Eds.), 2001, 274-282.
11. C o e l l o, C. A. C. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. – Knowledge and Information Systems. – An International Journal, **1**, 1999, No 3, 269-308.
12. C o r n e, D. W., N. R. J e r r a m, J. D. K n o w l e s, M. J. O a t e s. PESA-II: Region-Based Selection in Evolutionary Multiobjective Optimization. – In: Proc. of the Genetic and Evolutionary Computation Conference (GECCO'2001), San Francisco, California, Morgan Kaufmann Publishers (Lee Spector et al., Eds.), 2001, 283-290.
13. C o r n e, D. W., J. D. K n o w l e s, M. J. O a t e s. The Pareto Envelope-Based Selection Algorithm for Multi-Objective Optimization. – In: PPSN 2000. LNCS, Vol. **1917**, Springer, Heidelberg (K. Deb, et al., Eds.), 2000, 839-848.

14. Deb, K., Thiele, M., Laumanns, E., Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. – In: Evolutionary Multiobjective Optimization (Ajith Abraham, et al. Eds.). Theoretical Advances and Applications, USA, Springer, 2005, 105-145.
15. Deb, K., Mohan, S., Mishra. Towards a Quick Computation of Well-Spread Pareto-Optimal Solutions. – In: EMO 2003. LNCS, Vol. **2632** (C. M. Fonseca et al., Eds.). Heidelberg, Springer, 222-236.
16. Deb, K., Thiele, M., Laumanns, E., Zitzler. Scalable Multi-Objective Optimization Test Problems. – In: Congress on Evolutionary Computation (CEC'2002), Vol. **1**, Piscataway. New Jersey, May 2002. IEEE Service Center, 325-830.
17. Deb, K., Pratap, S., Agarwal, T., Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. – IEEE Transactions on Evolutionary Computation, Vol. **6**, Apr. 2002, No 2, 182-197.
18. Deb, K. Multi-Objective Optimization Using Evolutionary Algorithms. Wiley-Interscience Series in Systems and Optimization. Chichester, John Wiley & Sons, 2001.
19. Deb, K. Multiobjective Genetic Algorithms. Problem Difficulties and Construction of Test Problems. – Evolutionary Computation, Vol. **7**, 1999, No 3, 205-230.
20. Fonseca, C. M., Fleming. On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers. – In: Parallel Problem Solving from Nature – PPSN IV, Lecture Notes in Computer Science (Hans-Michael Voigt et al., Eds.), Berlin, Germany, Springer Verlag, September 1996, 584-593.
21. Fonseca, C. M., Fleming. An Overview of Evolutionary Algorithms in Multiobjective Optimization. – Evolutionary Computation, **3**, 1995, No 1, 1-16.
22. Fonseca, C. M., Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. – In: Proc. of the 5th International Conference on Genetic Algorithms, San Mateo, California, University of Illinois at Urbana-Champaign (Stephanie Forrest, Ed.), Morgan Kaufmann Publishers, 1993, 416-423.
23. Goldberg, D. E. Genetic Algorithms in Search, Optimization and Machine Learning. Reading, Mass, Addison Wesley, 1989.
24. Gravel, M., W. L. Price, C. Gagné. Scheduling Continuous Casting of Aluminium Using a Multiple Objective ant Colony Optimization Metaheuristic. – European Journal of Operational Research, **143**, 2002, No 1, 218-229.
25. Gutsch, M. Ant Algorithms in Stochastic and Multicriteria Environments. Ph. D. Thesis, Department of Economics and Business Engineering, University of Karlsruhe, Germany, 2004.
26. Hanne, T. On the Convergence of Multiobjective Evolutionary Algorithms. – European Journal of Operational Research, **117**, 1999, No 3, 553-564.
27. Hansén, P. Bicriterion Path Problems. – In: Proc. of 3rd Conference Multiple Criteria Decision Making Theory and Applications. Lecture Notes in Economics and Mathematical Systems, **177**, 1979, 109-127.
28. Horn, J., Nafpliotis, D. E. Goldberg. A Niche Pareto Genetic Algorithm for Multiobjective Optimization. – In: Proc. of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Vol. **I**, Piscataway, New Jersey, IEEE Service Center, June 1994, 82-87.
29. Janssens, G. K. Recent Challenges in the Use of Evolutionary Algorithms for Multi-Objective Optimization. – Proc. of the International Conference on Information Technologies (Info-Tech-2008), 19-20 September 2008, Bulgaria, 51-60.
30. Jenesen, M. T. Reducing the Run-Time Complexity of Multiobjective EAs: The NSGA-II and Other Algorithms. – IEEE Transactions on Evolutionary Computation, Vol. **7**, October 2003, No 5, 503-515.
31. Knowles, J., Hughes. Multiobjective Optimization on a Budget of 250 Evaluations. – In: Evolutionary Multi-Criterion Optimization. 3rd International Conference (Carlos A. Coello Coello et al., Eds.), EMO 2005, Guanajuato, Mexico, Springer. Lecture Notes in Computer Science, Vol. **3410**. March 2005, 176-190.

32. K n o w l e s, J. D., D. W. C o r n e. Bounded Pareto Archiving: Theory and Practice. – In: *Metaheuristics for Multiobjective Optimisation* (X. Gandibleux, et al., Eds.). Lecture Notes in Economics and Mathematical Systems, **535**, Berlin, Springer Verlag, 2004, 39-64.
33. K n o w l e s, J. D., D. W. C o r n e. On Metrics for Comparing Non-Dominated Sets. – In: *Proc. of the Congress on Evolutionary Computation (CEC 2002)*, IEEE Press, NJ, Piscataway, 2002, 711-716.
34. K n o w l e s, J. D., D. W. C o r n e. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. – *Evolutionary Computation*, Vol. **8**, 2000, No 2, 149-172.
35. K u h n, H. W., A. W. T u c k e r. Nonlinear Programming. – In: *Proc. of the Second Berkeley Symposium on Mathematical Statistics and Probability* (J. Neyman, Ed.), Berkeley, California, University of California Press, 1951, 481-492.
36. L a u m a n n s, M., L. T h i e l e, K. D e b, E. Z i t z l e r. Combining Convergence and Diversity in Evolutionary Multi-Objective Optimization. – *Evolutionary Computation*, Vol. **10**, 2002, No 3, 263-282.
37. M c M u l l e n, P. R. An Ant Colony Optimization Approach to Addressing a JIT Sequencing Problem with Multiple Objectives. – *Artificial Intelligence in Engineering*, **15**, 2001, 309-317.
38. M i e t t i n e n, K., P. N e i t t a a n m ä k i, M. M. M ä k e l ä, J. P é r i a u x. *Evolutionary Algorithms in Engineering and Computer Science*. Chichester, Weinheim, New York, Brisbane, Singapore, Toronto, John Wiley & Sons, Ltd., 1999.
39. M o l i n a - C r i s t o b a l, A., L. A. G r i f f i n, P. J. F l e m i n g, D. H. O w e n s. Multiobjective Controller Design: Optimising Controller Structure with Genetic Algorithms. – In: *Proc. of the IFAC World Congress on Automatic Control*, Prague, Czech Republic, July 2005.
40. M o s t a g h i m, S., J. T e i c h. Strategies for Finding Good Local Guides in Multi-Objective Particle Swarm Optimization (MOPSO). – In: *2003 IEEE Swarm Intelligence Symposium Proc.*, Indianapolis, Indiana, USA, April 2003, IEEE Computer Society Press, Los Alamitos, 26-33.
41. O k a b e, T., Y. J i n, B. S e n d h o f f. A Critical Survey of Performance Indices for Multi-Objective Optimisation. – In: *Evolutionary Computation*, 2003, CEC 2003, Vol. **2**, Issue 8-12 December 2003, 878-885.
42. P r a d i t w o n g, K., X i n Y a o. How Well Do Multi-Objective Evolutionary Algorithms Scale to Large Problems. – In: *Evolutionary Computation*, 2007. CEC 2007. IEEE Congress on Volume, Issue September 25-28 2007, 3959-3966.
43. R u d o l p h, G., A. A g a p i e. Convergence Properties of Some Multi-Objective Evolutionary Algorithms. – In: *Proc. of the Conference on Evolutionary Computation*, Vol. **2**, IEEE Press, 2000, 1010-1016.
44. R u d o l p h, G. On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set. – In: *IEEE International Conference on Evolutionary Computation (ICEC'98)*, IEEE Press, Piscataway, New Jersey, 1998, 511-516.
45. S c h a f f e r, J. D. Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms. Ph. D. Thesis, Vanderbilt University, Nashville, TN, 1984.
46. S c h a f f e r, J. D. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. – In: *Genetic Algorithms and Their Applications*. Proc. of the First International Conference on Genetic Algorithms, Lawrence Erlbaum, 1985, 93-100.
47. S a r k e r, R., M. M o h a m m a d i a n, X. Y a o. *Evolutionary Optimization*. Kluwer Academic Publishers, 2002.
48. S r i n i v a s, N., K. D e b. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. – *Evolutionary Computation*, Vol. **2**, 1994, No 3, 221-248.
49. T a n a k a, M., T. T a n i n o. Global Optimization by the Genetic Algorithm in a Multi-Objective Decision Support System. – In: *Proc. of the 10th International Conference on Multiple Criteria Decision Making*, Vol. **2**, 1992, 261-270.
50. V a s i l v i t s k i i, S., M. Y a n n a k a k i s. Efficiently Computing Succinct Trade-off Curves. – *Theoretical Computer Science*, **348**, 2005, No 2, 334-356.

51. V e l d h u i z e n, D. A. V., G. B. L a m o n t. Multiobjective Evolutionary Algorithm Test Suites. – In: Proc. of the 1999 ACM Symposium on Applied Computing, (J. Carroll, et al., Eds.), San Antonio, Texas. ACM, 351-357.
52. V e l d h u i z e n, D. A. V., G. B. L a m o n t. Evolutionary Computation and Convergence to a Pareto Front. – In: Genetic Programming 1998. Proc. of the Third Annual Conference, Morgan Kaufmann, San Francisco, California (J. R. Koza et al., Eds.), 1998, 22-25.
53. W a r b u r t o n, A. Approximation of Pareto Optima in Multiobjective Shortest Path Problems. – Operations Research, **35**, 1987, No 1, 70-79.
54. Z i t z l e r, E., M. L a u m a n n s, L. T h i e l e. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. – In: EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, Athens, Greece (K. Giannakoglou et al., Eds.), 2002, 95-100.
55. Z i t z l e r, E., K. D e b, L. T h i e l e. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. – Evolutionary Computation, Vol. **8**, Summer 2000, No 2, 173-195.
56. Z i t z l e r, E., L. T h i e l e. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. – In: IEEE Transactions on Evolutionary Computation, Vol. **3**, November 1999, No 4, 257-271.
57. Z i t z l e r, E., L. T h i e l e. Multiobjective Optimization Using Evolutionary Algorithms – A Comparative Case Study. – In: 5th International Conference on Parallel Problem Solving from Nature (PPSN-V), Springer, Berlin, Germany (A. E. Eiben et al., Eds.), 1998, 292-301.

Обзор эволюционных алгоритмов в многокритериальной оптимизации

Васил Гуляшки, Христо Тошев, Чавдар Корсемов

Институт информационных технологий, 1113 София

E-mails: vggul@yahoo.com toshev@iinf.bas.bg chkorsemov@iinf.bas.bg

(Резюме)

В работе рассматриваются эволюционные подходы и алгоритмы при решении задач многокритериальной оптимизации. Обсуждены сравнения некоторых из самых фундаментальных алгоритмов последних двух десятилетий, а так же и сходимость эволюционных алгоритмов в многокритериальной оптимизации. Рассмотрены некоторые направления для будущих исследованиях. Сделаны выводы об этой научной области.