EVOLUTIONARY METHODS FOR DESIGN, OPTIMISATION AND CONTROL
K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou and T. Fogarty (Eds.)

EVOLUTIONARY ALGORITHMS FOR MULTIOBJECTIVE OPTIMIZATION

Eckart Zitzler
Computer Engineering and Networks Laboratory (TIK)
Department of Information Technology and Electrical Engineering
Swiss Federal Institute of Technology (ETH) Zurich
Gloriastr. 35, CH-8092 Zurich, Switzerland
e-mail: zitzler@tik.ee.ethz.ch
web page: http://www.tik.ee.ethz.ch/~zitzler

Abstract. Multiple, often conflicting objectives arise naturally in most real-world optimization scenarios. As evolutionary algorithms possess several characteristics due to which they are well suited to this type of problem, evolution-based methods have been used for multiobjective optimization for more than a decade. Meanwhile evolutionary multiobjective optimization has become established as a separate subdiscipline combining the fields of evolutionary computation and classical multiple criteria decision making.

In this paper, the basic principles of evolutionary multiobjective optimization are discussed from an algorithm design perspective. The focus is on the major issues such as fitness assignment, diversity preservation, and elitism in general rather than on particular algorithms. Different techniques to implement these strongly related concepts will be discussed, and further important aspects such as constraint handling and preference articulation are treated as well. Finally, two applications will presented and some recent trends in the field will be outlined.

Key words: evolutionary algorithms, multiobjective optimization

1 INTRODUCTION

Almost every real-world problem involves simultaneous optimization of several incommensurable and often competing objectives such as performance and cost. If we consider only one of these objectives, the optimal solution is clearly defined as the search space is totally ordered: a solution is either faster resp. cheaper than another or not. The situation changes if we try to optimize all objectives at the same time. Then, the search space is only partially ordered, and two solutions can be indifferent to each other (one is cheap and slow while the other one provides maximum performance at maximum cost). As a consequence, there is usually not a single optimum but rather a set of optimal trade-offs, which in turn contains the single-objective optima.

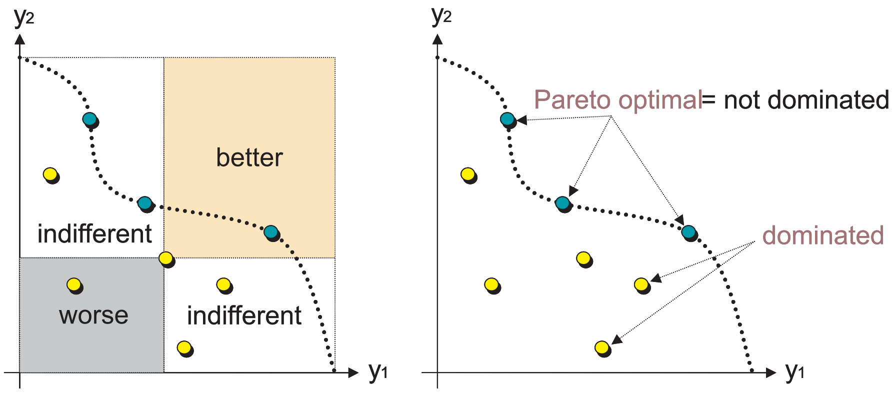

Figure 1: Illustration of the concept of Pareto optimality

This makes clear that the optimization of multiple objectives adds a further level of complexity compared to the single-objective case. In other words, single-objective optimization can be considered a special case of multiobjective optimization (and not vice versa). In this paper, current techniques will be presented which have been developed to deal with this additional complexity. The focus is on the basic principles of evolutionary multiobjective optimization rather than on specific algorithms.

2 FUNDAMENTAL CONCEPTS

In general, a multiobjective optimization problem is defined by a function f which maps a vector of decision variables, the so-called decision vector, to a vector of objective values, the so-called objective vector:

$$(y_1, y_2, \ldots, y_n) = f(x_1, x_2, \ldots, x_n)$$

Without loss of generality, it is assumed here and in the following that each of the n components of the objective vector is to be maximized.

In this scenario, a solution (defined by the corresponding decision vector) can be better, worse, equal, but also indifferent to another solution with respect to the objective values (cf. Fig 1 on the left hand side). "Better" means a solution is not worse in any objective and at least better in one objective than another; the superior solution is also said to *dominate* the inferior one. Using this concept one can define what an optimal solution is: a solution which is not dominated by any other solution in the search space. Such a solution is called *Pareto optimal*, and the entire set of optimal trade-offs is called the Pareto-optimal set, which is represented by the dotted line in Fig 1.

The concept of Pareto optimality is only the first step in solving a multiobjective optimization problem because at the end a single solution is sought. Therefore a decision making process is necessary in which preference information is used in order to select an appropriate trade-off. Although there are different ways of integrating this process, in the field of evolutionary multiobjective optimization it is usually assumed that optimization takes places before decision making. That is the goal is to find or approximate the Pareto-optimal set. In the remainder of this paper, this view will be adopted without implying that this is the only or best way to approach a multiobjective optimization problem.

3 BASIC ISSUES IN ALGORITHM DESIGN

The goal of approximating the Pareto-optimal front is itself multiobjective: on the one hand, the distance to the Pareto set is to be minimized, on the other hand, the achieved nondominated set should be as diverse as possible. The first objective is related to the problem of assigning scalar fitness values in the presence of multiple optimization criteria. The second objective raises the question of how to preserve diversity within the nondominated set. Finally, a third issue which addresses both of the above objectives is elitism, i.e., the question of how to prevent nondominated solutions from being lost.

In the following each of these issues will be discussed: fitness assignment, diversity preservation, and elitism. Remarkably, they are well reflected by the development of the field of evolutionary multiobjective optimization. While the first studies on multiobjective evolutionary algorithms (MOEAs) were mainly concerned with the problem of guiding the search towards the Pareto-optimal set,[1–3] all approaches of the second generation incorporated in addition a niching concept in order to address the diversity issue.[4–6] The importance of elitism was recognized and supported experimentally in the late nineties,[7,8] and most of the third generation MOEAs implement this concept in different ways.[9,10]

3.1 Fitness Assignment

In contrast to single-objective optimization, where objective function and fitness function are often identical, both fitness assignment and selection must allow for several objectives with multi-criteria optimization problems. In general, one can distinguish aggregation-based, criterion-based, and Pareto-based fitness assignment strategies.

One approach which is built on the traditional techniques for generating trade-off surfaces is to aggregate the objectives into a single parameterized objective function. The parameters of this function are systematically varied during the optimization run in order to find a set of nondominated solutions instead of a single trade-off. For instance, some MOEAs use weighted-sum aggregation, where the weights represent the parameters which are changed during the evolution process.[11,12]

Criterion-based methods switch between the objectives during the selection phase. Each time an individual is chosen for reproduction, potentially a different objective will decide which member of the population will be copied into the mating pool. For example, Schaffer[1] proposed filling equal portions of the mating pool according to the distinct objectives, while Kursawe[3] suggested assigning a probability to each objective which determines whether the objective will be the sorting criterion in the next selection step—the probabilities can be user-defined or chosen randomly over time.

The idea of calculating an individual's fitness on the basis of Pareto dominance goes back to Goldberg,[13] and different ways of exploiting the partial order on the population have been proposed. Some approaches use the dominance rank, i.e., the number of individuals by which an individual is dominated, to determine the fitness values.[4] Others make use of the dominance depth; here, the population is divided into several fronts and the depth reflects to which front an individual belongs to.[5] Alternatively, also the dominance count, i.e., the number of individuals dominated by a certain individual, can be taken into account. For instance, SPEA[9] and SPEA2[14] assign fitness values on the basis of both dominance rank and count.

Independent of the technique used, the fitness is related to the whole population in contrast to aggregation-based methods which calculate an individual's raw fitness value independently of other individuals.

3.2 Diversity Preservation

Most MOEAs try to maintain diversity along the current approximation of the Pareto set by incorporating density information into the selection process: an individual's chance of being selected is decreased the greater the density of individuals in its neighborhood. This issue is closely related to the estimation of probability density functions in statistics, and the methods used in MOEAs can be classified according to the categories for techniques in statistical density estimation.[15]

Kernel methods[15] define the neighborhood of a point in terms of a so-called Kernel function K which takes the distance to another point as an argument. In practice, for each individual the distances d_i to all other individuals i are calculated and after applying K the resulting values $K(d_i)$ are summed up. The sum of the K function values represents the density estimate for the individual. Fitness sharing is the most popular technique of this type within the field of evolutionary computation, which is used, e.g., in MOGA,[4] NSGA,[5] and NPGA.[6]

Nearest neighbor techniques[15] take the distance of a given point to its kth nearest neighbor into account in order to estimate the density in its neighborhood. Usually, the estimator is a function of the inverse of this distance. SPEA2,[14] for instance, calculates for each individual the distance to the kth nearest individual and adds the reciprocal value to the raw fitness value (fitness is to be minimized).

Histograms[15] define a third category of density estimators that use a hypergrid to define neighborhoods within the space. The density around an individual is simply estimated by the number of individuals in the same box of the grid. The hypergrid can be fixed, though usually it is adapted with regard to the current population as, e.g., in PAES.[10]

Due to space-limitations, a discussion of pros and cons of the various methods cannot be provided here—the interested reader is referred to Silverman's book.[15] Furthermore, note that all of the above methods require a distance measure which can be defined on the genotype, on the phenotype with respect to the decision space, or on the phenotype with respect to the objective space. Most approaches consider the distance between two individuals as the distance between the corresponding objective vectors.

3.3 Archiving Strategies

Although fitness assignment and diversity preservation techniques aim at guiding the population towards the Pareto-optimal set, still good solutions may be lost during the optimization process due to random effects. A common way to deal with this problem is to maintain a secondary population, the so-called archive, to which promising solutions in the population are copied at each generation. The archive may just be used as an external storage separate from the optimization engine or may be integrated into the EA by including archive members in the selection process.

Usually the size of the archive is restricted due to memory but also run-time limitations. Therefore, criteria have to be defined on this basis of which the solutions to be kept in the archive are selected. The dominance criterion is most commonly used, i.e., dominated archive members are removed and the archive comprises only the current approximation of the Pareto set. However, as this criterion is in general not sufficient (e.g., for continuous problems the Pareto set may contain an infinite number of solutions), additional information is taken into account to reduce the number of archive members further. Examples are density information[9,10] and the time that has been passed since the individual entered the archive.[16]

Most elitist MOEAs make use of a combination of dominance and density to choose the individuals that will be kept in the archive at every generation. However, these approaches may suffer from the problem of deterioration, i.e., solutions contained in the archive at generation t may be dominated by solutions that were members of the archive at any generation $t' < t$ and were discarded later. Recently, Laumanns et al.[17] presented an archiving strategy which avoids this problem and guarantees to maintain a diverse set of Pareto-optimal solutions (provided that the optimization algorithm is able to generate the Pareto-optimal solutions).

It should be mentioned that not all elitist MOEAs explicitly incorporate an archive, e.g., NSGA-II.[18] However, the basic principle is the same: during environmental selection special care is taken to not loose nondominated solutions.

4 ADVANCED DESIGN TOPICS

Besides the three fundamental design issues, two other topics will be briefly discussed here: constraint handling and preference articulation.

In evolutionary single-objective optimization, several ways to deal with different types of constraints have been proposed, e.g., the penalty function approach.[13] In principle, these techniques can be used in the presence of multiple criteria as well, but multiobjective optimization offers more flexibility with this respect. One possibility is to convert each of the constraints into a separate objective[19] which have to optimized besides the actual objectives. Alternatively, the constraints can be aggregated, and only one optimization criterion—to minimize the overall constraint violation—is added.[20] Both methods have the advantage that no modifications are necessary concerning the underlying MOEA. On the other hand, infeasible individuals which provide good values regarding the actual objectives are treated equally in comparison to feasible individuals with worse objective values, which in turn may slow down the convergence speed towards the feasible region. More sophisticated methods distinguish between feasible and infeasible solutions. Fonseca and Fleming,[21] for instance, suggested to handle each constraint as a distinct objective as above and to extend the definition of Pareto dominance in order to favor feasible over infeasible individuals. If two feasible solutions are checked for dominance, one is better than the other if it dominates the competitor with respect to the actual objectives. The same holds if both solutions are infeasible, however, dominance is then restricted to the constraint objectives only. Finally, feasible solutions are defined to dominate infeasible ones. This methods increases the selection pressure towards the feasible set (assuming a Pareto-based fitness assignment scheme is used). A slight modification of this approach is to consider the overall constraint violation instead of treating constraints separately;[22] an infeasible solution dominates another infeasible one if its overall constraint violation is lower.

Constraints represent one way of including existing knowledge about the application into the optimization process in order to focus on promising regions of the search space. Moreover, other types of preference information such as goals and objective rankings may be available which help to guide the search towards interesting regions

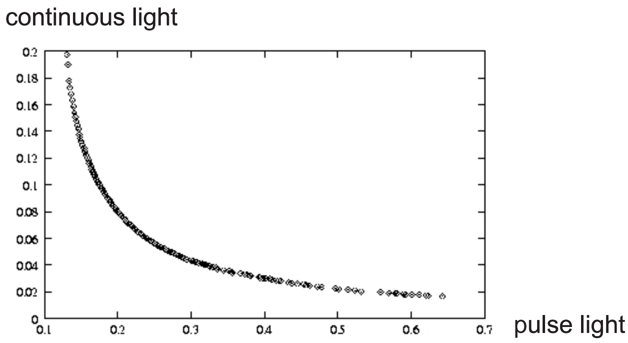

Figure 2: Trade-off front obtained for the model parameter optimization problem

of the Pareto-optimal set. For instance, Fonseca's and Fleming's extended definition of Pareto dominance allows in addition to the constraints to also include goals and priorities.[21] Another approach is to change the definition of Pareto dominance by, roughly speaking, considering general pointed convex cones instead of translated nonnegative orthants that represent the dominated area of a given solution.[23]

5 APPLICATIONS

Since almost every real-world optimization problem involves several objectives, there are numerous applications for which tools are needed that are able to approximate the Pareto-optimal set. Many studies demonstrate the usefulness of MOEAs in this context.[24] However, multiobjective optimization can even be beneficial for applications which at the first glance seem to be single-objective. In the following two examples will be given.

Bleuler et al.[25] presented a multiobjective approach to evolve compact programs and to reduce the effects caused by bloating in Genetic Programming (GP). As it is well known that trees tend to grow rapidly during a GP run, several methods have been suggested to avoid this phenomenon.[26] However, those techniques which incorporate the tree size in the optimization problem, e.g., as a constraint or by a weighted sum, usually are still single-objective. In contrast, the proposed technique considers the program size as a second, independent objective besides the program functionality. In combination with SPEA2,[14] this method was shown to outperform four other strategies to reduce bloat with regard to both convergence speed and size of the produced programs on a even-parity problem.

Another example is the fitting of a biochemical model. Hennig et al.[27,28] investigated the dynamics of a particular photoreceptor in Arabidopsis plant cells. They first grew Arabidopsis cells in darkness and then performed two types of experiments: one half of the seedlings were exposed to continuous light and the other half was exposed to pulse light. Afterwards, the experimental data were used to fit the parameters of a given model for the photoreceptor dynamics. At the Computer Engineering Laboratory at ETH Zurich, a multiobjective optimization was carried out that aimed at minimizing the deviations between the data predicted by the model and the experimental data. Here, for each of the two experiments a separate objective was introduced. The results are depicted in Figure 2. Interestingly, a trade-off front emerges, which indicates that there is no parameter setting for the model such that it explains well both scenarios under consideration at the same time. Independent of what conclusions can be drawn from this result, the fact that there is a trade-off offers valuable information to the biologists. This application demonstrates that multiobjective optimization can provide new insights about the problem—insights which would not have been gained in a pure single-objective approach.

6 CONCLUSIONS

This paper is an attempt to identify common concepts and general building blocks used in evolutionary multiobjective optimization. All of these techniques have advantages and disadvantages, and therefore the selection of the techniques integrated in an MOEA strongly depends on the problem to be solved.

Despite the variety of available methods, the field of multiobjective evolutionary computation is still quite young and there are many open research problems. Promising directions for future research might be: higher dimensional problems (more than two objectives), statistical frameworks for performance comparisons of MOEAs, interactive optimization which integrates the decision maker, comparison of evolutionary with non-evolutionary approaches, and theoretical studies which provide new insights into the behavior of MOEAs, to name only a few.

REFERENCES

[1] J. David Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In John J. Grefenstette, editor, *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, pages 93–100, Pittsburgh, PA. (1985). sponsored by Texas Instruments and U.S. Navy Center for Applied Research in Artificial Intelligence (NCARAI).

[2] Michael P. Fourman. Compaction of symbolic layout using genetic algorithms. In John J. Grefenstette, editor, *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, pages 141–153, Pittsburgh, PA. (1985). sponsored by Texas Instruments and U.S. Navy Center for Applied Research in Artificial Intelligence (NCARAI).

[3] Frank Kursawe. A variant of evolution strategies for vector optimization. In H.-P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature*, pages 193–197, Berlin. Springer, (1991).

[4] Carlos M. Fonseca and Peter J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California. Morgan Kaufmann, (1993).

[5] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, **2**(3), 221–248, (1994).

[6] Jeffrey Horn, Nicholas Nafpliotis, and David E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Computation*, volume 1, pages 82–87, Piscataway, NJ. IEEE Press, (1994).

[7] G. T. Parks and I. Miller. Selective breeding in a multiobjective genetic algorithm. In A. E. Eiben et al., editors, *Parallel Problem Solving from Nature – PPSN V*, pages 250–259, Berlin. Springer, (1998).

[8] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, **8**(2), 173–195, (2000).

[9] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, **3**(4), 257–271, (1999).

[10] J. D. Knowles and D. W. Corne. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In *Congress on Evolutionary Computation (CEC99)*, volume 1, pages 98–105, Piscataway, NJ. IEEE Press, (1999).

[11] P. Hajela and C.-Y. Lin. Genetic search strategies in multicriterion optimal design. *Structural Optimization*, **4**, 99–107, (1992).

[12] Hisao Ishibuchi and Tadahiko Murata. Multi-objective genetic local search algorithm. In *Proceedings of 1996 IEEE International Conference on Evolutionary Computation (ICEC'96)*, pages 119–124, Piscataway, NJ. IEEE Press, (1996).

[13] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, Massachusetts, (1989).

[14] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001.

[15] B. W. Silverman. *Density estimation for statistics and data analysis.* Chapman and Hall, London, (1986).

[16] G. Rudolph and A. Agapie. Convergence properties of some multi-objective evolutionary algorithms. In *Congress on Evolutionary Computation (CEC 2000)*, volume 2, pages 1010–1016, Piscataway, NJ. IEEE Press, (2000).

[17] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. On the convergence and diversity-preservation properties of multi-objective evolutionary algorithms. Technical Report 108, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001.

[18] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In Marc Schoenauer et al., editors, *Parallel Problem Solving from Nature – PPSN VI*, Berlin. Springer.

[19] Carlos A. Coello Coello. Constraint-handling using an evolutionary multiobjective optimization technique. *Civil Engineering and Environmental Systems*, **17**, 319–346, (2000).

[20] Jonathan Wright and Heather Loosemore. An infeasibility objective for use in constrained pareto optimization. In E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. Corne, editors, *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, volume 1993 of *Lecture Notes in Computer Science*, pages 256–268, Berlin. Springer-Verlag, (2001).

[21] Carlos M. Fonseca and Peter J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms—part i: A unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics*, **28**(1), 26–37, (1998).

[22] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms.* Wiley, Chichester, UK, (2001).

[23] Kaisa Miettinen. *Nonlinear Multiobjective Optimization.* Kluwer, Boston, (1999).

[24] E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. Corne, editors. *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, volume 1993 of *Lecture Notes in Computer Science*, Berlin, Germany, March 2001. Springer-Verlag.

[25] Stefan Bleuler, Martin Brack, Lothar Thiele, and Eckart Zitzler. Multiobjective genetic programming: Reducing bloat by using SPEA2. In *Congress on Evolutionary Computation (CEC-2001)*, pages 536–543, Piscataway, NJ. IEEE, (2001).

[26] Wolfgang Banzhaf, Peter Nordin, Robert E. Keller, and Frank D. Francone. *Genetic Programming — An Introduction.* Morgan Kaufmann, dpunkt, (1998).

[27] L. Hennig, C. Büche, K. Eichenberg, and E. Schäfer. Dynamic properties of endogenous phytochrome A in arabidopsis seedlings. *Plant Physiology*, **121**, 571–577, (1999).

[28] L. Hennig, C. Büche, and E. Schäfer. Degradation of phytochrome A and the high irradiance response in arabidopsis: A kinetic analysis. *Plant, Cell and Environment*, **23**, 727–734, (2000).