Evolutionary Multiobjective Optimization: A Short Survey of the State-of-the-art

M. Pilát

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. Many real-life problems have a natural representation in the framework of multiobjective optimization. Evolutionary algorithms are generally considered one of the most successful methods for solving the multiobjective optimization problems. In this paper we present state-of-the-art multiobjective evolutionary algorithms and briefly discuss their advantages and disadvantages. In the last section we suggest some possibilities for future research in this area.

Introduction

Many real life optimization problems require optimizing more than one objective at once. The methods used in multiobjective optimization deal with this kind of problems.

There are several multiobjective optimization methods, some of them are purely mathematical [Das and Dennis, 1998], others are based on Particle Swarm Optimization [Kennedy and Eberhart, 1995] or Ant Colony Optimization [Dorigo, 1992]. However, multiobjective evolutionary algorithms (MOGA), seem to be the best method used nowadays. One of their main advantages is that they are population based, thus finding more than one interesting solution in a single run. Another advantage is the lack of assumptions about the problem to be solved.

In this paper, we describe some of the most important and widely used MOGAs. It is organized as follows: in the next section, we define the basic terms we will need to talk about multiobjective optimization. Next, we describe the multiobjective evolutionary algorithms and finally, we provide some ideas for future research in this field.

Basic definitions

In multiobjective optimization the goal is to optimize several functions at once; finding a solution which is optimal in all of them. This cannot generally be achieved by a single solution, more often there is a set of solutions and each of them is better in at least one function compared to each other.

The following definition explains formally what a multiobjective optimization problem is.

Definition 1. The *multiobjective optimization problem (MOP)* is a quadruple $\langle D, O, \boldsymbol{f}, C\rangle$, where

- D is the *decision space*
- $O \subseteq \mathbb{R}^n$ is the *objective space*
- $C = \{g_1, \ldots, g_m\}$, where $g_i : D \to \mathbb{R}$ is the set of *constraint functions (constraints)* defining the *feasible space* $\Phi = \{\boldsymbol{x} \in D | g_i(\boldsymbol{x}) \leq 0\}$
- $\boldsymbol{f} : \Phi \to O$ is the vector of n *objective functions (objectives)*, $\boldsymbol{f} = (f_1, \ldots, f_n), f_i : \Phi \to \mathbb{R}$

$\boldsymbol{x} \in D$ is called the *decision vector* and $\boldsymbol{y} \in O$ is denoted as the *objective vector*.

The preference between solutions is given by the following relation. If a solution has a lower value of each objective function than another solution, it is considered to be better. The ordering of solutions is described in the following definition.

Definition 2. Given decision vectors $\boldsymbol{x}$, $\boldsymbol{y} \in D$ we say

- $\mathbf{x}$ *weakly dominates* $\mathbf{y}$ ($\boldsymbol{x} \preceq \boldsymbol{y}$) if $\forall i \in \{1 \ldots n\} : f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y})$.
- $\mathbf{x}$ *and* $\mathbf{y}$ *are incomparable* if neither $\boldsymbol{x} \preceq \boldsymbol{y}$ nor $\boldsymbol{y} \preceq \boldsymbol{x}$.
- $\mathbf{x}$ *does not dominate* $\mathbf{y}$ ($\boldsymbol{x} \not\preceq \boldsymbol{y}$) if $\boldsymbol{y} \preceq \boldsymbol{x}$ or $\boldsymbol{x}$ and $\boldsymbol{y}$ are incomparable

It is easy to show that $\preceq$ define a partial order on the decision space.

Finally, we can formally state the goal of multiobjective optimization: finding the set of solutions which are not dominated by any other decision vector.

Definition 3. The *solution of a MOP* is the *Pareto (optimal) set*

$$P^* = \{\boldsymbol{x} \in \Phi | \forall \boldsymbol{y} \in \Phi : \boldsymbol{y} \not\preceq \boldsymbol{x}\}$$

The projection of P^* under $\boldsymbol{f}$ is called the *Pareto optimal front*.

However, this set is often infinite or very large, thus we seek only a smaller set, which would describe it in a "good" way. We will call this set a Pareto set approximation.

Definition 4. Every $P \subseteq \Phi$ is called *Pareto set approximation*.

Thus, the (practical) goal of the multiobjective optimization is to find a good Pareto set approximation.

We often need to compare two Pareto set approximations and decide which is better. We could extend the dominance relation defined earlier to sets of vectors, however, using this partial ordering, two sets are rarely comparable in practice.

Several methods were proposed during the years, which provide ways to compare the quality of Pareto set approximations. Among them the *hypervolume metric* (or the $\mathcal{S}$ metric) gained popularity and is practically the only one used nowadays. This metric (formally defined bellow) measures the hypervolume of the space dominated by the solutions in the Pareto set approximation.

Definition 5. Let $R \subset O$ be a reference set. The *hypervolume metric* $\mathcal{S}$ is defined as

$$\mathcal{S}(A) = \lambda(H(A, R))$$

where

- $H(A, R) = \{x \in O | \exists \boldsymbol{a} \in A \exists \boldsymbol{r} \in R : \forall i \in \{1, \ldots, n\} : f_i(\boldsymbol{a}) \preceq \boldsymbol{x}_i \preceq \boldsymbol{r}_i\}$ where f_i is the i-th objective function
- λ is Lebesgue measure with $\lambda(H(A, R)) = \int_O \boldsymbol{1}_{H(A,R)}(z)dz$ and $\boldsymbol{1}_{H(A,R)}$ is the characteristic function of the set $H(A, R)$

This metric has some interesting features: it is biased towards the parts of the Pareto front, where the derivative is close to one, which are exactly those regions interesting for the decision maker as they provide fair trade-off between the objectives [Auger et al., 2009]. On the other hand, computing this metric is $\#P$-complete [Bringmann and Friedrich, 2008] and therefore computationally expensive for problems with many objective functions.

Multiobjective Evolutionary Algorithms

Evolutionary algorithms are a population based search technique inspired by Darwinian evolution. The individuals in the population create offspring by mating. Sometimes, mutation occurs. The probability, that an individual will be selected for the mating depends on his *fitness* (i.e. a function, which expresses the quality of the individual).

An evolutionary algorithm usually runs for a prespecified number of generations, however a different stopping criterion can be chosen.

The *mating selection* selects the individuals for crossover and mutation. Most of the evolutionary algorithms nowadays use so called *tournament selection*. It randomly chooses two individuals from the population and the one with better fitness is selected.

The selection and mutation operators are problem specific and also depend on the chosen encoding of the individuals. The *crossover operator* combines two individuals to create a new one (e.g. by swapping some bits between them), the *mutation operator* changes randomly a singe individual (e.g. flips randomly some of its bits). Both of these operators have a probability with which they are used. Usually, the probability of crossover is quite large (0.8), whereas the mutation probability is low (0.01).

The *environmental selection* chooses which individuals should remain in the population to the next generation. The simplest evolutionary algorithms simply use the offspring population. However, more advanced algorithms use an environmental selection which includes some kind of *elitism* (i.e. the best individuals always remain in the population).

The main difference between single-objective (SOGA) and multiobjective (MOGA) evolutionary algorithms is in the fitness assignment technique. In multiobjective evolutionary algorithms, the fitness should drive the algorithm towards the Pareto optimal front, while keeping good spread of solutions.

In the following sections, we discuss the most important multiobjective evolutionary algorithms.

Domination Based Algorithms

This group of algorithms uses the dominance relation in the fitness assignment process, thus following suggestions presented in [Goldberg, 1989]. As the dominance relation itself does not preserve the diversity in the population, another techniques, such as niching, are needed to obtain a good spread of solutions.

These algorithms have been very popular since mid 1990s and some of the state-of-the-art algorithms belong to this group.

In the following parts, the most important algorithms of this group are discussed. Another domination based algorithms were proposed for example by [Fonseca and Fleming, 1993], [Horn et al., 1994] and [Zitzler et al., 2001].

NSGA-II – Nondominated Sorting Genetic Algorithm II

NSGA-II [Deb et al., 2000] is a successor of an older algorithm NSGA [Srinivas and Deb, 1994]. It uses a different niching technique, adds elitism and provides faster nondominated sorting procedure.

Both the algorithms divide the population into nondominated fronts, which are sets of mutually incomparable individuals. These are formally defined using the following two definitions.

Definition 6. For a Pareto set approximation P, $\mathrm{ND}(P)$ is the *set of nondominated vectors in* P.

$$\mathrm{ND}(P) = \{\boldsymbol{x} \in P | \forall \boldsymbol{y} \in P : \boldsymbol{y} \not\preceq \boldsymbol{x}\}$$

Definition 7. The *nondominated fronts of population* P are $F_1, F_2, \ldots$ where

$$F_i = \begin{cases} \mathrm{ND}(P) & \text{if } i = 1 \\ \mathrm{ND}(P \setminus \bigcup_{k=1}^{i-1} F_i) & \text{otherwise} \end{cases}$$

In NSGA-II each individual is assigned a *crowding distance*, which is the sum of distances to the nearest neighbors. Best solutions in each objective function have the crowding distance set to infinity.

The elitism is implemented using the following environmental selection scheme: preceding the selection, the parent and children populations are merged, fitness is calculated for all the individuals in the merged population and the best individuals are selected to the next generation.

The fitness is not computed explicitly, instead, the following selection scheme is used: Each individual is assigned the number of nondominated front it belongs to and the crowding distance. When comparing two individuals i and j the one with lower front number is better. If both i and j belong to the same front the one with larger crowding distance is better.

Although NSGA-II is a relatively old algorithm it is still used today in applications. Its main advantages are its relative effectiveness compared to some newer hypervolume based algorithms. Moreover, it is competitive with the modern algorithms, when only two or three objectives are optimized.

However, [Ishibuchi et al., 2008] have shown that the dominance relation is practically useless, if the number of objective functions increases. They argue that when the number of objectives is about ten most of randomly generated vectors cannot be compared using this relation. In this case, the selection pressure in NSGA-II (and other similar algorithms) is provided only by the niching procedure and does not guide the search towards the Pareto optimal set.

Indicator Based EAs

Indicator based evolutionary algorithms present new approach in evolutionary multiobjective optimization.

These algorithms use indicators (i.e. the hypervolume) instead of (or together with) dominance ranking during fitness assignment, thus avoiding the problem of the missing selection pressure. The value of the indicator is optimized directly and niching is not needed as indicators itself help to provide diversity. Moreover, the indicators can express the decision maker's preferences.

The most commonly used indicators include the $I_{\epsilon+}$ indicator and a hypervolume based indicator I_{HD}, which can be defined as follows. Both of these indicators guide the search towards the Pareto optimal set (see [Zitzler and Künzli, 2004] for a more general condition, which implies this fact).

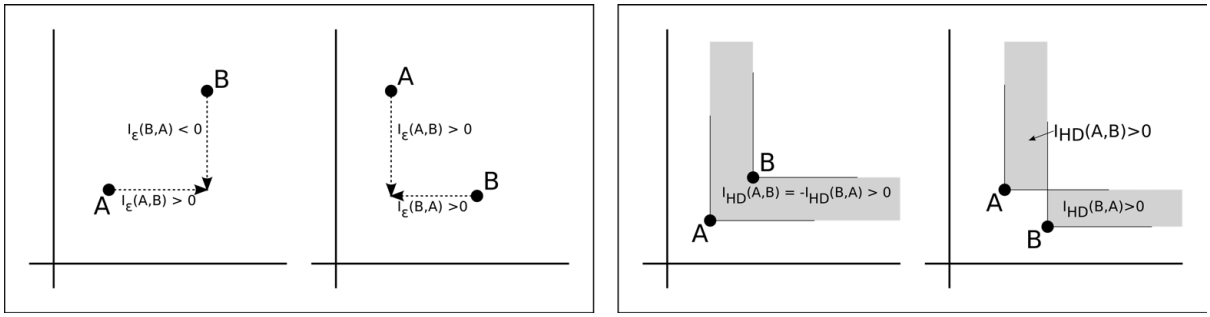

Figure 1. I_ϵ-indicator (left) and I_{HD}-indicator (right)

Definition 8. Let A, B be two Pareto set approximations,

$$I_{\epsilon+}(A,B) = \min_{\epsilon}\{\forall \boldsymbol{x} \in B \exists \boldsymbol{y} \in A \forall i : f_i(\boldsymbol{y}) - \epsilon \leq f_i(\boldsymbol{x})\}$$

This indicator shows, how much a Pareto set approximation has to be moved in the objective space in order to (weakly) dominate another approximation.

Definition 9. For two Pareto set approximations A, B the *hypervolume based indicator*

$$I_{HD}(A,B) = \begin{cases} \mathcal{S}(B) - \mathcal{S}(A) & \text{if } A \preceq B \\ \mathcal{S}(A+B) - \mathcal{S}(A) & \text{otherwise} \end{cases}$$

Here $+$ is the multiset union.

This indicator expresses the difference in the dominated hypervolume of the two approximations. See Figure 1 for a graphical representation of these indicators.

The most successful algorithms in this group use some kind of hypervolume metric. However, computing the value of this indicator is very inefficient and therefore, various ways of speeding up these algorithms are used. For example, [Brockhoff and Zitzler, 2007] proposed a method for reducing the number of objectives and [Bader and Zitzler, 2008] used Monte Carlo method for estimating the hypervolume indicator.

IBEA – Indicator Based EA

[Zitzler and Künzli, 2004] proposed a general indicator based evolutionary algorithm called IBEA.

The fitness in IBEA is assigned as follows: the indicator value of each pair of individuals is computed, and an individual i is assigned fitness

$$F(i) = \sum_{j \in P \setminus \{i\}} -e^{-I(\{j\},\{i\})/\kappa}$$

Here, κ is a scaling factor which has to be set in advance. The purpose of the exponential is to amplify the differences between dominated and nondominated individuals.

In the environmental selection phase, the worst individual is iterativelly removed from the population and the fitness of the other individuals is updated (it is, for individual i

$$F(i) = F(i) + e^{-I(\{i^*\},\{i\})/\kappa}$$

where i^* is the individual which was removed in the iteration).

As the algorithm is sometimes sensitive to the value of κ and to the chosen reference point for the hypervolume based indicator I_{HD}, the authors suggested scaling the objective values to the interval $[0,1]$ and choosing 2 for all the objective functions as the reference point for I_{HD}. They suggest using the values of indicators scaled to $[-1,1]$ and the value of $\kappa = 0.1$.

HypE – Hyper-volume Estimation Based MOGA

In order to overcome the problem with the complexity of computing the hypervolume, [Bader and Zitzler, 2008] proposed Monte Carlo sampling method to estimate it. The hypervolume is computed exactly only for MOPs with up to three objective functions, otherwise it is estimated using Monte Carlo sampling.

In HypE the fitness of each individual is the hypervolume it dominates, but the parts of it which are dominated by more individuals are divided uniformly among them. Thus the sum of fitnesses of all individuals equals the $\mathcal{S}$ metric of the population.

Although this may seem difficult to compute, the authors proposed modified algorithm for computing hypervolume, which also computes this fitness. Also a Monte Carlo version of it exists.

The environmental selection uses the ideas from NSGA-II. The parent and the offspring populations are merged and divided into nondominated fronts and the individuals from the best fronts are added to the new population. From the front, that does not fit completely to the population, the individuals are iteratively removed according to their fitness. After each iteration, the fitness is recalculated.

The Monte Carlo sampling uses a fixed number of samples (authors suggest 10,000). They also tested adaptive Monte Carlo simulation, but the overhead for maintaining the necessary information and the additional computation outweighs the advantages.

Conclusions and Future Work

In the previous sections we described the current state-of-the-art in the field of evolutionary multiobjective optimization. Based on this knowledge, we would like to create more interesting algorithms.

For example, we would like to develop a new crossover operator specifically designed for multiobjective optimization. We believe that such an operator could significantly improve the results of the optimizers. Another field we would like to explore is the parallelization of multiobjective evolutionary algorithms. This should help with the complexity of solving difficult optimization problems with hard-to-compute objective functions or with many objectives. Hybridization with different optimization techniques (like gradient methods for numerical optimization) seems to be one of the most promising ways of speeding up the (parallel) evolutionary algorithms.

References

Auger, A., Bader, J., Brockhoff, D., and Zitzler, E., Theory of the Hypervolume Indicator: Optimal μ-Distributions and the Choice of the Reference Point, in *Foundations of Genetic Algorithms (FOGA 2009)*, workshop version, 2009.

Bader, J. and Zitzler, E., HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization, TIK Report 286, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, 2008.

Bringmann, K. and Friedrich, T., Approximating the volume of unions and intersections of high-dimensional geometric objects, *CoRR*, *abs/0809.0835*, 2008.

Brockhoff, D. and Zitzler, E., Improving Hypervolume-based Multiobjective Evolutionary Algorithms by Using Objective Reduction Methods, in *Congress on Evolutionary Computation (CEC 2007)*, pp. 2086–2093, IEEE Press, 2007.

Das, I. and Dennis, J. E., Normal-boundary intersection: a new method for generating Pareto optimal points in multicriteria optimization problems, *SIAM Journal on Optimization*, *8(3)*, 631–657, URL `citeseer.ist.psu.edu/das96normalboundary.html`, 1998.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T., A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGA-II, in *PPSN*, edited by M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. M. Guervós, and H.-P. Schwefel, vol. 1917 of *Lecture Notes in Computer Science*, pp. 849–858, Springer, 2000.

Dorigo, M., *Optimization, Learning, Natural Algorithms*, Ph.D. thesis, Politecnico di Milano, Italy, 1992.

Fonseca, C. M. and Fleming, P. J., Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, in *Genetic Algorithms: Proceedings of the Fifth International Conference*, pp. 416–423, Morgan Kaufmann, URL `http://citeseer.ist.psu.edu/175195.html`, 1993.

Goldberg, D. E., *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley, 1989.

Horn, J., Nafpliotis, N., and Goldberg, D. E., A niched pareto genetic algorithm for multiobjective optimization, in *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, pp. 82–87, 1994.

Ishibuchi, H., Tsukamoto, N., and Nojima, Y., Evolutionary many-objective optimization: A short review, in *Proceedings of 2008 IEEE Congress on Evolutionary Computation*, pp. 2424–2431, 2008.

Kennedy, J. and Eberhart, R., Particle swarm optimization, in *IEEE Int'l. Conf. on Neural Networks, IV*, pp. 1942–1948, 1995.

Srinivas, N. and Deb, K., Multiobjective optimization using nondominated sorting in genetic algorithms, *Evolutionary Computation*, *2*, 221–248, 1994.

Zitzler, E. and Künzli, S., Indicator-Based Selection in Multiobjective Search, in *Conference on Parallel Problem Solving from Nature (PPSN VIII)*, edited by X. Yao et al., vol. 3242 of *LNCS*, pp. 832–842, Springer, 2004.

Zitzler, E., Laumanns, M., and Thiele, L., SPEA2: Improving the strength pareto evolutionary algorithm, TIK Report 103, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Zurich, Switzerland, 2001.