Evolutionary Multiobjective Optimization

Ajith Abraham[1] and Lakhmi Jain[2]

[1] Department of Computer Science, Oklahoma State University, USA
ajith.abraham@ieee.org
[2] University of South Australia, Adelaide, Australia
lakhmi.jain@unisa.edu.au

Summary. Very often real world applications have several multiple conflicting objectives. Recently there has been a growing interest in evolutionary multiobjective optimization algorithms which combines two major disciplines: evolutionary computation and the theoretical frameworks of multicriteria decision making. In this introductory chapter, we define some fundemental concepts of multiobjective optimization emphasizing the motivation and advantages of using evolutionary algorithms. We then layout the important contributions of the remaining chapters of this volume.

1 What is Multiobjective Optimization?

Even though some real world problems can be reduced to a matter of single objective very often it is hard to define all the aspects in terms of a single objective. Defining multiple objectives often gives a better idea of the task. Multiobjective optimization has been available for about two decades, and recently its application in real world problems is continously increasing. In contrast to the plethora of techniques available for single-objective optimization, relatively few techniques have been developed for multiobjective optimization.

In single objective optimization, the search space is often well defined. As soon as there are several possibly contradicting objectives to be optimized simultaneously, there is no longer a single optimal solution but rather a whole set of possible solutions of equivalent quality. When we try to optimize several objectives at the same time the search space also becomes partially ordered. To obtain the optimal solution, there will be a set of optimal trade-offs between the conflicting objectives. A multiobjective optimization problem is defined by a function f which maps a set of constraint variables to a set of objective values.

As shown in Figure 1, a solution could be best, worst and also indifferent to other solutions (neither dominating or dominated) with respect to the objective values. Best solution means a solution not worst in any of the objectives and at least better in one objective than the other. An optimal solution is the solution that is not dominated by any other solution in the search space. Such an optimal solution is called Pareto optimal and the entire set of such optimal trade-offs solutions is called Pareto optimal set. As evident, in a real world situation a decision making (trade-off) process is required to obtain the optimal solution. Even though there are several ways to approach a multiobjective optimization problem, most work is concentrated on the approximation of the Pareto set.

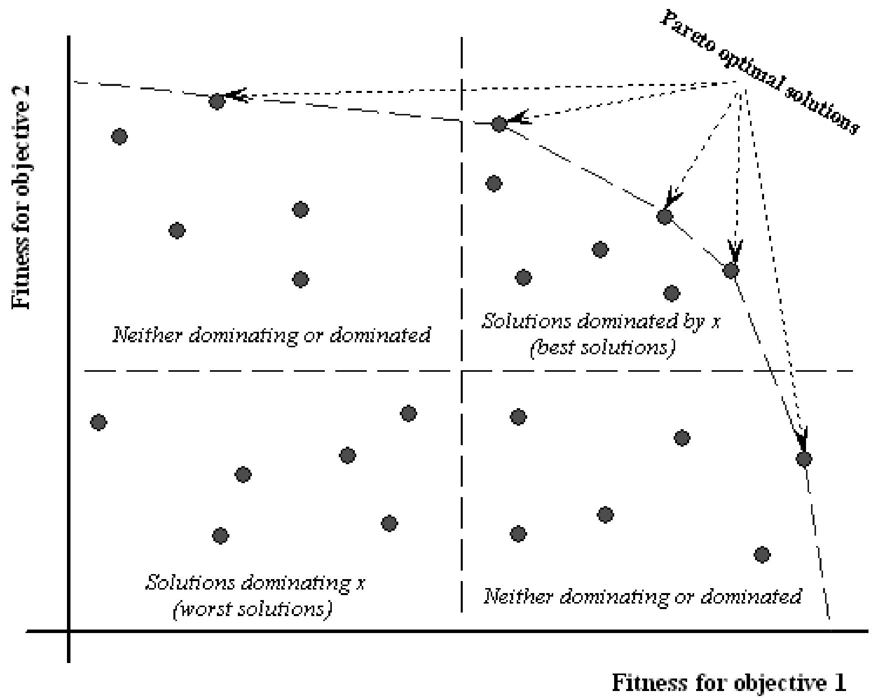

Figure 1. Concept of Pareto optimality

2 Why Use Evolutionary Algorithms for Multiobjective Optimization?

A number of stochastic optimization techniques like simulated annealing; tabu search, ant colony optimization etc. could be used to generate the Pareto set. Just because of the working procedure of these algorithms, the solutions obtained very often tend to be stuck at a good approximation and they do not guarantee to identify optimal trade-offs. Evolutionary algorithm is characterized by a population of solution candidates and the reproduction process enables to combine existing solutions to generate new solutions. Finally, natural selection determines which individuals of the current population participate in the new population.

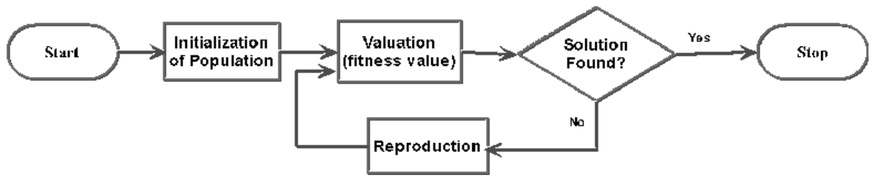

Figure 2. Flowchart of evolutionary algorithm iteration

The iterative computation process is illustrated in Figure 2. Multiobjective evolutionary algorithms can yield a whole set of potential solutions, which are all optimal in some sense. After the first pioneering work on multiobjective evolutionary optimization in the eighties [1], several different algorithms have been proposed and successfully applied to various problems. For comprehensive overviews and discussions, the reader is referred to [2][3]. Some of the other advantages of having evolutionary algorithms is that they require very little knowledge about the problem being solved, easy to implement, robust and could be implemented in a parallel environment.

3 Evolutionary Multiobjective Optimization: Challenges, Advances and Applications

The main challenge in a multiobjective optimization environment is to minimize the distance of the generated solutions to the Pareto set and to maximize the diversity of the developed Pareto set. A good Pareto set may be obtained by appropriate guiding of the search process through careful design of reproduction operators and fitness assignment strategies. To obtain diversification special care has to be taken in the selection process. Special care is also to be taken care to prevent non-dominated solutions from being lost [4][5]. Addressing the evolutionary multiobjective optimization problem and the various design challenges using different intelligent approaches is the novelty of this edited volume. This volume comprises of 12 chapters and each chapter is complete by itself. The rest of the volume is organized as follows.

In the following Chapter Coello presents the basic concepts of multiobjective evolutionary algorithms, its potential appications, metrics, test functions and concludes with some of the most promising future research directions.

Laumanns begins Chapter 3 by presenting the convergence behavior of simple evolutionary algorithms with different selection strategies on a continuous multiobjective optimization problem. Special focus is given to the problem of controlling the mutation strength, since an adaptation of the mutation strength is necessary to converge to the optimum with arbitrary precision, and to achieve linear convergence order. Experiment results reveal that the convergence properties achieved by a self-adaptation of the mutation strength on single-objective problems do not carry over to the multiobjective case, if a simple dominance-based selection scheme is used. As a solution, a combined strategy is proposed using dominance-based selection in the archive and scalarizing functions in the working population.

In Chapter 4, Mumford explains a Pareto-based approach to evolutionary multi-objective optimization, that avoids most of the time consuming global calculations typical of other multi-objective evolutionary techniques. The new approach uses a simple uniform selection strategy within a steady-state evolutionary algorithm

and employs a straightforward elitist mechanism for replacing population members with their offspring. An important advantage of the proposed method is that global calculations for fitness and Pareto dominance are not needed. The perormance of the algorithm is demonstrated using some benchmark combinatorial problems and continuous functions.

Mostaghim and Teich in Chapter 5 shows the importance of special data structures for storing and updating archives which would have a great impact on the required computational time, especially when optimizing higher-dimensional problems with large Pareto-sets. Authors introduce Quad-trees as an alternative data structure to linear lists for storing Pareto-sets. Performance of the quad-trees data structures are evaluated and compared using several multi-objective example problems. The results presented show that typically, linear lists perform better for small population sizes and higher-dimensional Pareto-fronts (large archives) whereas Quad-trees perform better for larger population sizes and Pareto-sets of small cardinality.

In Chapter 6, Deb et al. suggests three different approaches for systematically designing test problems for evaluating multiobjective evolutionary algorithms. The simplicity of construction, scalability to any number of decision variables and objectives, knowledge of the shape and the location of the resulting Pareto-optimal front, and introduction of controlled difficulties in both converging to the true Pareto-optimal front and maintaining a widely distributed set of solutions are the main features of the suggested test problems. These test problems should be found useful in various research activities on new multiobjective evolutionary algorithms and to enhance the understanding of the working principles of multiobjective evolutionary algorithms.

Srinivasan and Seow in Chapter 7 presents an hybrid combination of particle swarm optimization and evolutionary algorithm for multiobjective optimization problems. The main algorithm for swarm intelligence is Particle Swarm Optimization, which is inspired by the paradigm of birds flocking.

The core updating mechanism of the particle swarm optimization algorithm relies only on two simple self-updating equations and the process of updating the individuals per iteration is fast as compared to the computationally expensive reproduction mechanism using mutation or crossover operations in typical evolutionary algorithm. While additional domain-specific heuristics related to the real-world problems cannot be easily incorporated in the particle swarm optimization algorithm; in an evolutionary algorithm, heuristics can be easily incorporated in the population generator and mutation operator to prevent leading the individuals to infeasible solutions. Therefore, a direct particle swarm optimization does not perform well in its search in complex multi-constrained solution spaces, which are the case for many complex real world problems. To overcome the limitations of particle swarm optimization and evolutionary algorithms, a hybridized algorithm is proposed to use a synergistic combination of particle swarm optimization and evolutionary algorithm. Experiment results using some test functions illustrates the feasibility of the hybrid approach as a multiobjective search algorithm.

In Chapter 8, Dumitrescu et al. propose a new evolutionary elitist approach combining a non-standard solution representation and an evolutionary optimization technique which permits the detection of continuous decision regions. Each solution in the final population corresponds to a decision region of Pareto optimal set. The proposed method is evaluated using some test functions.

Hiroyasu et al. in Chapter 9 addresses the parallel implementation of multiobjective evolutionary algorithms to manage the computational costs especially for higher-dimensional problems with large Pareto-sets . They propose a parallel genetic algorithm for multi objective optimization problems called Multiobjective Genetic Algorithm with Distributed Environment Scheme (MOGADES). Further a new mechanism is added to multiobjective genetic algorithms called Distributed Cooperation model of Multi-Objective Genetic Algorithm (DCMOGA). In DCMOGA, there are not only individuals for searching Pareto optimum solutions but also individuals for searching the solution of one object. After illustrating MOGADES and DCMOGA, these two algorithms were combined. This hybrid algorithm is called "Distributed Cooperation model of Multi-Objective Genetic Algorithm with Environmental Scheme (DCMOGADES). The performance of DCMOGADES is illustrated using some test finctions.

In Chapter 10, Montes and Coello describe the general multiobjective optimization concepts that can be used to incorporate constraints of any type (linear, nonlinear, equality and inequality) into the fitness function of a genetic algorithm used for global optimization. Several approaches reported in the literature are also described and four of them are compared using several test functions.

Goldberg and Hammerman in Chapter 11 present a new operator which, when added to a genetic algorithm (GA), improved the performance of the GA for locating optimal finite state automata. The new operator (termed MTF) reorganizes a finite state automaton (FSA) genome during the execution of the genetic algorithm. MTF systematically renames the states and moves them to the front of the genome. The operator was tested on the ant trail problem. Across different criteria (failure rate, processing time to locate a solution, number of generations needed to locate a solution), the MTF-enhanced GA realized speedups between 110% and 579% over the non-enhanced version. In addition, the successful FSAs found by the genetic algorithm augmented with MTF were 25%-46% smaller in size than those found by the original GA.

In the last chapter, Lagaros et al. deals with a practical problem of structural sizing. The aim is to minimize the weight of the structure under certain restrictions imposed by design codes. Authors present two approaches (rigorous and simplified) with respect to the loading condition and their efficiency is compared to find the optimum design of a structure under multiple objectives. In the context of the rigorous approach a number of artificial accelerograms are produced from the design response spectrum of the region for elastic structural response, which constitutes the multiple loading conditions under which the structures are optimally designed. This approach is compared with the approximate one based on simplifications adopted by the seismic codes. Experiment results reveal that the Pareto sets obtained by the rigorous approach and the simplified one were different.

4 References

1. J.D. Schaffer, Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, Ph.D. Thesis, Vanderbilt University, Nashville, TN, 1984.
2. C. A. Coello Coello, Comprehensive Survey of Evolutionary-Based Multi-objective Optimization Techniques, Knowledge and Information Systems. An International Journal, 1(3):269–308, 1999.
3. K. Deb, Multiobjective Optimization using Evolutionary Algorithms, John Wiley & Sons, UK, 2001.
4. M. Laumanns, L. Thiele, K. Deb, E. Zitzler, Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. Evolutionary Computation, MIT Press, Vol. 10, No. 3, pages 263-282, 2002.
5. E. Zitzler, M. Laumanns, S. Bleuler, A Tutorial on Evolutionary Multi-objective Optimization, Workshop on Multiple Objective Metaheuristics (MOMH 2002), Springer-Verlag, Berlin, Germany, 2004.