Evolutionary Multiobjective Optimization: Current and Future Challenges

Carlos A. Coello Coello

CINVESTAV-IPN
Evolutionary Computation Group
Departamento de Ingeniería Eléctrica
Sección de Computación
Av. Instituto Politécnico Nacional No. 2508
Col. San Pedro Zacatenco
México, D.F. 07300, MEXICO
ccoello@cs.cinvestav.mx

Summary. In this paper, we will briefly discuss the current state of the research on evolutionary multiobjective optimization, emphasizing the main achievements obtained to date. Achievements in algorithmic design are discussed from its early origins until the current approaches which are considered as the "second generation" in evolutionary multiobjective optimization. Some relevant applications are discussed as well, and we conclude with a list of future challenges for researchers working (or planning to work) in this area in the next few years.

1 Introduction

Several years ago, researchers realized that the principle of "survival of the fittest" used by nature could be simulated to solve problems [11]. This gave rise to a type of heuristics known as Evolutionary Algorithms (EAs). EAs have been very popular in search and optimization tasks in the last few years with a constant development of new algorithms, theoretical achivements and novel applications [15, 1].

One of the emergent research areas in which EAs have become increasingly popular is multiobjective optimization. In multiobjective optimization problems, we have two or more objective functions to be optimized at the same time, instead of having only one. As a consequence, there is no unique solution to multiobjective optimization problems, but instead, we aim to find all of the good trade-off solutions available (the so-called Pareto optimal set).

The first implementation of a multi-objective evolutionary algorithm dates back to the mid-1980s [36]. Since then, a considerable amount of research has

been done in this area, now known as evolutionary multi-objective optimization (EMO for short). The growing importance of this field is reflected by a significant increment (mainly during the last eight years) of technical papers in international conferences and peer-reviewed journals, books, special sessions in international conferences and interest groups on the Internet [6].[1]

Evolutionary algorithms seem also particularly desirable for solving multiobjective optimization problems because they deal simultaneously with a set of possible solutions (the so-called population) which allows us to find several members of the Pareto optimal set in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of the traditional mathematical programming techniques. Additionally, evolutionary algorithms are less susceptible to the shape or continuity of the Pareto front (e.g., they can easily deal with discontinuous and concave Pareto fronts), whereas these two issues are a real concern for mathematical programming techniques [6].

This paper deals with some of the current and future research trends in evolutionary multiobjective optimization. The paper is organized as follows. Section 2 presents some basic concepts used in multiobjective optimization. Section 3 briefly describes the origins of evolutionary multiobjective optimization. Section 4 introduces the so-called first generation multiobjective evolutionary algorithms. Second generation multiobjective evolutionary algorithms are discussed in Section 5, emphasizing the role of elitism in evolutionary multiobjective optimization. Finally, Section 7 discusses some of the research trends that are likely to be predominant in the next few years.

2 Basic Concepts

The emphasis of this paper is the solution of multiobjective optimization problems (MOPs) of the form:

$$\text{minimize } [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})] \tag{1}$$

subject to the m inequality constraints:

$$g_i(\mathbf{x}) \geq 0 \quad i = 1, 2, \dots, m \tag{2}$$

and the p equality constraints:

$$h_i(\mathbf{x}) = 0 \quad i = 1, 2, \dots, p \tag{3}$$

where k is the number of objective functions $f_i : \mathbb{R}^n \to \mathbb{R}$. We call $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ the vector of decision variables. We wish to determine from

---

[1] The author maintains an EMO repository with over 1000 bibliographical entries at: `http://delta.cs.cinvestav.mx/~ccoello/EMOO`, with mirrors at `http://www.lania.mx/~ccoello/EMOO/` and `http://www.jeo.org/emo/`

among the set $\mathcal{F}$ of all vectors which satisfy (2) and (3) the particular set of values $x_1^*, x_2^*, \ldots, x_n^*$ which yield the optimum values of all the objective functions.

2.1 Pareto optimality

It is rarely the case that there is a single point that simultaneously optimizes all the objective functions. Therefore, we normally look for "trade-offs", rather than single solutions when dealing with multiobjective optimization problems. The notion of "optimality" is therefore, different in this case. The most commonly adopted notion of optimality is the following:

We say that a vector of decision variables $\mathbf{x}^* \in \mathcal{F}$ is *Pareto optimal* if there does not exist another $\mathbf{x} \in \mathcal{F}$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ for all $i = 1, \ldots, k$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one j.

In words, this definition says that $\mathbf{x}^*$ is Pareto optimal if there exists no feasible vector of decision variables $\mathbf{x} \in \mathcal{F}$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion. Unfortunately, this concept almost always gives not a single solution, but rather a set of solutions called the *Pareto optimal set*. The vectors $\mathbf{x}^*$ correspoding to the solutions included in the Pareto optimal set are called *nondominated*. The image of the Pareto optimal set under the objective functions is called *Pareto front*.

3 On the origins of evolutionary multiobjective optimization

The first actual implementation of what it is now called a multi-objective evolutionary algorithm (or MOEA, for short) was Schaffer's *Vector Evaluated Genetic Algorithm* (VEGA), which was introduced in the mid-1980s, mainly aimed for solving problems in machine learning [36]. VEGA basically consisted of a simple genetic algorithm (GA) with a modified selection mechanism. At each generation, a number of sub-populations were generated by performing proportional selection according to each objective function in turn. Thus, for a problem with k objectives, k sub-populations of size M/k each would be generated (assuming a total population size of M). These sub-populations would then be shuffled together to obtain a new population of size M, on which the GA would apply the crossover and mutation operators in the usual way. Schaffer realized that the solutions generated by his system were nondominated in a local sense, because their nondominance was limited to the current population, which was obviously not appropriate. Also, he noted a problem that in genetics is known as "speciation" (i.e., we could have the evolution of "species" within the population which excel on different aspects of performance). This problem arises because this technique selects individuals who excel in one dimension of performance, without looking at the other

dimensions. The potential danger doing that is that we could have individuals with what Schaffer called "middling" performance[2] in all dimensions, which could be very useful for compromise solutions, but which will not survive under this selection scheme, since they are not in the extreme for any dimension of performance (i.e., they do not produce the best value for any objective function, but only moderately good values for all of them). Speciation is undesirable because it is opposed to our goal of finding Pareto optimal solutions. Although VEGA's speciation can be dealt with using heuristics or other additional mechanisms, it remains as the main drawback of VEGA.

From the second half of the 1980s up to the first half of the 1990s, few other researchers developed MOEAs. Most of the work reported back then involves rather simple evolutionary algorithms that use an aggregating function (linear in most cases) [23], lexicographic ordering [14], and target-vector approaches (i.e., nonlinear aggregating functions) [19]. All of these approaches were strongly influenced by the work done in the operations research community and in most cases did not require any major modifications to the evolutionary algorithm adopted.

The algorithms proposed in this initial period are rarely referenced in the current literature except for VEGA (which is still used by some researchers). However, the period is of great importance because it provided the first insights into the possibility of using evolutionary algorithms for multiobjective optimization. The fact that only relatively naive approaches were developed during this stage is natural considering that these were the initial attempts to develop multiobjective extensions of an evolutionary algorithm. Such approaches kept most of the original evolutionary algorithm structure intact (only the fitness function was modified in most cases) to avoid any complex additional coding. The emphasis in incorporating the concept of Pareto dominance into the search mechanism of an evolutionary algorithm would come later.

4 MOEAs: First Generation

The major step towards the first generation of MOEAs was given by David E. Goldberg on pages 199 to 201 of his famous book on genetic algorithms published in 1989 [15]. In his book, Goldberg analyzes VEGA and proposes a selection scheme based on the concept of Pareto optimality. Goldberg not only suggested what would become the standard first generation MOEA, but also indicated that stochastic noise would make such algorithm useless unless some special mechanism was adopted to block convergence. First generation MOEAs typically adopted niching or fitness sharing for that sake. The most representative algorithms from the first generation are the following:

---

[2] By "middling", Schaffer meant an individual with acceptable performance, perhaps above average, but not outstanding for any of the objective functions.

1. **Nondominated Sorting Genetic Algorithm** (NSGA): This algorithm was proposed by Srinivas and Deb [37]. The approach is based on several layers of classifications of the individuals as suggested by Goldberg [15]. Before selection is performed, the population is ranked on the basis of non-domination: all nondominated 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 nondominated 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. This allows to search for nondominated regions, and results in convergence of the population toward such regions. Sharing, by its part, helps to distribute the population over this region (i.e., the Pareto front of the problem).

2. **Niched-Pareto Genetic Algorithm** (NPGA): Proposed by Horn et al. [22]. The NPGA uses a tournament selection scheme based on Pareto dominance. The basic idea of the algorithm is the following: 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 nondominated individual wins. When both competitors are either dominated or nondominated (i.e., there is a tie), the result of the tournament is decided through fitness sharing [16].

3. **Multi-Objective Genetic Algorithm** (MOGA): Proposed by Fonseca and Fleming [12]. In MOGA, the rank of a certain individual corresponds to the number of chromosomes in the current population by which it is dominated. Consider, for example, an individual x_i at generation t, which is dominated by $p_i^{(t)}$ individuals in the current generation.
The rank of an individual is given by [12]:

$$\text{rank}(x_i, t) = 1 + p_i^{(t)} \tag{4}$$

All nondominated individuals are assigned rank 1, while dominated ones are penalized according to the population density of the corresponding region of the trade-off surface. Fitness assignment is performed in the following way [12]:

a) Sort population according to rank.
b) Assign fitness to individuals by interpolating from the best (rank 1) to the worst (rank $n \leq M$, where M is the total population size) in the way proposed by Goldberg (1989), according to some function, usually linear, but not necessarily.

c) Average the fitnesses of individuals with the same rank, so that all of them are sampled at the same rate. This procedure keeps the global population fitness constant while maintaining appropriate selective pressure, as defined by the function used.

The main questions raised during the first generation were:

- Are aggregating functions (so common before and even during the golden years of Pareto ranking) really doomed to fail when the Pareto front is non-convex [7]? Are there ways to deal with this problem? Is it worth trying? Some recent work seems to indicate that even linear aggregating functions are not death yet [25].
- Can we find ways to maintain diversity in the population without using niches (or fitness sharing), which requires a process $O(M^2)$ where M refers to the population size?
- If assume that there is no way of reducing the $O(kM^2)$ process required to perform Pareto ranking (k is the number of objectives and M is the population size), how can we design a more efficient MOEA?
- Do we have appropriate test functions and metrics to evaluate quantitatively an MOEA? Not many people worried about this issue until near the end of the first generation. During this first generation, practically all comparisons were done visually (plotting the Pareto fronts produced by different algorithms) or were not provided at all (only the results of the proposed method were reported).
- When will somebody develop theoretical foundations for MOEAs?

Summarizing, the first generation was characterized by the use of selection mechanisms based on Pareto ranking and fitness sharing was the most common approach adopted to maintain diversity. Much work remained to be done, but the first important steps towards a solid research area had been already taken.

5 MOEAs: Second Generation

The second generation of MOEAs was born with the introduction of the notion of elitism. 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 nondominated individuals. The use of this external population (or file) raises several questions:

- How does the external file interact with the main population?
- What do we do when the external file is full?
- Do we impose additional criteria to enter the file instead of just using Pareto dominance?

Elitism can also be introduced through the use of a $(\mu + \lambda)$-selection in which parents compete with their children and those which are nondominated (and possibly comply with some additional criterion such as providing a better distribution of solutions) are selected for the following generation.

The previous points bring us to analyze in more detail the true role of elitism in evolutionary multiobjective optimization. For that sake, we will review next the way in which some of the second-generation MOEAs implement elitism:

1. **Strength Pareto Evolutionary Algorithm** (SPEA): This algorithm was introduced by Zitzler and Thiele [43]. This approach was conceived as a way of integrating different MOEAs. SPEA uses an archive containing nondominated solutions previously found (the so-called external nondominated set). At each generation, nondominated individuals are copied to the external nondominated set. For each individual in this external set, a *strength* value is computed. This strength is similar to the ranking value of MOGA, since it is proportional to the number of solutions to which a certain individual dominates. It should be obvious that the external nondominated set is in this case the elitist mechanism adopted. In SPEA, the fitness of each member of the current population is computed according to the strengths of all external nondominated solutions that dominate it. Additionally, a clustering technique called "average linkage method" [28] is used to keep diversity.

2. **Strength Pareto Evolutionary Algorithm 2** (SPEA2): SPEA2 has three main differences with respect to its predecessor [42]: (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, and (3) it has an enhanced archive truncation method that guarantees the preservation of boundary solutions. Thefore, in this case the elitist mechanism is just an improved version of the previous.

3. **Pareto Archived Evolution Strategy** (PAES): This algorithm was introduced by Knowles and Corne [26]. 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 some of the nondominated 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. However, an interesting aspect of this algorithm is the mechanism 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 "geograph-

ical 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).

4. **Nondominated Sorting Genetic Algorithm II** (NSGA-II): Deb et al. [8] proposed a revised version of the NSGA [37], called NSGA-II, which is more efficient (computationally speaking), uses elitism and a crowded comparison operator that keeps diversity without specifying any additional parameters. The NSGA-II does not use an external memory as the previous algorithms. Instead, the elitist mechanism consists of combining the best parents with the best offspring obtained (i.e., a $(\mu + \lambda)$-selection).

5. **Niched Pareto Genetic Algorithm 2** (NPGA 2): Erickson et al. [9] proposed a revised version of the NPGA [22] called the NPGA 2. This algorithm uses Pareto ranking but keeps tournament selection (solving ties through fitness sharing as in the original NPGA). In this case, no external memory is used and the elitist mechanism is similar to the one adopted by the NSGA-II. Niche counts in the NPGA 2 are calculated using individuals in the partially filled next generation, rather than using the current generation.

6. **Micro Genetic Algorithm**: This approach was introduced by Coello Coello & Toscano Pulido [5]. A micro-genetic algorithm is a GA with a small population and a reinitialization process. The micro-GA starts with a random population that feeds the population memory, which is divided in two parts: a replaceable and a non-replaceable portion. The non-replaceable portion of the population memory never changes during the entire run and is meant to provide the required diversity for the algorithm. In contrast, the replaceable portion experiences changes after each cycle of the micro-GA. The population of the micro-GA at the beginning of each of its cycles is taken (with a certain probability) from both portions of the population memory so that there is a mixture of randomly generated individuals (non-replaceable portion) and evolved individuals (replaceable portion). During each cycle, the micro-GA undergoes conventional genetic operators. After the micro-GA finishes one cycle, two nondominated vectors are chosen[3] from the final population and they are compared with the contents of the external memory (this memory is initially empty). If either of them (or both) remains as nondominated after comparing it against the vectors in this external memory, then they are included there (i.e., in the external memory). This is the historical archive of nondominated vectors. All dominated vectors contained in the external

---

[3] This is assuming that there are two or more nondominated vectors. If there is only one, then this vector is the only one selected.

memory are eliminated. The micro-GA uses then three forms of elitism: (1) it retains nondominated solutions found within the internal cycle of the micro-GA, (2) it uses a replaceable memory whose contents is partially "refreshed" at certain intervals, and (3) it replaces the population of the micro-GA by the nominal solutions produced (i.e., the best solutions found after a full internal cycle of the micro-GA).

Second generation MOEAs can be characterized by an emphasis on efficiency and by the use of elitism (in the two main forms previously described). During the second generation, some important theoretical work also took place, mainly related to convergence [34, 20]. Also, metrics and standard test functions were developed to validate new MOEAs [41].

The main concerns during the second generation (which we are still living nowadays) are the following:

- Are our metrics reliable? What about our test functions? We have found out that developing good metrics is in itself a multiobjective optimization problem, too. In fact, it is ironic that nowadays we are going back to trusting more visual comparisons than metrics as during the first generation.
- Are we ready to tackle problems with more than two objective functions efficiently? Is Pareto ranking doomed to fail when dealing with too many objectives? If so, then what is the limit up to which Pareto ranking can be used to select individuals reliably?
- What are the most relevant theoretical aspects of evolutionary multiobjective optimization that are worth exploring in the short-term?

6 Applications

An analysis of the evolution of the EMO literature reveals some interesting facts (see [6] for details). From the first MOEA published in 1985 [36] up to the first survey of the area published in 1995 [13], the number of published papers related to EMO is relatively low. However, from 1995 to our days, the increase of EMO-related papers is exponential. Today, the EMO repository registers over 1000 papers, from which a vast majority are applications. The vast number of EMO papers currently available makes it impossible to attempt to produce a detailed review of them in this section. Instead, we will discuss the most popular application fields, indicating some of the specific areas within them in which researchers have focused their main efforts.

Current EMO applications can be roughly classified in three large groups: engineering, industrial and scientific. Some specific areas within each of these groups are indicated next. We will start with the engineering applications, which are, by far, the most popular in the literature. This should not be too surprising, since engineering disciplines normally have problems with better understood mathematical models which facilitates the use of evolutionary algorithms. A representative sample of engineering applications is the following

(aeronautical engineering seems to be the most popular subdiscipline within this group):

- Electrical engineering [40]
- Hydraulic engineering [33]
- Structural engineering [27]
- Aeronautical engineering [29]
- Robotics [30]
- Control [39]
- Telecommunications [32]
- Civil engineering [2]

Industrial applications occupy the second place in popularity in the EMO literature. Within this group, scheduling is the most popular subdiscipline. A representative sample of industrial applications is the following:

- Design and manufacture [35]
- Scheduling [38]
- Management [24]

Finally, we have a variety of scientific applications, from which the most popular are (for obvious reasons) those related to computer science:

- Chemistry [21]
- Physics [17]
- Medicine [31]
- Computer science [3]

The above distribution of applications indicates a strong interest for developing real-world applications of EMO algorithms (something not surprising considering that most real-world problems are of a multiobjective nature). Furthermore, the previous sample of EMO applications should give a general idea of the application areas that have not been explored in enough depth yet (e.g., computer vision, coordination of agents, pattern recognition, etc. [6]).

7 Future Challenges

Once we have been able to distinguish between the first and second generations in evolutionary multiobjective optimization, a reasonable question is: where are we heading now? In the last few years, there has been a considerable growth in the number of publications related to evolutionary multiobjective optimization. However, the variety of topics covered is not as rich as the number of publications released each year. The current trend is to either develop new algorithms (validating them with some of the metrics and test functions available) or to develop interesting applications of existing algorithms. We will finish this section with a list of some of the research topics that we believe that will keep researchers busy during the next few years:

- **Incorporation of preferences in MOEAs**: Despite the efforts of some researchers to incorporate user's preferences into MOEAs as to narrow the search, most of the multicriteria decision making techniques developed in Operations Research have not been applied in evolutionary multiobjective optimization [4]. Such incorporation of preferences is very important in real-world applications since the user only needs one Pareto optimal solution and not the whole set as normally assumed by EMO researchers.

- **Highly-Constrained Search Spaces**: There is little work in the current literature regarding the solution of multiobjective problems with highly-constrained search spaces. However, it is rather common to have such problems in real-world applications and it is then necessary to develop novel constraint-handling techniques that can deal with highly-constrained search spaces efficiently.

- **Parallelism**: We should expect more work on parallel MOEAs in the next few years. Currently, there is a noticeable lack of research in this area [6] and it is therefore open to new ideas. It is necessary to have more algorithms, formal models to prove convergence, and more real-world applications that use parallelism.

- **Theoretical Foundations**: It is quite important to develop the theoretical foundations of MOEAs. Although a few steps have been taken regarding proving convergence using Markov Chains (e.g., [34]), and analyzing metrics [41], much more work remains to be done (see [6]).

- **Use of More Efficient Data Structures**: The usage of more efficient data structures to store nondominated vectors is just beginning to be analyzed in evolutionary multiobjective optimization (see for example [10]). Note however, that such data structures have been in use for a relatively long time in Operations Research [18].

8 Conclusions

This paper has provided a general view of the field known as evolutionary multiobjective optimization. We have provided a historical analysis of the development of the area, emphasizing the algorithmic differences between the two main stages that we have undergone so far (the two so-called "generations"). The notion of elitism has been identified as the main responsible of the current generation of algorithms used in this area. Also, some of the most important issues (stated in the form of questions) raised during each of these two generations were briefly indicated.

In the final part of the paper, we have discussed some of the most relevant applications developed in the literature and we identified certain research

trends. We have finished this paper with some promising areas of future research in evolutionary multiobjective optimization, hoping that this information may be useful to newcomers who whish to contribute to this emerging research field.

9 Acknowledgments

References

1. Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors. *Handbook of Evolutionary Computation*. Institute of Physics Publishing and Oxford University Press, 1997.
2. Richard Balling and Scott Wilson. The maximim fitness function for multi-objective evolutionary computation: Application to city planning. In Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 1079–1084, San Francisco, CA, 2001. Morgan Kaufmann Publishers.
3. Stefan Bleuler, Martin Brack, Lothar Thiele, and Eckart Zitzler. Multiobjective Genetic Programming: Reducing Bloat Using SPEA2. In *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, volume 1, pages 536–543, Piscataway, New Jersey, May 2001. IEEE Service Center.
4. Carlos A. Coello Coello. Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. In *2000 Congress on Evolutionary Computation*, volume 1, pages 30–37, Piscataway, New Jersey, July 2000. IEEE Service Center.
5. Carlos A. Coello Coello and Gregorio Toscano Pulido. Multiobjective Optimization using a Micro-Genetic Algorithm. In Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 274–282, San Francisco, California, 2001. Morgan Kaufmann Publishers.
6. Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, May 2002. ISBN 0-3064-6762-3.
7. Indraneel Das and John Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14(1):63–69, 1997.
8. Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

9. Mark Erickson, Alex Mayer, and Jeffrey Horn. The Niched Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 681–695. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

10. Richard M. Everson, Jonathan E. Fieldsend, and Sameer Singh. Full Elite Sets for Multi-Objective Optimisation. In I.C. Parmee, editor, *Proceedings of the Fifth International Conference on Adaptive Computing Design and Manufacture (ACDM 2002)*, volume 5, pages 343–354, University of Exeter, Devon, UK, April 2002. Springer-Verlag.

11. David B. Fogel, editor. *Evolutionary Computation. The Fossil Record. Selected Readings on the History of Evolutionary Algorithms*. The Institute of Electrical and Electronic Engineers, New York, 1998.

12. 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, CA, 1993. Morgan Kaufmann Publishers.

13. Carlos M. Fonseca and Peter J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, Spring 1995.

14. Michael P. Fourman. Compaction of symbolic layout using genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 141–153, Hillsdale, NJ, 1985. Lawrence Erlbaum.

15. David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, MA, 1989.

16. David E. Goldberg and Jon Richardson. Genetic algorithm with sharing for multimodal function optimization. In John J. Grefenstette, editor, *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49, Hillsdale, NJ, 1987. Lawrence Erlbaum.

17. I. Golovkin, R. Mancini, S. Louis, Y. Ochi, K. Fujita, H. Nishimura, H. Shirga, N. Miyanaga, H. Azechi, R. Butzbach, I. Uschmann, E. Förster, J. Delettrez, J. Koch, R.W. Lee, and L. Klein. Spectroscopic Determination of Dynamic Plasma Gradients in Implosion Cores. *Physical Review Letters*, 88(4), January 2002.

18. W. Habenicht. Quad trees: A data structure for discrete vector optimization problems. In *Lecture Notes in Economics and Mathematical Systems*, volume 209, pages 136–145, 1982.

19. P. Hajela and C. Y. Lin. Genetic search strategies in multicriterion optimal design. *Structural Optimization*, 4:99–107, 1992.

20. T. Hanne. On the convergence of multiobjective evolutionary algorithms. *European Journal of Operational Research*, 117(3):553–564, September 2000.

21. Mark Hinchliffe, Mark Willis, and Ming Tham. Chemical process systems modelling using multi-objective genetic programming. In John R. Koza, Wolfgang Banzhaf, Kumar Chellapilla, Kalyanmoy Deb, Marco Dorigo, David B. Fogel, Max H. Garzon, David E. Goldberg, Hitoshi Iba, and Rick L. Riolo, editors, *Proceedings of the Third Annual Conference on Genetic Programming*, pages 134–139, San Mateo, CA, July 1998. Morgan Kaufmann Publishers.

22. 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 Intelligence*, volume 1, pages 82–87, Piscataway, NJ, June 1994. IEEE Service Center.
23. W. Jakob, M. Gorges-Schleuter, and C. Blume. Application of genetic algorithms to task planning and learning. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature, 2nd Workshop*, Lecture Notes in Computer Science, pages 291–300, Amsterdam, 1992. North-Holland Publishing Company.
24. Andrzej Jaszkiewicz, Maciej Hapke, and Pawel Kominek. Performance of Multiple Objective Evolutionary Algorithms on a Distribution System Design Problem—Computational Experiment. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 241–255. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.
25. Yaochu Jin, Tatsuya Okabe, and Bernhard Sendhoff. Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization: Why Does It Work and How? In Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 1042–1049, San Francisco, California, 2001. Morgan Kaufmann Publishers.
26. Joshua D. Knowles and David W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
27. A. Kurapati and S. Azarm. Immune Network Simulation with Multiobjective Genetic Algorithms for Multidisciplinary Design Optimization. *Engineering Optimization*, 33:245–260, 2000.
28. J.N. Morse. Reducing the size of the nondominated set: Pruning by clustering. *Computers and Operations Research*, 7(1–2):55–66, 1980.
29. Shigeru Obayashi, Takanori Tsukahara, and Takashi Nakamura. Multiobjective genetic algorithm applied to aerodynamic design of cascade airfoils. *IEEE Transactions on Industrial Electronics*, 47(1), February 2000.
30. Andrzej Osyczka, Stanislaw Krenich, and K. Karaś. Optimum design of robot grippers using genetic algorithms. In *Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization (WCSMO)*, Buffalo, New York, May 1999.
31. Andrei Petrovski and John McCall. Multi-objective Optimisation of Cancer Chemotherapy Using Evolutionary Algorithms. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 531–545. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.
32. W. Pullan. Optimising Multiple Aspects of Network Survivability. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 115–120, Piscataway, New Jersey, May 2002. IEEE Service Center.
33. Patrick M. Reed, Barbara S. Minsker, and David E. Goldberg. A multiobjective approach to cost effective long-term groundwater monitoring using an elitist nondominated sorted genetic algorithm with historical data. *Journal of Hydroinformatics*, 3(2):71–89, April 2001.

34. Günter Rudolph and Alexandru Agapie. Convergence Properties of Some Multi-Objective Evolutionary Algorithms. In *Proceedings of the 2000 Conference on Evolutionary Computation*, volume 2, pages 1010–1016, Piscataway, NJ, July 2000. IEEE Press.

35. Ivo F. Sbalzarini, Sibylle Müller, and Petros Koumoutsakos. Microchannel Optimization Using Multiobjective Evolution Strategies. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 516–530. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

36. J. David Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100, Hillsdale, NJ, 1985. Lawrence Erlbaum.

37. N. Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.

38. El-Ghazali Talbi, Malek Rahoual, Mohamed Hakim Mabed, and Clarisse Dhaenens. A Hybrid Evolutionary Approach for Multicriteria Optimization Problems: Application to the Flow Shop. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 416–428. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

39. K. C. Tan, T. H. Lee, E. F. Khor, and K. Ou. Control system design unification and automation using an incremented multi-objective evolutionary algorithm. In M. H. Hamza, editor, *Proceedings of the 19th IASTED International Conference on Modeling, Identification and Control*. IASTED, Innsbruck, Austria, 2000.

40. Robert Thomson and Tughrul Arslan. An Evolutionary Algorithm for the Multi-Objective Optimisation of VLSI Primitive Operator Filters. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 37–42, Piscataway, New Jersey, May 2002. IEEE Service Center.

41. Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.

42. 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.

43. Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.