Advanced Information and Knowledge Processing

Series Editors
Professor Lakhmi Jain
Xindong Wu

Also in this series

Gregoris Mentzas, Dimitris Apostolou, Andreas Abecker and Ron Young
Knowledge Asset Management
1-85233-583-1

Michalis Vazirgiannis, Maria Halkidi and Dimitrios Gunopulos
Uncertainty Handling and Quality Assessment in Data Mining
1-85233-655-2

Asunción Gómez-Pérez, Mariano Fernández-López, Oscar Corcho
Ontological Engineering
1-85233-551-3

Arno Scharl (Ed.)
Environmental Online Communication
1-85233-783-4

Shichao Zhang, Chengqi Zhang and Xindong Wu
Knowledge Discovery in Multiple Databases
1-85233-703-6

Jason T.L. Wang, Mohammed J. Zaki, Hannu T.T. Toivonen and Dennis Shasha (Eds)
Data Mining in Bioinformatics
1-85233-671-4

C.C. Ko, Ben M. Chen and Jianping Chen
Creating Web-based Laboratories
1-85233-837-7

Manuel Graña, Richard Duro, Alicia d'Anjou and Paul P. Wang (Eds)
Information Processing with Evolutionary Algorithms
1-85233-886-0

Colin Fyfe
Hebbian Learning and Negative Feedback Networks
1-85233-883-0

Yun-Heh Chen-Burger and Dave Robertson
Automatic Business Modelling
1-85233-835-0

Dirk Husmeier, Richard Dybowski and Stephen Roberts (Eds)
Probabilistic Modeling in Bioinformatics and Medical Informatics
1-85233-778-8

K.C. Tan, E.F. Khor and T.H. Lee
Multiobjective Evolutionary Algorithms and Applications
1-85233-863-9

Ajith Abraham, Lakhmi Jain and
Robert Goldberg (Eds)

Evolutionary Multiobjective Optimization

Theoretical Advances and Applications

With 173 Figures

Springer

Ajith Abraham
Department of Computer Science, Oklahoma State University, USA

Lakhmi Jain
KES Center, University of South Australia, Australia

Robert Goldberg
Department of Computer Science, Queens College, New York, USA

Series Editors
Xindong Wu
Lakhmi Jain

British Library Cataloguing in Publication Data
Evolutionary multiobjective optimization : theoretical
Advances and applications. – (Advanced information and knowledge processing)
1. Mathematical optimization 2. Evolutionary computing
I. Abraham, Ajith, 1968– II. Jain, L.C. III. Goldberg, Robert, 1939–
519.6
ISBN 1852337877

Library of Congress Cataloging-in-Publication Data
Evolutionary multiobjective optimization : theoretical advances and applications /
[edited by] Ajith Abraham, Lakhmi Jain, Robert Goldberg.
p .cm. -- (Advanced information and knowledge processing)
Includes bibliographical references and index.
ISBN 1-85233-787-7 (alk. paper)
1. Evolutionary programming (Computer science) 2. Genetic algorithms. 3. Mathematical optimization. I. Abraham, Ajith, 1968– II. Jain, L. C. III. Goldberg, Robert, 1939– IV. Series

T57.E95 2004
005.1--dc22 2004052555

AI&KP ISSN 1610-3947
ISBN-10: 1-85233-787-7
ISBN-13: 978-1-85233-787-2
Springer Science+Business Media
springeronline.com

Printed in the United States of America

Typesetting: Electronic text files prepared by editors
34/3830-543210 Printed on acid-free paper SPIN 10968215

Preface

Multiobjective optimization has been available for about two decades, and its application in real world problems is continuously increasing. An important task in multiobjective optimization is to identify the set of Pareto-optimal solutions. An evolutionary algorithm is characterized by a population of solution candidates and the reproduction operator enables the process to combine existing solutions to generate new solutions. As a result, the computation finds several members of the Pareto-optimal set in a single run instead of performing a series of separate runs, which is the case for some of the conventional stochastic processes. 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 to prevent non-dominated solutions from being lost.

Addressing the various issues of evolutionary multiobjective optimization problems and the various design challenges using different intelligent approaches is the novelty of this edited volume. This volume comprises 12 chapters' including two introductory chapters giving the fundamental definitions and some important research challenges. Several complex test functions and a practical problem involving the multiobjective optimization of space structures under static and seismic loading conditions used to illustrate the importance of the evolutionary algorithm approach.

First, we would like to thank Lance Chambers (Australia) for initiating this book project in 2001. We are very much grateful to the authors of this volume and to the reviewers for their tremendous service by critically reviewing the chapters. The editors would like to thank Beverly Ford and Catherine Drury of Springer Verlag, London office and Jenny Wolkowicki of Springer Verlag, New York Office for the editorial assistance and excellent cooperative collaboration to produce this important scientific work. We are also indebted to Berend Jan

van der Zwaag (University of Twente, The Netherlands) for the tremendous help during the preparation of the manuscript. We hope that the reader will share our excitement to present this volume on 'Evolutionary Multiobjective Optimization: Theoretical Advances and Applications' and will find it useful.

Oklahoma State University, USA — Ajith Abraham
University of South Australia, Australia — Lakhmi Jain
Queens College, USA — Robert Goldberg
(Editors)

Contents

List of Contributors

Ajith Abraham (Ed.)
Department of Computer Science, Oklahoma State University, USA
ajith.abraham@ieee.org

Carlos A. Coello Coello
Depto. de Ingeniería Eléctrica, Sección de Computación,
Av. Instituto Politécnico Nacional No. 2508,
Col. San Pedro Zacatenco, D.F. 07300, México
ccoello@cs.cinvestav.mx

Kalyanmoy Deb
Kanpur Genetic Algorithms Laboratory (KanGAL),
Indian Institute of Technology Kanpur, PIN 208016, India
deb@iitk.ac.in

D. Dumitrescu
Department of Computer Science,
Faculty of Mathematics and Computer Science, University Babes-Bolyai,
Cluj-Napoca, 3400 Cluj – Napoca, Romania
ddumitr@cs.ubbcluj.ro

Robert Goldberg (Ed.)
Department of Computer Science, 65-30 Kissena Blvd.,
Queens College, Flushing, NY 11367, USA
rrg@solomon.cs.qc.edu

Crina Grosan
Department of Computer Science,
Faculty of Mathematics and Computer Science, University Babes-Bolyai,
Cluj-Napoca, 3400 Cluj – Napoca, Romania
cgrosan@cs.ubbcluj.ro

Natalie Hammerman
Department of Mathematics and Computer Science,
Molloy College, 1000 Hempstead Avenue, Rockville Centre, NY 11571, USA
nhammerman@molloy.edu

Hiro Hiroyasu
Department of Mechanical System Engineering,
Kinki University, Hiroshima, Japan
hiro@hiro.kindai.ac.jp

Tomoyuki Hiroyasu
Department of Knowledge Engineering and Computer Sciences,
Doshisha University, Tatara Miyakodani Kyotanabe-shi, Kyoto Japan
tomo@is.doshisha.ac.jp

Lakhmi Jain (Ed.)
University of South Australia, Adelaide, Australia
lakhmi.jain@unisa.edu.au

Jiro Kamiura
Department of Knowledge Engineering and Computer Sciences,
Doshisha University, Tatara Miyakodani Kyotanabe-shi, Kyoto Japan

Nikos D. Lagaros
Institute of Structural Analysis & Seismic Research
National Technical University Athens
Zografou Campus, Athens 157 80, Greece
nlagaros@central.ntua.gr

Marco Laumanns
Swiss Federal Institute of Technology (ETH) Zürich,
Computer Engineering and Networks Laboratory (TIK),
CH-8092 Zürich, Switzerland
laumanns@tik.ee.ethz.ch

Efrén Mezura-Montes
Computer Science Section, Electrical Engineering Department
CINVESTAV-IPN, Av. Instituto Politécnico Nacional No. 2508
Col. San Pedro Zacatenco, México, D.F. 07300
emezura@computacion.cs.cinvestav.mx

Mitsunori Miki
Department of Knowledge Engineering and Computer Sciences,
Doshisha University, Tatara Miyakodani Kyotanabe-shi, Kyoto Japan
mmiki@mail.doshisha.ac.jp

Sanaz Mostaghim
Electrical Engineering Department,
University of Paderborn, Paderborn, Germany
mostaghim@date.upb.de

Christine L. Mumford-Valenzuela
School of Computer Science, Cardiff University, United Kingdom
christine@cs.cardiff.ac.uk

Mihai Oltean
Department of Computer Science,
Faculty of Mathematics and Computer Science, University Babes-Bolyai,
Cluj-Napoca, 3400 Cluj – Napoca, Romania
molteang@cs.ubbcluj.ro

Manolis Papadrakakis
Institute of Structural Analysis & Seismic Research
National Technical University Athens
Zografou Campus, Athens 157 80, Greece
mpapadra@central.ntua.gr

Vagelis Plevris
Institute of Structural Analysis & Seismic Research
National Technical University Athens
Zografou Campus, Athens 157 80, Greece
vplevrisg@central.ntua.gr

Tian Hou Seow
10 Kent Ridge Crescent, Singapore 119260
Department of Electrical & Computer Engineering
National University of Singapore

Dipti Srinivasan
10 Kent Ridge Crescent, Singapore 119260
Department of Electrical & Computer Engineering
National University of Singapore

Jürgen Teich
Computer Science Department, Friedrich-Alexander-University,
Erlangen, Germany
teich@informatik.uni-erlangen.de

Lothar Thiele
Swiss Federal Institute of Technology (ETH) Zürich
Computer Engineering and Networks Laboratory (TIK),
CH-8092 Zürich, Switzerland
thiele@tik.ee.ethz.ch

Shinya Watanabe
Department of Knowledge Engineering and Computer Sciences,
Doshisha University, Tatara Miyakodani Kyotanabe-shi, Kyoto Japan

Eckart Zitzler
Swiss Federal Institute of Technology (ETH) Zürich
Computer Engineering and Networks Laboratory (TIK),
CH-8092 Zürich, Switzerland
zitzlerg@tik.ee.ethz.ch

1

Evolutionary Multiobjective Optimization

Ajith Abraham and Lakhmi Jain

Summary. Very often real-world applications have several multiple conflicting objectives. Recently there has been a growing interest in evolutionary multiobjective optimization algorithms that combine two major disciplines: evolutionary computation and the theoretical frameworks of multicriteria decision making. In this introductory chapter, some fundamental concepts of multiobjective optimization are introduced, emphasizing the motivation and advantages of using evolutionary algorithms. We then lay out the important contributions of the remaining chapters of this volume.

1.1 What Is Multiobjective Optimization?

Even though some real-world problems can be reduced to a matter of a 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 its application in real-world problems is continuously 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 could be written in the form minimize $[f_1(x), f_2(x),\ldots, f_k(x)]$ for k objective functions f_i: $\Re^n \rightarrow \Re$ subject to several equality and inequality constraints. For $x = [x_1, x_2,\ldots, x_n]^T$, the vector of decision variables, our task is to determine the set F of all vectors which satisfy all the constraints, the particular set of values $[x_1^*, x_2^*..., x_n^*]$ and also yields the optimum values for all the objective functions [1].

As shown in Figure 1.1, 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 a Pareto-optimal and the entire set of such optimal trade-offs solutions is called a 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.

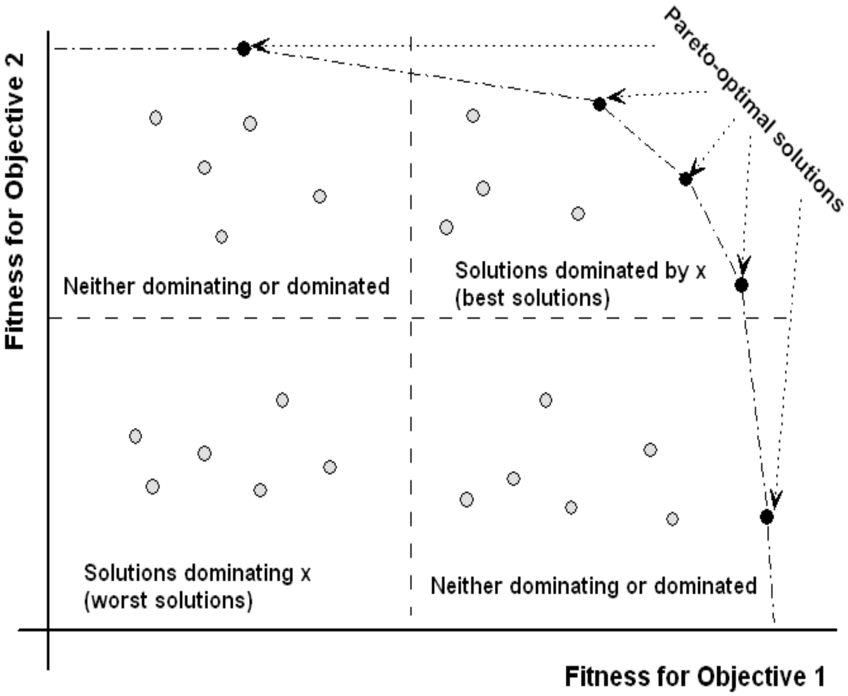

Figure 1.1. Concept of Pareto optimality.

1.2 Why Use Evolutionary Algorithms for Multiobjective Optimization?

A number of stochastic optimization techniques such as 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 the combination of existing solutions to generate new solutions. This enables finding several members of the Pareto-optimal set in a single run instead of performing a series of separate runs, which is the case for some of the conventional stochastic processes. Finally, natural selection determines which individuals of the current population participate in the new population.

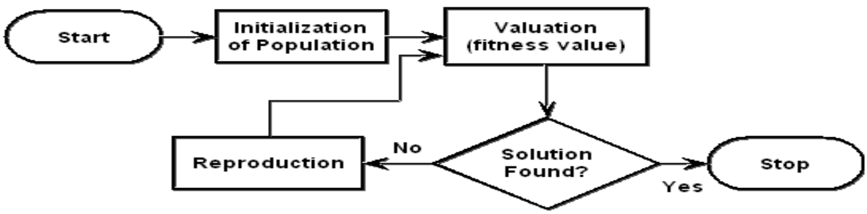

Figure 1.2. Flowchart of evolutionary algorithm iteration.

The iterative computation process of an evolutionary algorithm is illustrated in Figure 1.2. 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 by Schaffer in the mid-1980s [2], several different algorithms have been proposed and successfully applied to various problems. For comprehensive overviews and discussions, the reader is referred to Coello Coello [1] and Deb [3]. Some of the other advantages of having evolutionary algorithms is that they require very little knowledge about the problem being solved, less susceptible to the shape or continuity of the Pareto front, easy to implement, robust, and could be implemented in a parallel environment.

1.3 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 to prevent non-dominated solutions from being lost [4, 5]. Elitism addresses the problem of losing good solutions during the optimization process. 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 12 chapters and the rest of the volume is organized as follows.

In the following Chapter, Coello Coello presents the basic concepts of evolutionary multiobjective algorithms, their potential applications, 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-Valenzuela explains a Pareto-based approach to evolutionary multi-objective optimization, that avoids most of the time-consuming global calculations typical of other multiobjective 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 performance 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. The 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 multiobjective 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. suggest 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 present a 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 a 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 illustrate 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 the Pareto-optimal set. The proposed method is evaluated using some test functions.

Hiroyasu et al. in Chapter 9 address 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 evolutionary algorithm for multiobjective 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 Multiobjective 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 Multiobjective Genetic Algorithm with Environmental Scheme (DCMOGADES). The performance of DCMOGADES is illustrated using some test functions.

In Chapter 10, Mezura-Montes and Coello Coello describe the general multiobjective optimization concepts that can be used to incorporate constraints of any type (linear, non-linear, 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% to 46% smaller in size than those found by the original GA.

In the last chapter, Lagaros et al. deal 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.

References

1. Coello Coello, CA, Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques, International Journal of Knowledge and Information Systems, 1999; 1(3):269-308.
2. Schaffer, JD, Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, PhD Thesis, Vanderbilt University, Nashville, USA, 1984.
3. Deb, K, Multiobjective Optimization using Evolutionary Algorithms, John Wiley & Sons, Chichester, 2001.
4. Laumanns, M, Thiele, L, Deb, K and Zitzler, E, Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. Evolutionary Computation, MIT Press, 2002; 10(3):263-282.
5. Zitzler, E, Laumanns, M and Bleuler, S, A Tutorial on Evolutionary Multiobjective Optimization, Workshop on Multiple Objective Metaheuristics (MOMH), Springer-Verlag, Berlin, 2004.

2

Recent Trends in Evolutionary Multiobjective Optimization

Carlos A. Coello Coello

Summary. This chapter presents a brief review of some of the most relevant research currently taking place in evolutionary multiobjective optimization. The main topics covered include algorithms, applications, metrics, test functions, and theory. Some of the most promising future paths of research are also addressed.

2.1 Introduction

Evolutionary algorithms (EAs) are heuristics that use natural selection as their search engine to solve problems. The use of EAs for search and optimization tasks has become very popular in the last few years with a constant development of new algorithms, theoretical achievements and novel applications [1, 2, 3]. 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 multiobjective evolutionary algorithm (MOEA) dates back to the mid-1980s [4, 5]. Since then, a considerable amount of research has been done in this area, now known as evolutionary multiobjective optimization (EMO for short). The growing importance of this field is reflected by a significant increment (mainly during the last ten years) of technical papers in international conferences and peer-reviewed journals, books, special sessions at international conferences and interest groups on the Internet [6].[1]

---

[1]The author maintains an EMO repository which currently contains over 1450 bibliographical entries at: `http://delta.cs.cinvestav.mx/~ccoello/EMOO`, with a mirror at `http://www.lania.mx/~ccoello/EMOO/`

The main motivation for using EAs to solve multiobjective optimization problems is because EAs 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 [7]. Additionally, EAs 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, 8, 9].

2.2 Basic Concepts

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

$$\text{minimize } [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})] \tag{2.1}$$

subject to the m inequality constraints:

$$g_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \ldots, m \tag{2.2}$$

and the p equality constraints:

$$h_i(\mathbf{x}) = 0 \quad i = 1, 2, \ldots, p \tag{2.3}$$

where k is the number of objective functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$. We call $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ the vector of decision variables. We wish to determine from among the set $\mathcal{F}$ of all vectors which satisfy Equations 2.2 and 2.3 the particular set of values $x_1^*, x_2^*, \ldots, x_n^*$ which yield the optimum values of all the objective functions.

2.2.1 Pareto Optimality

It is rarely the case that there is a single point that simultaneously optimizes all the objective functions of a multiobjective optimization problem. 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 that originally proposed by Francis Ysidro Edgeworth [10] and later generalized by Vilfredo Pareto [11]. Although some authors call this notion *Edgeworth-Pareto optimality* (see for example [12]), we will use the most commonly accepted term: *Pareto optimality.*

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 *non-dominated*. The image of the Pareto-optimal set under the objective functions is called *Pareto front*.

2.3 Algorithms

The potential of evolutionary algorithms for solving multiobjective optimization problems was hinted at in the late 1960s by Rosenberg in his PhD thesis [13]. Rosenberg's study contained a suggestion that would have led to multiobjective optimization if he had carried it out as presented. His suggestion was to use multiple *properties* (nearness to some specified chemical composition) in his simulation of the genetics and chemistry of a population of single-celled organisms. Since his actual implementation contained only one single property, the multiobjective approach could not be shown in his work.

The first actual implementation of what is now called a multiobjective evolutionary algorithm (or MOEA, for short) was Schaffer's *Vector Evaluation Genetic Algorithm* (VEGA), which was introduced in the mid-1980s, mainly aimed at solving problems in machine learning [4, 5, 14]. Since then, a wide variety of algorithms have been proposed in the literature [6, 8, 15].

We can roughly divide MOEAs into the following types:

- Aggregating functions
- Population-based approaches
- Pareto-based approaches

We will briefly discuss each of them in the following subsections.

2.3.1 Aggregating Functions

Perhaps the most straightforward approach to handling multiple objectives with any technique is to combine all the objectives into a single one using either an addition, multiplication or any other combination of arithmetical operations. These techniques are normally known as "aggregating functions" because they combine (or "aggregate") all the objectives of the problem into a single one. In fact, aggregating approaches are the oldest mathematical programming methods for multiobjective optimization, since they can be derived from the Kuhn-Tucker conditions for non-dominated solutions [16].

An example of this approach is a linear sum of weights of the form:

$$\min \sum_{i=1}^{k} w_i f_i(\mathbf{x}) \tag{2.4}$$

where $w_i \geq 0$ are the weighting coefficients representing the relative importance of the k objective functions of our problem. It is usually assumed that

$$\sum_{i=1}^{k} w_i = 1 \ . \tag{2.5}$$

Aggregating functions may be linear (as in the previous example) or nonlinear (e.g., the aggregating functions adopted by game theory [17, 18], goal programming [19, 20], goal attainment [21, 22] and the min-max algorithm [23, 24]). Both types of aggregating functions have been used with evolutionary algorithms in a number of occasions, with relative success.

Aggregating functions have been largely underestimated by EMO researchers mainly because of the well-known limitation of linear aggregating functions (i.e., they cannot generate non-convex portions of the Pareto front regardless of the weight combination used [25]). Note, however, that non-linear aggregating functions do not necessarily present such limitation [6]. In fact, even linear aggregating functions can be cleverly defined such that concave Pareto fronts can be generated [26]. However, the EMO community tends to show little interest in new algorithms based on aggregating functions, hence their relatively low popularity among EMO researchers.

2.3.2 Population-based Approaches

In these techniques, the population of an EA is used to diversify the search, but the concept of Pareto dominance is not directly incorporated into the selection process. The classical example of this sort of approach is the Vector Evaluated Genetic Algorithm (VEGA), proposed by Schaffer [5]. VEGA basically consists of a simple genetic algorithm with a modified selection mechanism. At each generation, a number of sub-populations are 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 are generated (assuming a total population size of M). These sub-populations are then shuffled together to obtain a new population of size M, on which the genetic algorithm (GA) applies the crossover and mutation operators.

VEGA has several problems, from which the most serious is that its selection scheme is opposed to the concept of Pareto dominance. If, for example, there is an individual that encodes a good compromise solution for all the objectives, but it is not the best in any of them, it will be discarded. Note however, that such individual should really be preserved because it encodes a Pareto-optimal solution. Schaffer suggested some heuristics to deal with this problem. For example, to use a heuristic selection preference approach for non-dominated individuals in each generation, to protect individuals that encode

Pareto-optimal solutions but are not the best in any single objective function. Also, crossbreeding among the "species" could be encouraged by adding some mate selection heuristics instead of using the random mate selection of the traditional GA. Nevertheless, the fact that Pareto dominance is not directly incorporated into the selection process of the algorithm remains its main disadvantage.

One interesting aspect of VEGA is that despite its drawbacks it remains in current use by some researchers mainly because it is appropriate for problems in which we want the selection process to be biased and in which we have to deal with a large number of objectives (e.g., when handling constraints as objectives in single-objective optimization [27]).

Other researchers have proposed variations of VEGA or other similar population-based approaches (e.g., [28, 29, 30, 31]). Despite the limitations of these approaches, their simplicity has attracted several researchers and we should expect to see more work on population-based approaches in the next few years.

2.3.3 Pareto-based Approaches

Taking as a basis the main drawbacks of VEGA, Goldberg discussed on pages 199 to 201 of his famous book on genetic algorithms [1] a way of tackling multiobjective problems. His procedure consists of a selection scheme based on the concept of Pareto optimality. Goldberg not only suggested what would become the standard MOEA for several years, but also indicated that stochastic noise would make such algorithms useless unless some special mechanism was adopted to block convergence. Niching or fitness sharing [32] was suggested by Goldberg as a way to maintain diversity and avoid convergence of the GA to a single solution.

Pareto-based approaches can be historically studied as covering two generations. The first generation is characterized by the use of fitness sharing and niching combined with Pareto ranking (as defined by Goldberg or adopting a slight variation). The most representative algorithms from the first generation are the following:

1. **Non-dominated Sorting Genetic Algorithm** (NSGA): This algorithm was proposed by Srinivas and Deb [33]. The approach is based on several layers of classifications of the individuals as suggested by Goldberg [1]. Before selection is performed, the population is ranked on the basis of non-domination: all non-dominated individuals are classified into one category (with a dummy fitness value, which is proportional to the population size, to provide an equal reproductive potential for these individuals). To maintain the diversity of the population, these classified individuals are shared with their dummy fitness values. Then this group of classified individuals is ignored and another layer of non-dominated individuals is considered. The process continues until all individuals in the

population are classified. Stochastic remainder proportionate selection is adopted for this technique. 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 non-dominated regions, and results in convergence of the population toward such regions. Sharing, 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. [34]. 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 non-dominated individual wins. When both competitors are either dominated or non-dominated (i.e., there is a tie), the result of the tournament is decided through fitness sharing [35].

3. **Multiobjective Genetic Algorithm** (MOGA): Proposed by Fonseca and Fleming [36]. 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 [36]:

   $$\mathrm{rank}(x_i, t) = 1 + p_i^{(t)} \ . \tag{2.6}$$

   All non-dominated 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 [36]:
   a) Sort population according to rank.
   b) Assign fitness to individuals by interpolating from the best (rank 1) to the worst (rank $n \leq M$) 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 second generation of MOEAs was born with the introduction of the notion of elitism. In the context of multiobjective optimization, elitism usually (althought not necessarily) refers to the use of an external population (also called secondary population) to retain the non-dominated individuals. However, the use of this external 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?

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

The most representative second generation MOEAs are the following:

1. **Strength Pareto Evolutionary Algorithm** (SPEA): This algorithm was introduced by Zitzler and Thiele [37]. This approach was conceived as a way of integrating different MOEAs. SPEA uses an archive containing non-dominated solutions previously found (the so-called external non-dominated set). At each generation, non-dominated individuals are copied to the external non-dominated set. For each individual in this external set, a *strength* value is computed. This strength is similar to the ranking value of MOGA, since it is proportional to the number of solutions to which a certain individual dominates.
   In SPEA, the fitness of each member of the current population is computed according to the strengths of all external non-dominated solutions that dominate it. Additionally, a clustering technique called "average linkage method" [38] is used to keep diversity.

2. **Strength Pareto Evolutionary Algorithm 2** (SPEA2): This approach has three main differences with respect to its predecessor [39]: (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.

3. **Pareto Archived Evolution Strategy** (PAES): This algorithm was introduced by Knowles and Corne [40]. 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 non-dominated solutions previously found. This archive is used as a reference set against which each mutated individual is being compared. An interesting aspect of this algorithm is its procedure used to maintain diversity which consists of a crowding procedure that divides objective space in a recursive manner. Each solution is placed in a certain grid location based on the values of its objectives (which are used as its "coordinates" or "geographical location"). A map of this grid is