An Introduction to Evolutionary Multiobjective Optimization

Carlos A. Coello Coello

CINVESTAV-IPN

Departamento de Computación

Av. Instituto Politécnico Nacional No. 2508

Col. San Pedro Zacatenco

México, D. F. 07360, MEXICO

email: ccoello@cs.cinvestav.mx

Motivation

Most problems in nature have several (possibly conflicting) objectives to be satisfied. Many of these problems are frequently treated as single-objective optimization problems by transforming all but one objective into constraints.

What is a multiobjective optimization problem?

The **Multiobjective Optimization Problem** (MOP) (also called multicriteria optimization, multiperformance or vector optimization problem) can be defined (in words) as the problem of finding (Osyczka, 1985):

> a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term "optimize" means finding such a solution which would give the values of all the objective functions acceptable to the decision maker.

A Formal Definition

The general Multiobjective Optimization Problem (MOP) can be formally defined as:

Find the vector $\vec{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ which will satisfy the m inequality constraints:

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

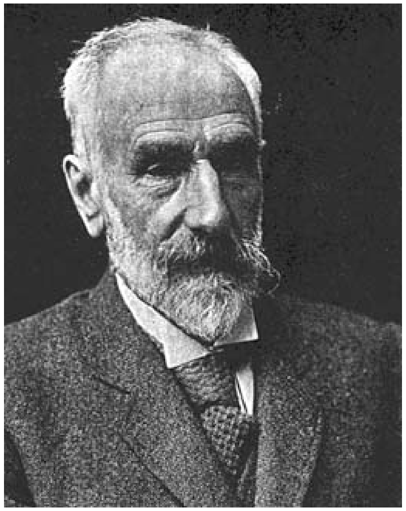
the p equality constraints

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

and will optimize the vector function

$$\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})]^T \tag{3}$$

What is the notion of optimum in multiobjective optimization?



Having several objective functions, the notion of "optimum" changes, because in MOPs, we are really trying to find good compromises (or "trade-offs") rather than a single solution as in global optimization. The notion of "optimum" that is most commonly adopted is that originally proposed by Francis Ysidro Edgeworth in 1881.

What is the notion of optimum in multiobjective optimization?

This notion was later generalized by Vilfredo Pareto (in 1896). Although some authors call *Edgeworth-Pareto optimum* to this notion, we will use the most commonly accepted term: *Pareto optimum.*

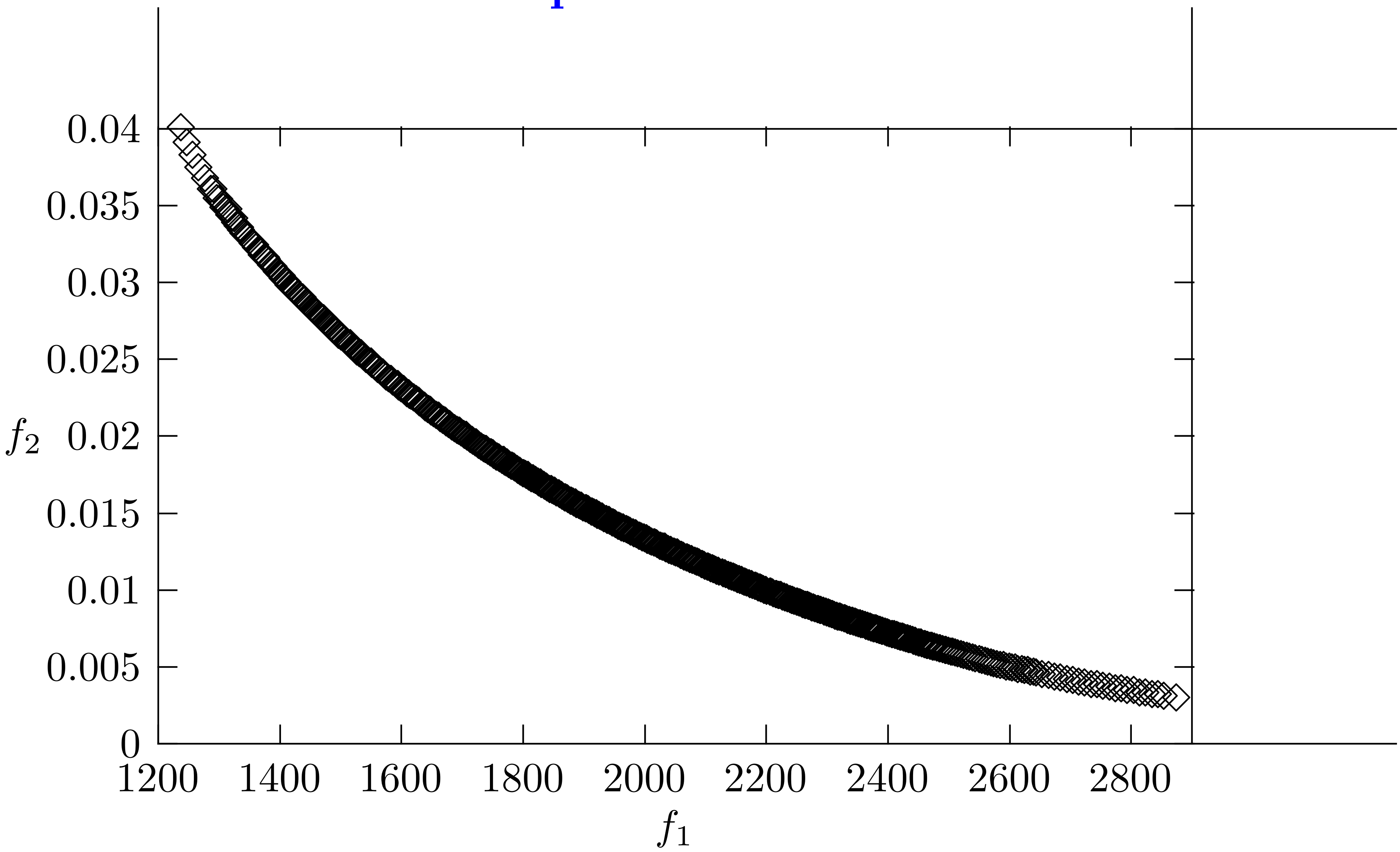
Definition of Pareto Optimality

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

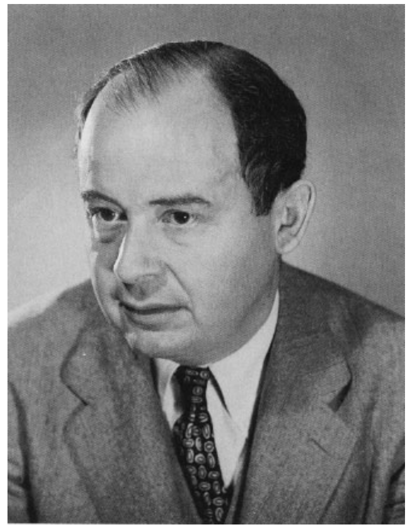
Definition of Pareto Optimality

In words, this definition says that $\vec{x}^*$ is Pareto optimal if there exists no feasible vector of decision variables $\vec{x} \in \mathcal{F}$ 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 $\vec{x}^*$ correspoding to the solutions included in the Pareto optimal set are called *nondominated*. The plot of the objective functions whose nondominated vectors are in the Pareto optimal set is called the *Pareto front*.

Sample Pareto Front



Some Historical Highlights



As early as 1944, John von Neumann and Oskar Morgenstern mentioned that an optimization problem in the context of a social exchange economy was "a peculiar and disconcerting mixture of several conflicting problems" that was "nowhere dealt with in classical mathematics".

Some Historical Highlights

In 1951 Tjalling C. Koopmans edited a book called *Activity Analysis of Production and Allocation*, where the concept of "efficient" vector was first used in a significant way.

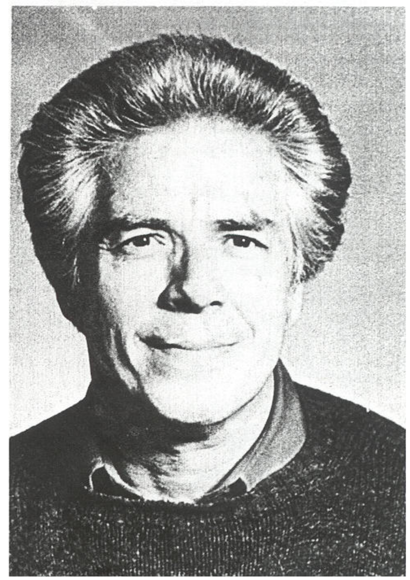
Some Historical Highlights

The origins of the mathematical foundations of multiobjective optimization can be traced back to the period that goes from 1895 to 1906. During that period, Georg Cantor and Felix Hausdorff laid the foundations of infinite dimensional ordered spaces.

Some Historical Highlights

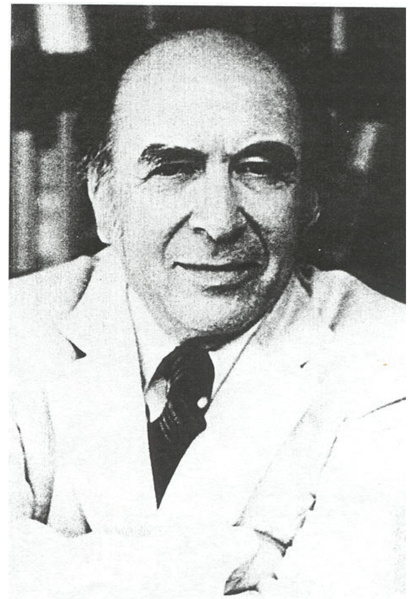
Cantor also introduced equivalence classes and stated the first sufficient conditions for the existence of a utility function. Hausdorff also gave the first example of a complete ordering.

Some Historical Highlights



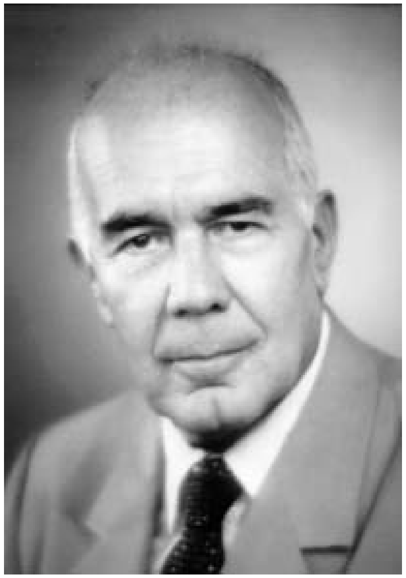
However, it was the concept of *vector maximum problem* introduced by Harold W. Kuhn and Albert W. Tucker (1951) which made multiobjective optimization a mathematical discipline on its own.

Some Historical Highlights



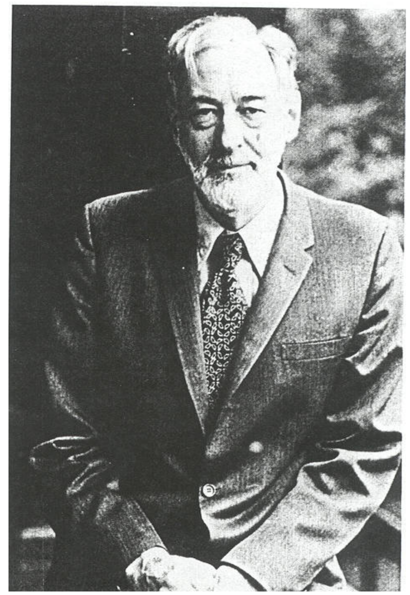
However, multiobjective optimization theory remained relatively undeveloped during the 1950s. It was until the 1960s that the foundations of multiobjective optimization were consolidated and taken seriously by pure mathematicians when Leonid Hurwicz generalized the results of Kuhn & Tucker to topological vector spaces.

Antecedentes Históricos



Perhaps the most important research outcome from the 1950s was the development of Goal Programming, which was originally introduced by Abraham Charnes and William Wager Cooper in 1957.

Some Historical Highlights

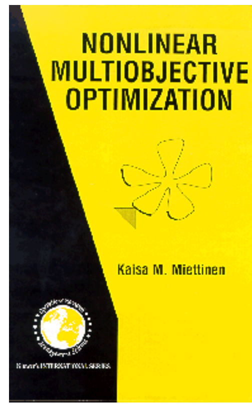


The application of multiobjective optimization to domains outside economics began with the work by Koopmans (1951) in production theory and with the work of Marglin (1967) in water resources planning.

Some Historical Highlights

The first engineering application reported in the literature was a paper by Zadeh in the early 1960s. However, the use of multiobjective optimization became generalized until the 1970s.

Current State of the Area



Currently, there are over 30 mathematical programming techniques for multiobjective optimization. However, these techniques tend to generate elements of the Pareto optimal set one at a time. Additionally, most of them are very sensitive to the shape of the Pareto front (e.g., they do not work when the Pareto front is concave or when the front is disconnected).

Why Evolutionary Algorithms?

Evolutionary algorithms seem particularly suitable to solve multiobjective optimization problems, because they deal simultaneously with a set of possible solutions (the so-called population). This 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 or concave Pareto fronts), whereas these two issues are a real concern for mathematical programming techniques.

Historical Development

The potential of evolutionary algorithms in multiobjective optimization was hinted by Rosenberg in the 1960s, but the first actual implementation was produced in the mid-1980s (Schaffer, 1984). The first scheme to incorporate user's preferences into a multi-objective evolutionary algorithm (MOEA) was proposed in the early 1990s (Tanaka, 1992). During ten years, the field remain practically inactive, but it started growing in the mid-1990s, in which several techniques and applications were developed.

Evolutionary Algorithms

We can consider, in general, two main types of MOEAs:

1. Algorithms that do not incorporate the concept of Pareto dominance in their selection mechanism (e.g., approaches that use linear aggregating functions).
2. Algorithms that rank the population based on Pareto dominance. For example, MOGA, NSGA, NPGA, etc.

Evolutionary Algorithms

Historically, we can consider the existence of two main generations of MOEAs:

1. **First Generation**: Characterized by the use of Pareto ranking and niching (or fitness sharing). Relatively simple algorithms. Other (more rudimentary) approaches were also developed (e.g., linear aggregating functions). It is also worth mentioning VEGA, which is a population-based (not Pareto-based) approach.

2. **Second Generation**: The concept of elitism is introduced in two main forms: using $(\mu + \lambda)$ selection and using a secondary (external) population.

Representative MOEAs (First Generation)

- Aggregating Functions
- VEGA
- MOGA
- NSGA
- NPGA & NPGA 2

Aggregating Functions

- These techniques are called "aggregating functions" because they combine (or "aggregate") all the objectives into a single one. We can use addition, multiplication or any other combination of arithmetical operations.
- Oldest mathematical programming method, since aggregating functions can be derived from the Kuhn-Tucker conditions for nondominated solutions.

Aggregating Functions

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

$$\min \sum_{i=1}^{k} w_i f_i(\vec{x}) \tag{4}$$

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

$$\sum_{i=1}^{k} w_i = 1 \tag{5}$$

Advantages and Disadvantages

- Easy to implement
- Efficient
- Linear combinations of weights do not work when the Pareto front is concave, regardless of the weights used (Das, 1997). Note however, that the weights can be generated in such a way the the Pareto front is rotated (Jin et al., 2001). In this last case, concave Pareto fronts can be efficiently generated.

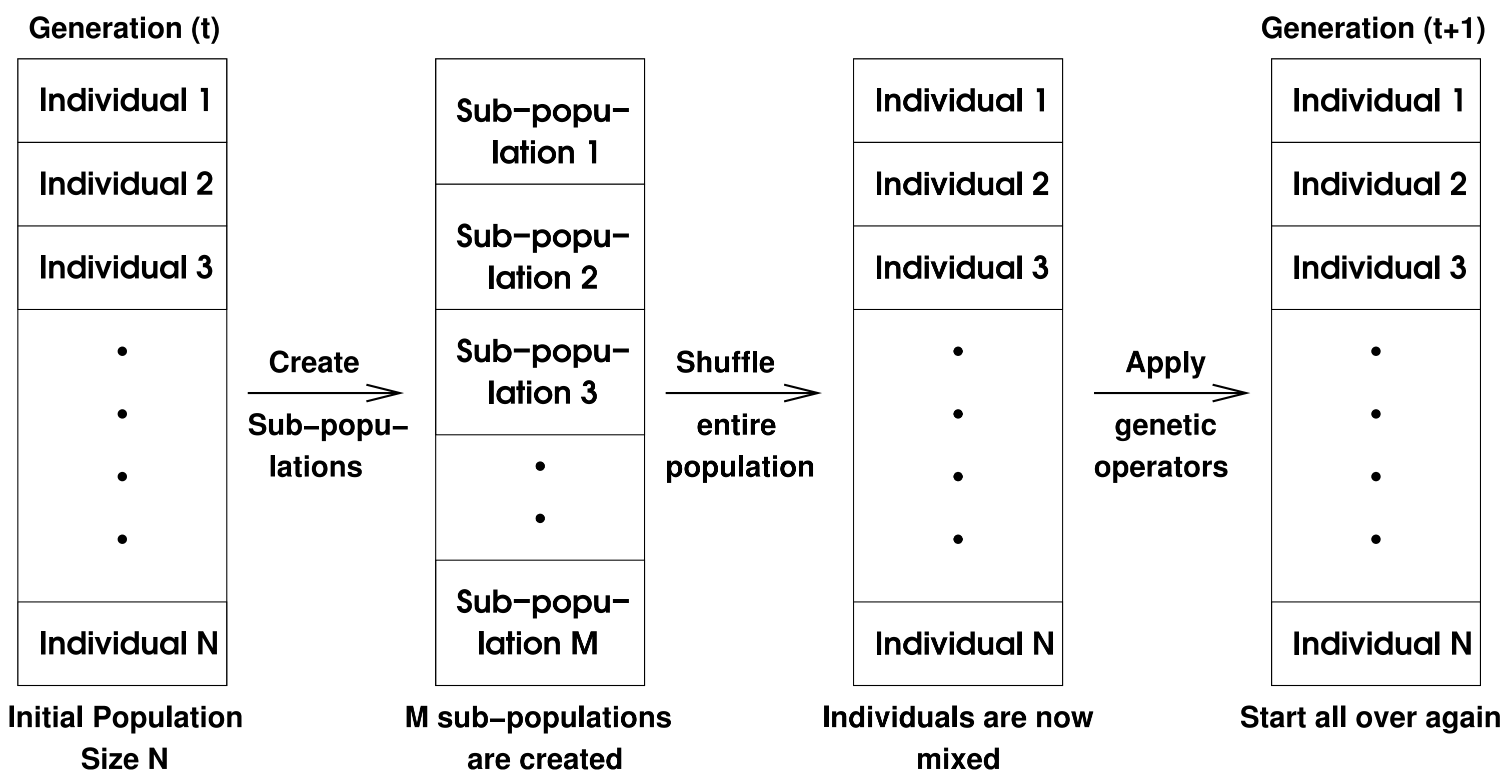
Sample Applications

- Design of DSP systems (Arslan, 1996).
- Water quality control (Garrett, 1999).
- System-level synthesis (Blickle, 1996).
- Design of optical filters for lamps (Eklund, 1999).

Vector Evaluated Genetic Algorithm (VEGA)

- Proposed by Schaffer in the mid-1980s (1984,1985).
- It uses subpopulations that optimize each objective separately. The concept of Pareto optimum is not directly incorporated into the selection mechanism of the GA.

VEGA



Generation (t)
Individual 1
Individual 2
Individual 3
Individual N
Initial Population
Size N
Create
Sub–popu–
lations
Sub–popu–
lation 1
Sub–popu–
lation 2
Sub–popu–
lation 3
Sub–popu–
lation M
M sub–populations
are created
Shuffle
entire
population
Individual 1
Individual 2
Individual 3
Individual N
Individuals are now
mixed
Apply
genetic
operators
Generation (t+1)
Individual 1
Individual 2
Individual 3
Individual N
Start all over again

Advantages and Disadvantages

- Efficient and easy to implement.
- It doesn't have an explicit mechanism to maintain diversity. It doesn't necessarily produce nondominated vectors.

Sample Applications

- Aerodynamic optimization (Rogers, 2000).
- Combinational circuit design at the gate-level (Coello, 2000).
- Design multiplierless IIR filters (Wilson, 1993).
- Groundwater pollution containment (Ritzel, 1994).

Multi-Objective Genetic Algorithm (MOGA)

- Proposed by Fonseca and Fleming (1993).
- The approach consists of a scheme in which the rank of a certain individual corresponds to the number of individuals in the current population by which it is dominated.
- It uses fitness sharing and mating restrictions.

Advantages and Disadvantages

- Efficient and relatively easy to implement.
- Its performance depends on the appropriate selection of the sharing factor.
- MOGA was the most popular first-generation MOEA and it normally outperformed all of its contemporary competitors.

Some Applications

- Fault diagnosis (Marcu, 1997).
- Control system design (Chipperfield 1995; Whidborne, 1995; Duarte, 2000).
- Design of antennas (Thompson, 2001).
- System-level synthesis (Dick, 1998).
- Rehabilitation of a water distribution system (Cheung, 2003).
- Forest management (Ducheyne, 2003)

Advantages and Disadvantages

- Relatively easy to implement.
- Seems to be very sensitive to the value of the sharing factor.

Sample Applications

- Water quality control (Reed et al., 2001).
- Design of control systems (Blumel, 2001).
- Constellation design (Mason, 1999).
- Computational fluid dynamics (Marco, 1999).
- Chemical engineering (Kasat, 2003).

Niched-Pareto Genetic Algorithm (NPGA)

- Proposed by Jeffrey Horn et al. (1993,1994).
- It uses a tournament selection scheme based on Pareto dominance. Two individuals randomly chosen are compared against a subset from the entire population (typically, around 10% of the population). When both competitors are either dominated or nondominated (i.e., when there is a tie), the result of the tournament is decided through fitness sharing in the objective domain (a technique called *equivalent class sharing* is used in this case).

Advantages and Disadvantages

- Easy to implement.
- Efficient because does not apply Pareto ranking to the entire population.
- It seems to have a good overall performance.
- Besides requiring a sharing factor, it requires another parameter (tournament size).

Sample Applications

- Analysis of experimental spectra (Golovkin, 2000).
- Feature selection (Emmanouilidis, 2000).
- Fault-tolerant systems design (Schott, 1995).
- Road systems design (Haastrup & Pereira, 1997).

NPGA 2

Erickson et al. (2001) proposed the NPGA 2, which uses Pareto ranking but keeps tournament selection (solving ties through fitness sharing as in the original NPGA).

Niche counts in the NPGA 2 are calculated using individuals in the partially filled next generation, rather than using the current generation. This is called continuously updated fitness sharing, and was proposed by Oei et al. (1991).

Sample Applications

- Design of groundwater remediation systems (Erickson et al., 2001).

Representative MOEAs (Second Generation)

- SPEA and SPEA2
- NSGA-II
- PAES, PESA and PESA II
- MOMGA and MOMGA II
- The microGA for multiobjective optimization and the microGA2

The Strength Pareto Evolutionary Algorithm (SPEA)

SPEA was introduced by Zitzler & Thiele (1999). It uses an external archive containing nondominated solutions previously found. SPEA computes a strength value similar to the ranking value used by MOGA. A clustering technique called "average linkage method" is used to keep diversity.

Sample Applications

- Exploration of trade-offs of software implementations for DSP algorithms (Zitzler, 1999)
- Treatment planning (Petrovski, 2001)
- Allocation in radiological facilities (Lahanas, 2001)
- Atrial disease diagnosis (de Toro, 2003).
- Rehabilitation of a water distribution system (Cheung, 2003).

The Strength Pareto Evolutionary Algorithm 2 (SPEA2)

A revised version of SPEA has been recently proposed: SPEA2 (Zitzler, 2001). SPEA2 has three main differences with respect to its predecessor: (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.

Sample Applications

- Control code size and reduce bloat in genetic programming (Bleuler, 2001).
- Airfoil design (Willmes, 2003).

The Nondominated Sorting Genetic Algorithm II (NSGA-II)

Deb et al. (2000,2002) proposed a new version of the Nondominated Sorting Genetic Algorithm (NSGA), called NSGA-II, which is more efficient (computationally speaking), it uses elitism and a crowded comparison operator that keeps diversity without specifying any additional parameters.

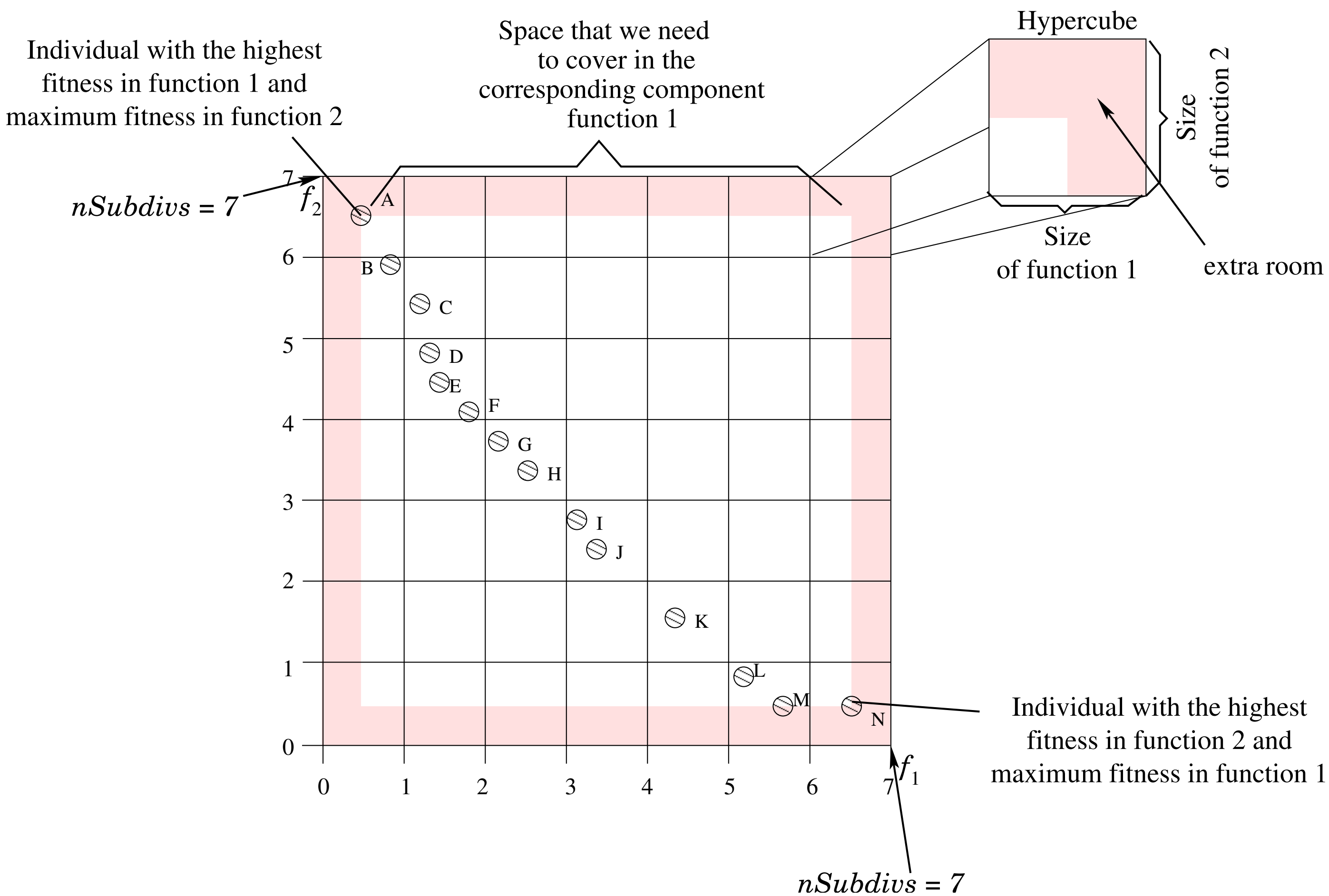
Sample Applications

- Shape optimization (Deb, 2001).
- Safety systems design (Greiner, 2003).
- Polymer extrusion (Gaspar-Cunha, 2003).
- Water quality management (Dorn, 2003).
- Intensity modulated beam radiation therapy (Lahanas, 2003).

The Pareto Archived Evolution Strategy (PAES)

PAES was introduced by Knowles & Corne (2000). It uses a (1+1) evolution strategy together with an external archive that records all the nondominated vectors previously found. PAES uses an adaptive grid to maintain diversity.

PAES (Adaptive Grid)



Sample Applications

- Off-line routing problem (Knowles, 1999)
- Adaptive distributed database management problem (Knowles, 2000)

The Pareto Envelope-based Selection Algorithm (PESA)

PESA was proposed by Corne et al. (2000). This approach uses a small internal population and a larger external (or secondary) population. PESA uses the same hyper-grid division of phenotype (i.e., objective funcion) space adopted by PAES to maintain diversity. However, its selection mechanism is based on the crowding measure used by the hyper-grid previously mentioned. This same crowding measure is used to decide what solutions to introduce into the external population (i.e., the archive of nondominated vectors found along the evolutionary process).

Sample Applications

- Telecommunications problems (Corne et al., 2000).

The Pareto Envelope-based Selection Algorithm-II (PESA-II)

PESA-II (Corne et al., 2001) is a revised version of PESA in which region-based selection is adopted. In region-based selection, the unit of selection is a hyperbox rather than an individual. The procedure consists of selecting (using any of the traditional selection techniques) a hyperbox and then randomly select an individual within such hyperbox.

Sample Applications

- Telecommunications problems (Corne et al., 2001).

The Multi-Objective Messy Genetic Algorithm (MOMGA)

MOMGA was proposed by Van Veldhuizen and Lamont (2000). This is an attempt to extend the messy GA to solve multiobjective optimization problems.

MOMGA consists of three phases: (1) Initialization Phase, (2) Primordial Phase, and (3) Juxtapositional Phase. In the *Initialization Phase*, MOMGA produces all building blocks of a certain specified size through a deterministic process known as partially enumerative initialization. The *Primordial Phase* performs tournament selection on the population and reduces the population size if necessary. In the *Juxtapositional Phase*, the messy GA proceeds by building up the population through the use of the cut and splice recombination operator.

Sample Applications

- Design of controllers (Herreros, 2000).
- Traditional benchmarks (Van Veldhuizen & Lamont, 2000).

The Multi-Objective Messy Genetic Algorithm-II (MOMGA-II)

Zydallis et al. (2001) proposed MOMGA-II. In this case, the authors extended the fast-messy GA, which consists of three phases: (1) Initialization Phase, (2) Building Block Filtering, and (3) Juxtapositional Phase. Its main difference with respect to the original messy GA is in the two first phases. The *Initialization Phase* utilizes probabilistic complete initialization which creates a controlled number of building block clones of a specified size. The Building *Block Filtering Phase* reduces the number of building blocks through a filtering process and stores the best building blocks found. The *Juxtapositional Phase* is the same as in the MOMGA.

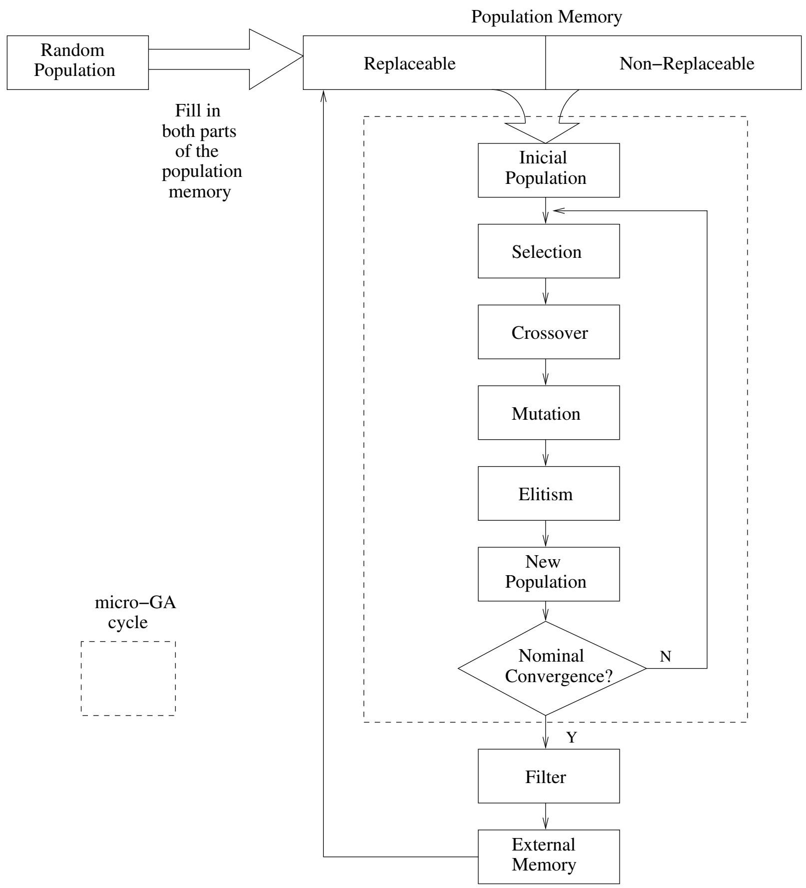
Sample Applications

- Traditional benchmarks (Zydallis et al., 2001).

The Micro Genetic Algorithm for Multiobjective Optimization

Proposed by Coello and Toscano-Pulido [2001]. A micro genetic algorithms uses a population size ≤ 5 individuals. The key aspect of the microGA is the use of a reinitialization process once nominal convergence is reached. The microGA for multiobjective optimization uses 3 forms of elitism and the adaptive grid from PAES.

The Micro Genetic Algorithm



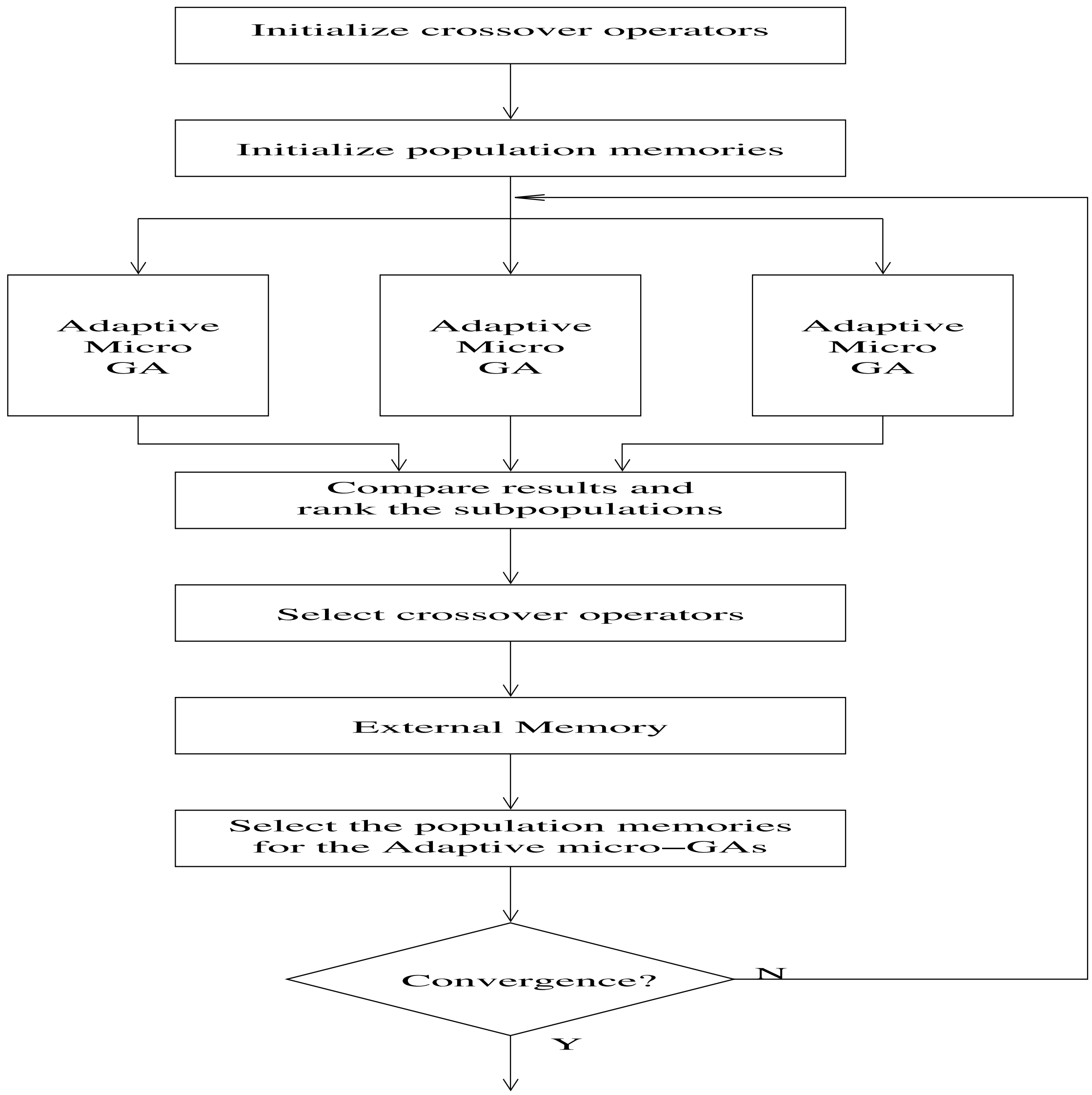
Sample Applications

- Supersonic jet design (Chung et al., 2003).
- Structural design (Coello, 2002).
- Hardware/Software Partitioning of UML Specifications (Fornaciari, 2003).

The Micro Genetic Algorithm2 (μGA^2)

Proposed by Toscano Pulido & Coello [2003]. The main motivation of the μGA^2 was to eliminate the 8 parameters required by the original algorithm. The $muGA^2$ uses on-line adaption mechanisms that make unnecessary the fine-tuning of any of its parameters. The $muGA^2$ can even decide when to stop (no maximum number of generations has to be provided by the user). The only parameter that it requires is the size of external archive (although there is obviously a default value for this parameter).

The Micro Genetic Algorithm2 (μGA^2)



Sample Applications

Since it is a very recent algorithm, so far it has been used only with traditional benchmarks.

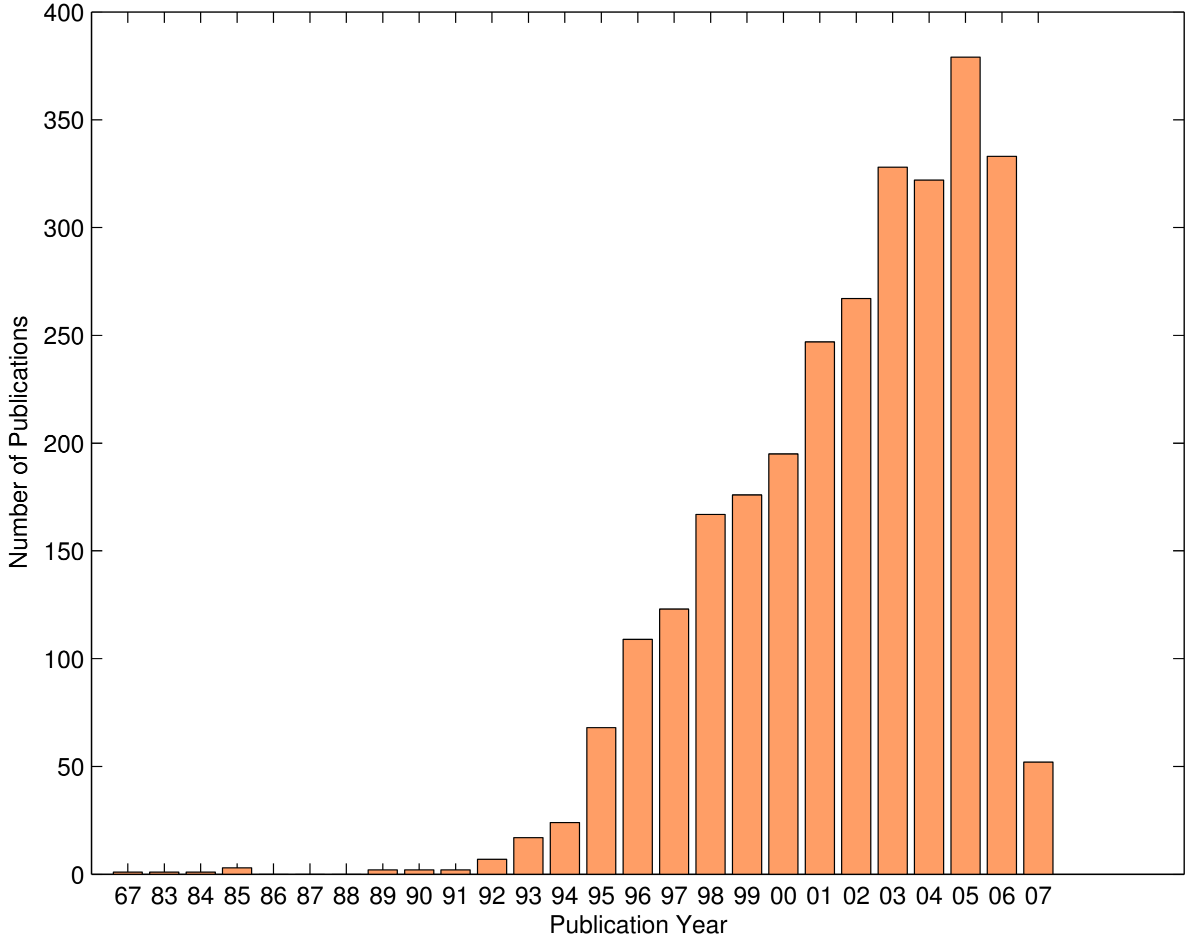
Current Trends in MOEAs

- After great success for over 10 years, first generation MOEAs have finally started to become obsolete in the literature (NSGA, NPGA, MOGA and VEGA).
- From the late 1990s, second generation MOEAs are considered the state-of-the-art in evolutionary multiobjective optimization (e.g., SPEA, SPEA2, NSGA-II, MOMGA, MOMGA-II, PAES, PESA, PESA II, microGA, etc.).

Current Trends in MOEAs

- Second generation MOEAs emphasize computational efficiency. One of the main goals is to find ways around the computational complexity of Pareto ranking ($O(kM^2)$, where k is the number of objective functions and M is the population size) and around the computational cost of niching ($O(M^2)$).
- Largely ignored by a significant number of researchers, non-Pareto MOEAs are still popular in Operations Research (e.g., in multiobjective combinatorial optimization), where they have been very successful.

Current state of the literature (mid of 2007)



Maintaining Diversity

An important aspect of MOEAs is to be able to maintain diversity in the population. For many years, *fitness sharing* was the main mechanism adopted for this purpose (Goldberg & Richardson, 1987). The idea of fitness sharing is to subdivide the population into several subpopulations based on the similarity among individuals. Note that when referring to MOEAs, “similarity” can be measured in the space of the decision variables (either encoded or decoded) or in the space of the objective functions.

Maintaining Diversity

Fitness sharing is defined in the following way:

$$\phi(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{share}}\right)^{\alpha}, & d_{ij} < \sigma_{share} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Maintaining Diversity

In the expression of the previous slide: $\alpha = 1$, d_{ij} indicates the distance between solutions i and j (in any space defined), and σ_{share} is a parameter (or threshold) that defines the size of a *niche* or neighborhood. Any solutions within this distance will be considered as part of the same niche. The fitness of an individual i is computed using:

$$f_{s_i} = \frac{f_i}{\sum_{j=1}^{M} \phi(d_{ij})} \tag{7}$$

where M is the number of individuals located in the neighborhood (or niche) of the i-th individual.

Maintaining Diversity

Several other schemes are possible to maintain diversity and to encourage a good spread of solutions. For example:

- Crowding techniques.
- Clustering techniques.
- The adaptive grid of PAES (Knowles & Corne, 2000).
- The use of the concept of entropy.

Maintaining Diversity

Other authors have also proposed the use of mating restrictions (which operate with a parameter σ_{mate} similar to σ_{share}). There are also several other mechanisms to maintain diversity (see for example Mahfoud's PhD thesis), although few of them have been adopted with MOEAs. Note however, that there are no standard guidelines regarding the suitability of any of these mechanisms for any specific application or algorithm and their use is based on the preferences of the user.

Theory

The most important theoretical work related to EMOO has concentrated on the following issues:

- Studies of convergence towards the Pareto optimum set (Rudolph, 1998, 2000, 2001; Hanne 2000,2000a; Van Veldhuizen, 1998).
- Ways to compute appropriate sharing factors (or niche sizes) (Horn, 1997, Fonseca, 1993).
- Run-time analysis (Laumanns et al., 2002, 2004).
- Extensions of the No Free Lunch Theorem to multiobjective optimization problems (Corne & Knowles, 2003).

Theory

Much more work is needed. For example:

- To study the structure of fitness landscapes (Kaufmann, 1989) in multiobjective optimization problems.
- Convergence of parallel MOEAs.
- Theoretical limit to the number of objective functions that can be used in practice.

Theory

- Formal models of alternative heuristics used for multiobjective optimization (e.g., ant system, particle swarm optimization, etc.).
- Complexity analysis of MOEAs and running time analysis (Laumanns et al., 2002).
- Study of population dynamics for an MOEA.

Test Functions

- Good benchmarks were disregarded for many years.
- Recently, there have been several proposals to design test functions suitable to evaluate EMOO approaches.
- Constrained test functions are of particular interest.
- Multiobjective combinatorial optimization problems have also been proposed.

Test Functions

- After using biobjective test functions for a few years, test functions with three objectives and higher numbers of decision variables are becoming popular.
- What about dynamic test functions (Farina et al., 2003), uncertainty and real-world applications?

Sample Test Functions

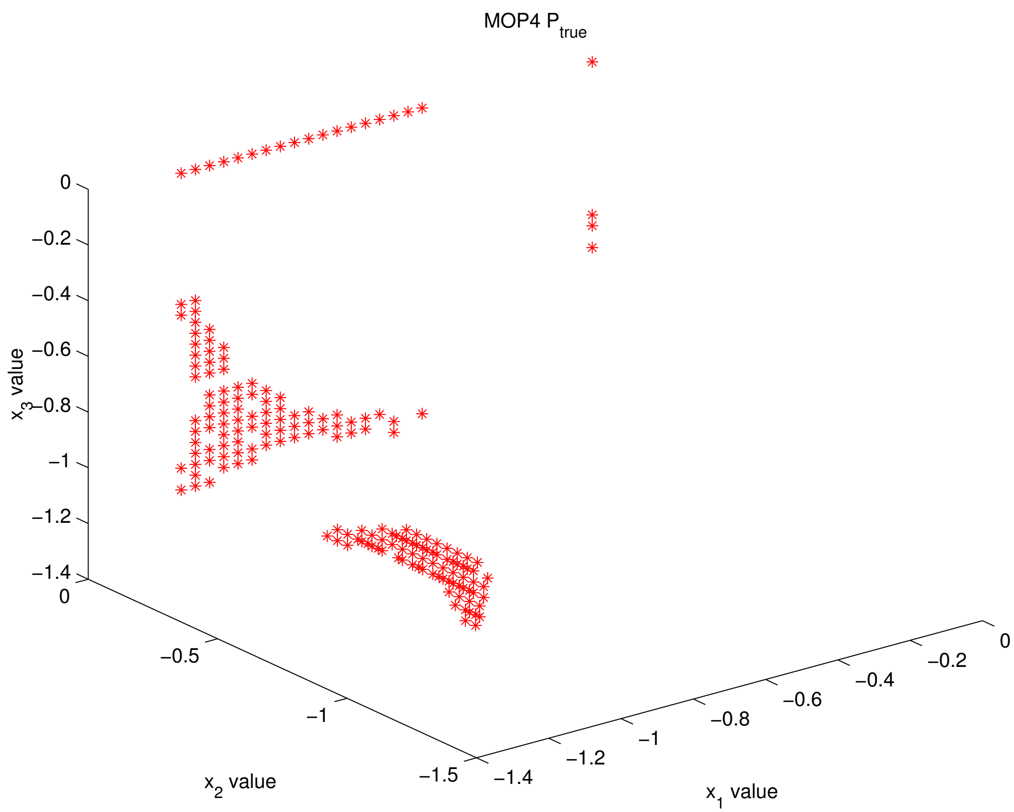


Figure 1: P_{true} of one of Deb's test functions.

Sample Test Functions

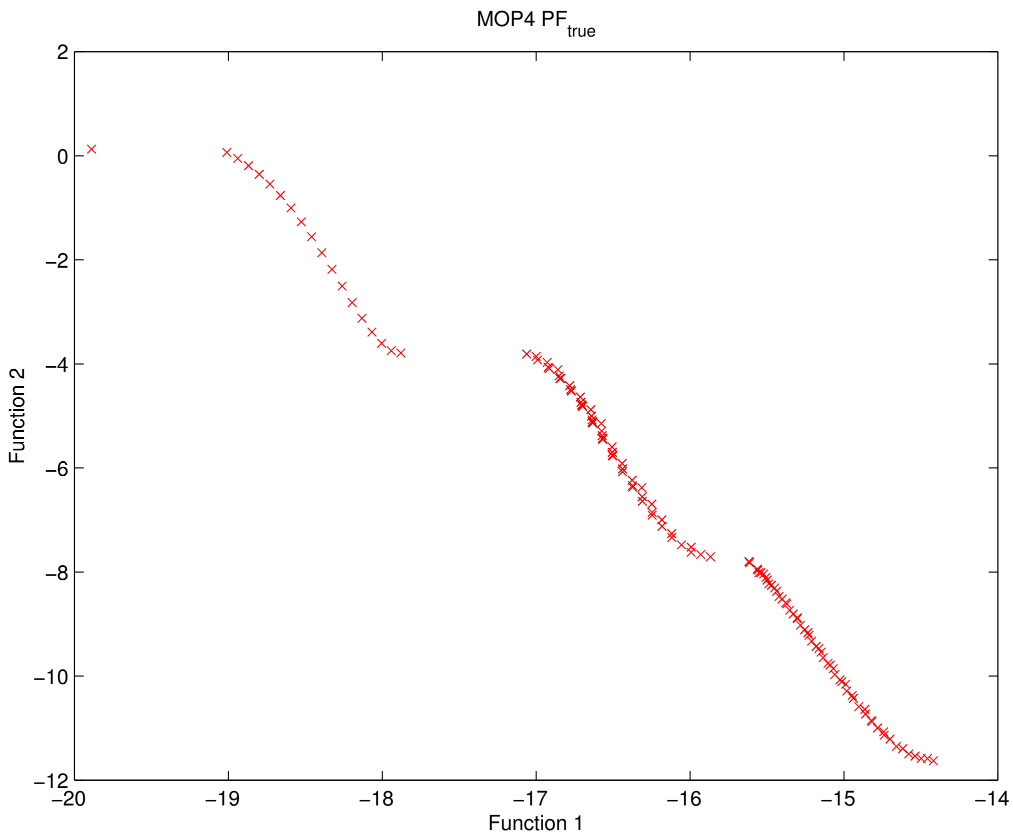


Figure 2: PF_{true} of one of Deb's test functions.

Sample Test Functions

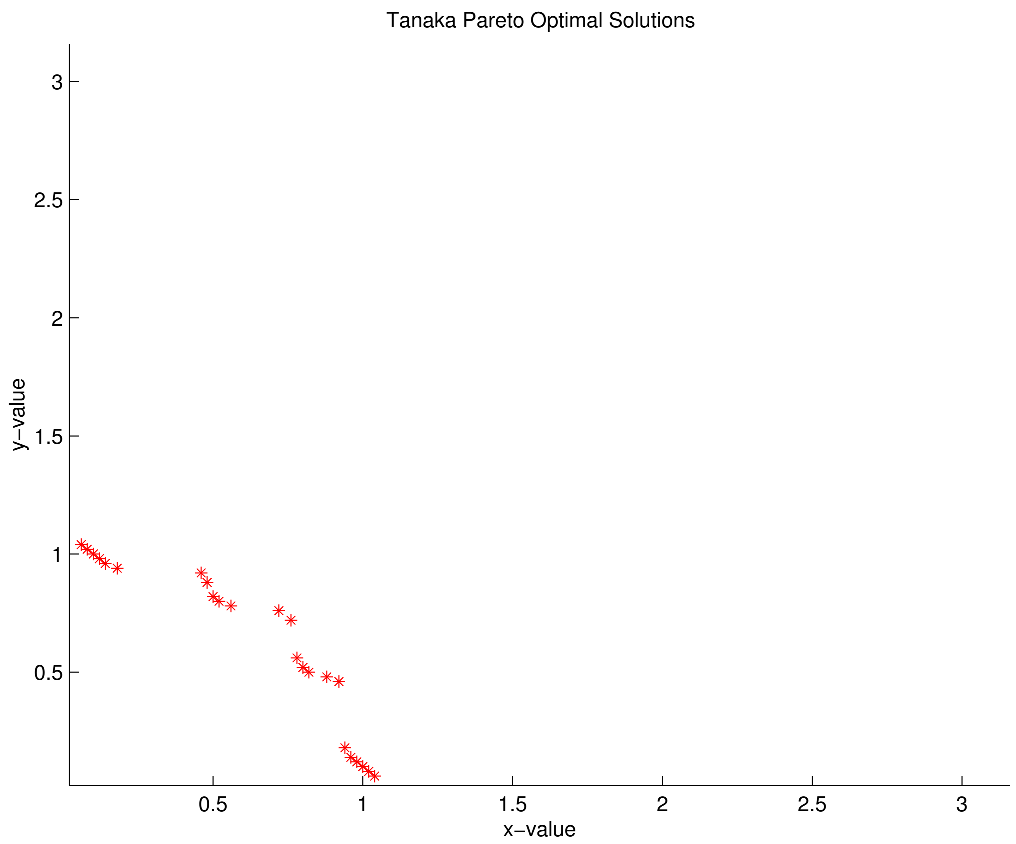


Figure 3: P_{true} of Tanaka's function.

Sample Test Functions

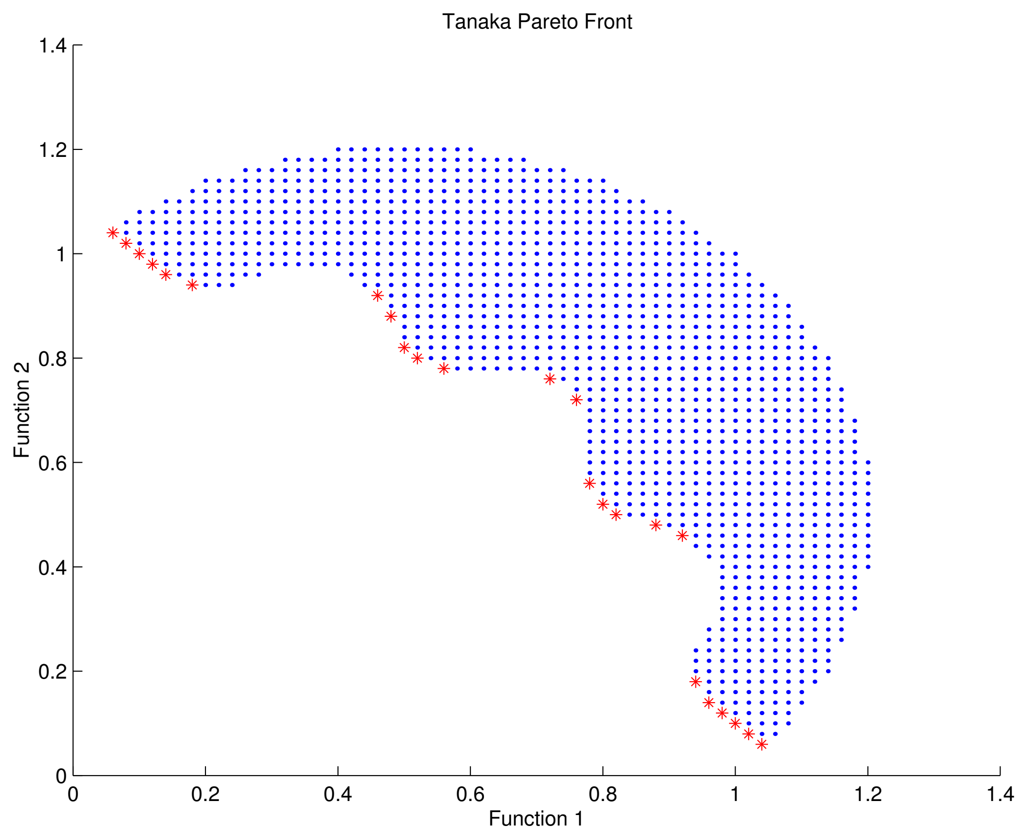


Figure 4: PF_{true} of Tanaka's function.

Metrics

Three are normally the issues to take into consideration to design a good metric in this domain (Zitzler, 2000):

1. Minimize the distance of the Pareto front produced by our algorithm with respect to the true Pareto front (assuming we know its location).
2. Maximize the spread of solutions found, so that we can have a distribution of vectors as smooth and uniform as possible.
3. Maximize the number of elements of the Pareto optimal set found.

Sample Metrics

Error ratio: Enumerate the entire intrinsic search space explored by an evolutionary algorithm and then compare the true Pareto front obtained against those fronts produced by any MOEA.

Obviously, this metric has some serious scalability problems.

Sample Metrics

Spread: Use of a statistical metric such as the chi-square distribution to measure "spread" along the Pareto front.

This metric assumes that we know the true Pareto front of the problem.

Sample Metrics

Attainment Surfaces: Draw a boundary in objective space that separates those points which are dominated from those which are not (this boundary is called "attainment surface").

Perform several runs and apply standard non-parametric statistical procedures to evaluate the quality of the nondominated vectors found.

It is unclear how can we really assess how much better is a certain approach with respect to others.

Sample Metrics

Generational Distance: Estimates how far is our current Pareto front from the true Pareto front of a problem using the Euclidean distance (measured in objective space) between each vector and the nearest member of the true Pareto front.

The problem with this metric is that only distance to the true Pareto front is considered and not uniform spread along the Pareto front.

Sample Metrics

Coverage: Measure the size of the objective value space area which is covered by a set of nondominated solutions.

It combines the three issues previously mentioned (distance, spread and amount of elements of the Pareto optimal set found) into a single value. Therefore, sets differing in more than one criterion cannot be distinguished.

A Word of Caution About the Use of Metrics

Recent research has shown the limitations of many of the metrics in current use (Zitzler et al., 2003). The most astonishing conclusion of this work is that many of the metrics in current use do not allow us to make strong staments about our results (i.e., “algorithm A is better than algorithm B”).

Promising areas of future research

- Alternative data structures (e.g., quadtrees) that allow efficient storage and retrieval of nondominated vectors (Mostaghim et al., 2002).
- More theoretical studies (convergence, mathematical models, fitness landscapes, run-time analysis, etc.).
- Study of parallel MOEAs (convergence, performance, comparisons, etc.).

Promising areas of future research

- New approaches (hybrids with other heuristics) and extensions of alternative heuristics (e.g., scatter search, cultural algorithms, reinforcement learning, etc.).
- New applications (e.g., in computer vision, robotics, physics, medicine, computer architecture, operating systems, etc.).
- What to expect for the third generation?

Promising areas of future research

- Tackling dynamic (multiobjective) test functions, handling uncertainty and high epistasis.
- Answering fundamental questions such as: what makes difficult a multiobjective optimization problem for an EA? Can we really produce reliable metrics for multiobjective optimization? Can we design robust MOEAs? Is there a way around the dimensionality curse in multiobjective optimization? Can we benefit from coevolutionary schemes?

To know more about evolutionary multiobjective optimization

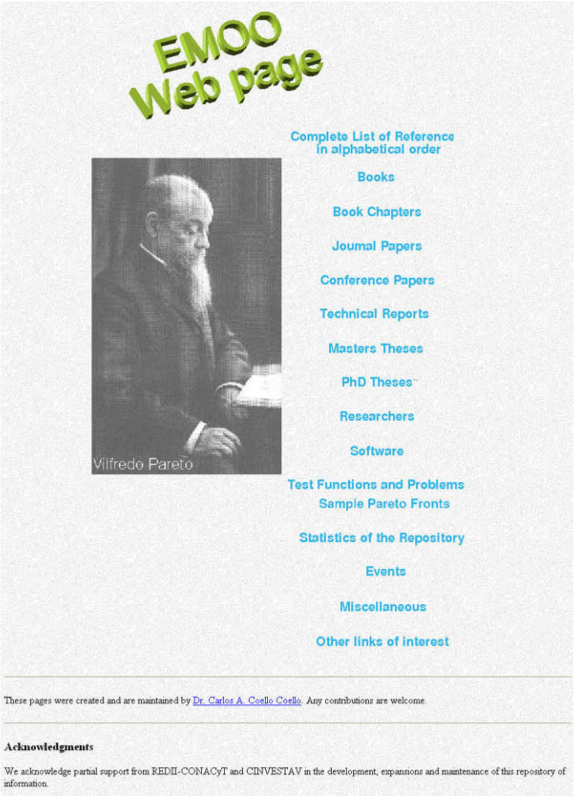
Please visit our EMOO repository located at:

http://delta.cs.cinvestav.mx/~ccoello/EMOO

with a mirror at:

http://www.lania.mx/~ccoello/EMOO

To know more about evolutionary multiobjective optimization

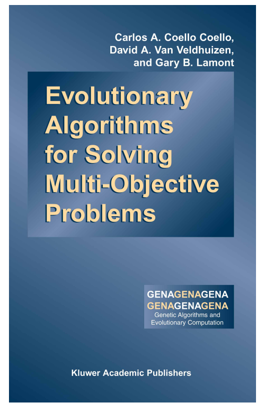


To know more about evolutionary multiobjective optimization

The EMOO repository currently contains:

- Over 2850 bibliographic references including over 170 PhD theses, 24 Masters theses, over 700 journal papers and over 1540 conference papers.
- Contact information of 66 EMOO researchers
- Public domain implementations of SPEA, NSGA, NSGA-II, the microGA, MOGA, ϵ-MOEA, MOPSO and PAES, among others.

To know more about evolutionary multiobjective optimization



You can consult the following book:

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.

(**The second edition will be published this year**)