Metaheuristics for Multiobjective Optimization

Carlos A. Coello Coello

CINVESTAV-IPN

Depto. de Computación

Av. Instituto Politécnico Nacional No. 2508

Col. San Pedro Zacatenco

México, D. F. 07300, MEXICO

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.

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:

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

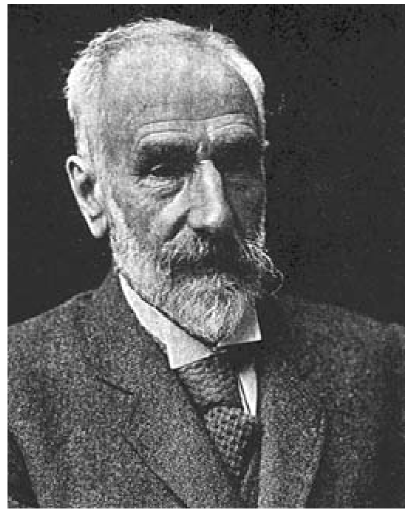
the p equality constraints

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

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*.

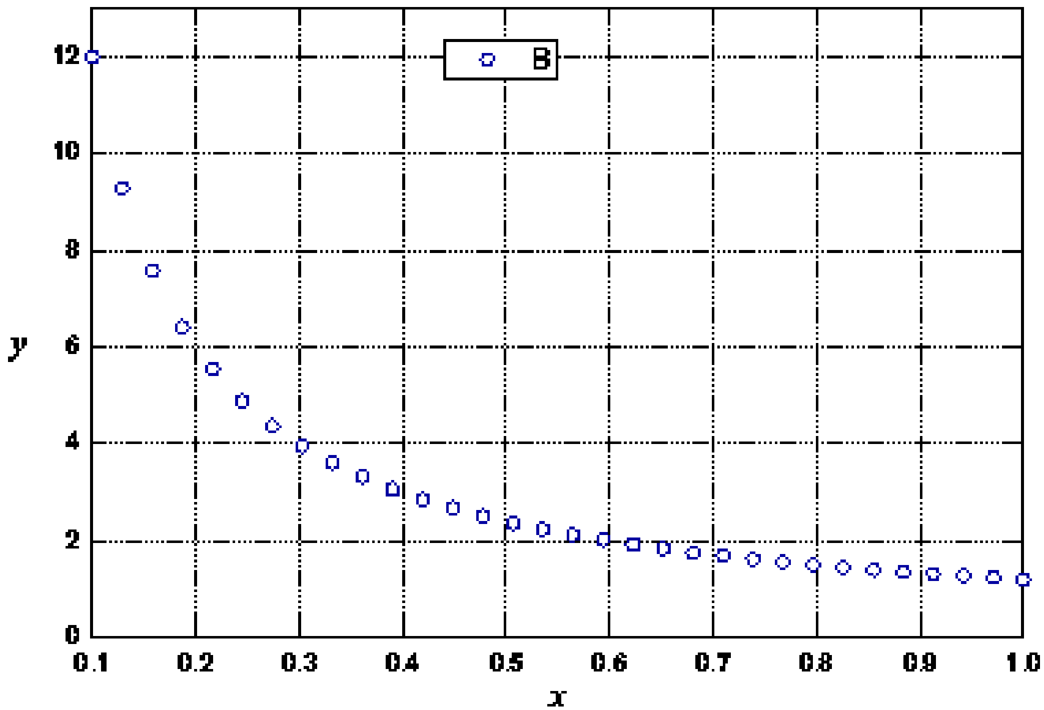
Definition of Pareto Optimality

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.

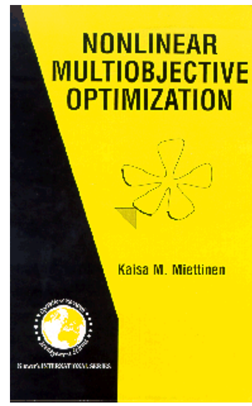
Definition of Pareto Optimality

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



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 Metaheuristics?

Metaheuristics seem particularly suitable to solve multiobjective optimization problems, because they 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 this is a real concern for mathematical programming techniques. Additionally, many current metaheuristics (e.g., evolutionary algorithms, particle swarm optimization, etc.) are population-based, which means that we can aim to generate several elements of the Pareto optimal set in a single run.

Evolutionary Algorithms

EAs use a selection mechanism based on fitness. We can consider, in general, two main types of multi-objective evolutionary algorithms (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)

- VEGA
- MOGA
- NSGA
- NPGA

Vector Evaluated Genetic Algorithm (VEGA)

- Proposed by David 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.

Vector Evaluated Genetic Algorithm (VEGA)

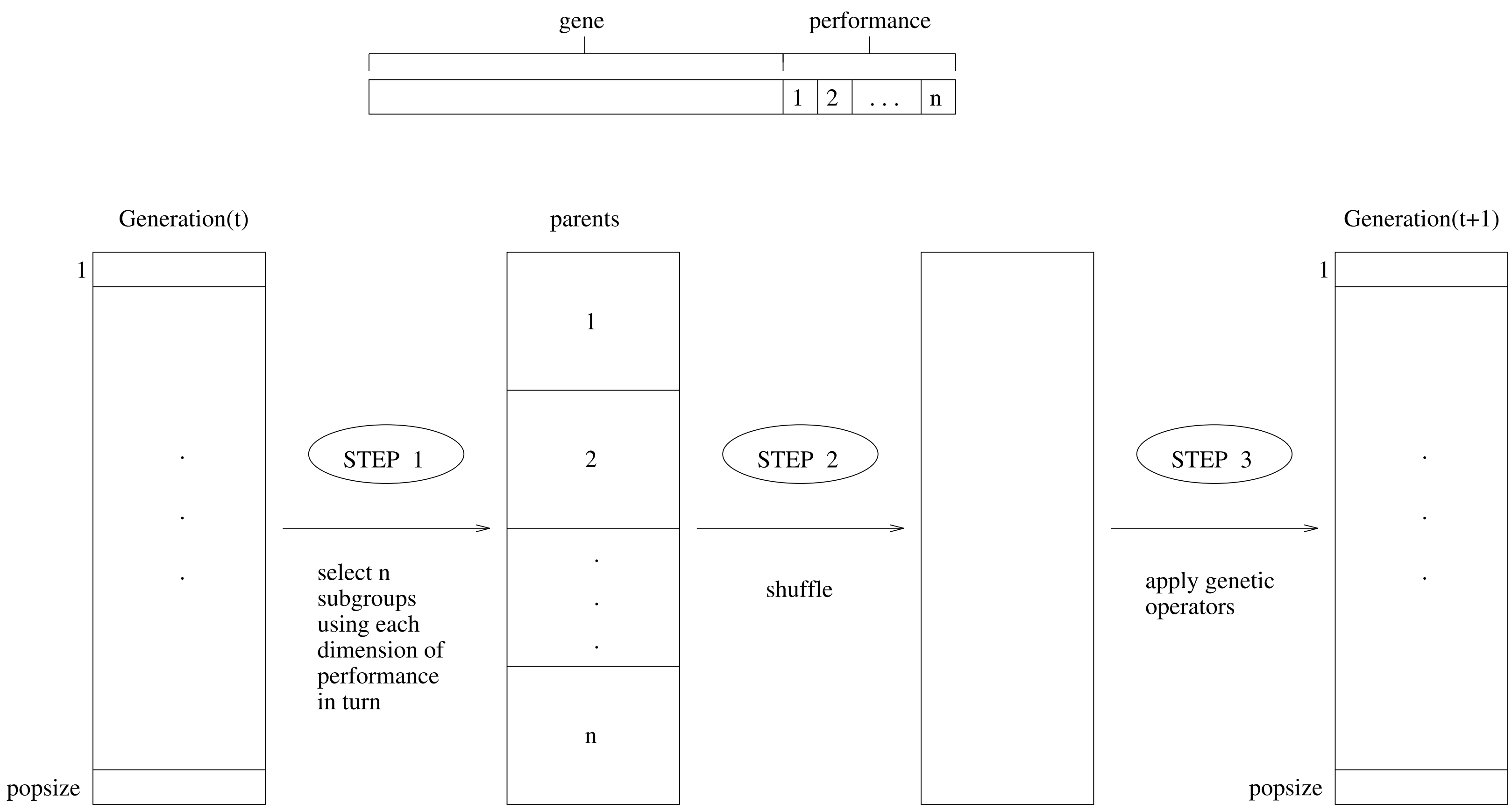


Figure 1: Schematic of VEGA selection.

Multi-Objective Genetic Algorithm (MOGA)

- Proposed by Carlos Fonseca and Peter 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.

Nondominated Sorting Genetic Algorithm (NSGA)

- Proposed by N. Srinivas and Kalyanmoy Deb (1994).
- It is based on several layers of classifications of the individuals. Nondominated individuals get a certain dummy fitness value and then are removed from the population. The process is repeated until the entire population has been classified.
- To maintain the diversity of the population, classified individuals are shared (in decision variable space) with their dummy fitness values.

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).

Representative MOEAs (Second Generation)

- SPEA and SPEA2
- NSGA-II
- PAES, PESA and PESA II
- The microGA for multiobjective optimization and the μGA^2

The Strength Pareto Evolutionary Algorithm (SPEA)

SPEA was introduced by Eckart Zitzler & Lothar Thiele (1999). It uses an external archive containing nondominated solutions previously found.

It computes a strength value similar to the ranking value used by MOGA. A clustering technique called "average linkage method" is used to keep diversity.

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.

The Nondominated Sorting Genetic Algorithm II (NSGA-II)

Kalyanmoy Deb and co-workers (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.

The Pareto Archived Evolution Strategy (PAES)

PAES was introduced by Joshua Knowles & David Corne (2000).

It uses a (1+1) evolution strategy together with an external archive that records all the nondominated vectors previously found.

It uses an adaptive grid to maintain diversity.

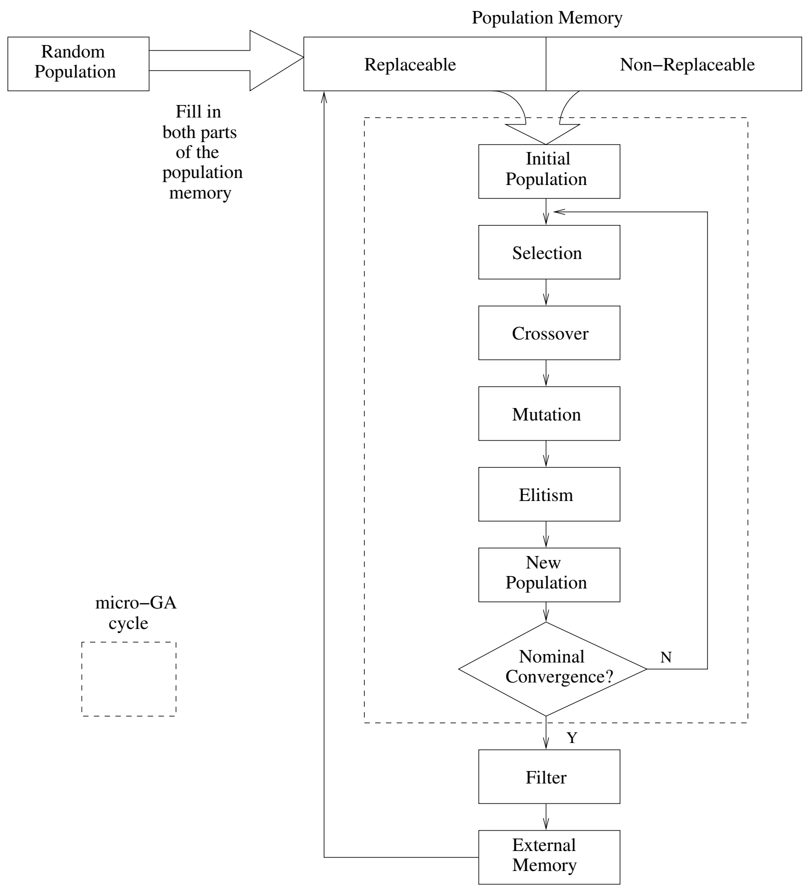
The Pareto Envelope-based Selection Algorithm (PESA)

PESA was proposed by David Corne and co-workers (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).

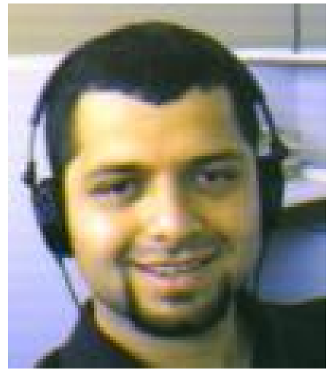
The Pareto Envelope-based Selection Algorithm-II (PESA-II)

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.

The Micro Genetic Algorithm for Multiobjective Optimization

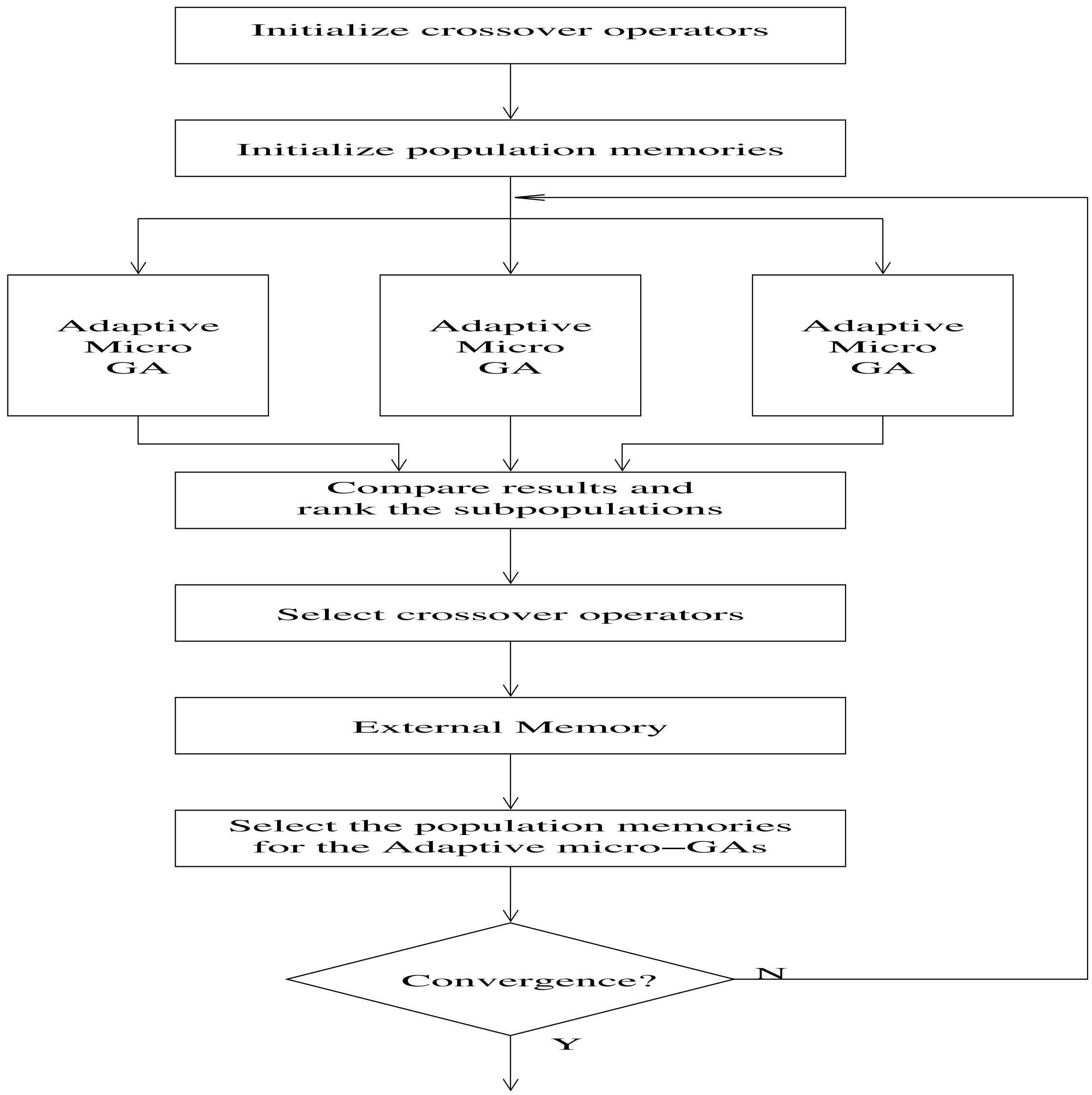


The Micro Genetic Algorithm2 (μGA^2)



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 μGA^2 uses on-line adaption mechanisms that make unnecessary the fine-tuning of any of its parameters. The μGA^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)



MOEAs that the world forgot

- The **Incrementing Multi-Objective Evolutionary Algorithm** (IMOEA) (Tan et al., 2001). It uses a dynamic population size, based on the current approximation of the Pareto front. It also adopts an adaptive niching method.
- The **Constraint Method-Based Evolutionary Algorithm** (CMEA) for Multiobjective Optimization (Ranjithan et al., 2001): Based on the ϵ-constraint method.
- The **Orthogonal Multi-Objective Evolutionary Algorithm** (OMOEA) (Zeng et al., 2004). Based on orthogonal design and other statistical techniques. It adopts niching.

MOEAs that the world forgot

- The **MaxiMin** Method (Balling, 2000). It uses some sort of compromise programming to assign fitness and estimate density with a single mathematical expression.
- The **Method based on Distances and the Contact Theorem** from Osyczka and Kundu (1996). It uses a nonlinear aggregation function that estimates distance with respect to the ideal vector.
- The **Non-Generational Genetic Algorithm for Multiobjective Optimization** from Valenzuela-Rendón and Uresti-Charre (1997). It uses an aggregation function that combines dominance count with niching (it uses fitness sharing).

MOEAs that the world forgot

- The **Thermodynamic Genetic Algorithm** (Kita, 1996). This is the multi-objective version of an algorithm originally proposed for combinatorial optimization. This algorithm incorporates the concept of entropy and the use of a cooling schedule in its selection mechanism.
- The **Nash Genetic Algorithm** (Sefrioui, 1996). It uses a co-evolutionary scheme to try to approximate a Nash equilibrium point (in a Nash strategy, each player tries to optimize his/her own criterion, assuming that the other players' criteria are fixed). It uses a distance-based mutation operator. It requires certain mathematical calculations to define the model to be optimized with a genetic algorithm and its outcome is a single solution.

MOEAs that the world forgot

- **Use of genders:** Allenson (1992) used a variation of VEGA in which genders were used to distinguish between the two objective functions related to the planning of a path composed by several rectilinear pipe segments. In this approach, recombination is only possible between pairs of individuals having a different gender (a male and a female) and the gender is randomly assigned to an offspring. In the initial population, it is ensured that half of the population are male and half are female, but this balance is no longer maintained upon the application of the genetic operators. At each generation, the worst individual (chosen from one of the two genders) is eliminated and its place is taken by another individual (randomly selected) from its same gender.

MOEAs that the world forgot

Allenson used evolution strategies to implement the sexual attractors that modify the way in which recombination takes place. The idea is to model the sexual attraction that occurs in nature and which determines a not so random mating. In 1996, Lis and Eiben proposed a generalization of this approach in which there are as many genders as objectives.

MOEAs that the world forgot

- **ε-MOEA:** Deb (2003, 2005) proposed a MOEA based on a relaxed form of Pareto dominance called ε-dominance (Laumanns et al., 2002). This approach uses steady state selection and adopts an external population that incorporates ε-dominance.
- **C-NSGA-II:** The *Clustered NSGA-II* is a version of NSGA-II in which *crowding* is replaced by the *clustering* algorithm adopted in SPEA.

Modern MOEAs (After 2002)

- MOEA/D
- Indicator-based Selection (hypervolume and others)
- NSGA-III

MOEA/D

The *Multi-Objective Evolutionary Algorithm based on Decomposition* (MOEA/D) proposed by Zhang et al. (2007) is one of the most competitive MOEAs in current used. This approach decomposes a multi-objective problem into several single-objective optimization problems, which are simultaneously solved. Each subproblem is optimized using information from its neighboring subproblems, in contrast with similar approaches (e.g., MOGLS (Ishibuchi & Murata, 1996)). This MOEA is inspired on a mathematical programming technique called *Normal Boundary Intersection* (NBI) (Das, 1998).

Indicator-based Selection

Perhaps the most important trend on the design of moderns MOEAs is the use of a performance measure in their selection mechanism. For example:

- **ESP**: The *Evolution Strategy with Probability Mutation* uses a measure based on the hypervolume, which is scale independent and doesn't require any parameters, in order to truncate the contents of the external archive (Huband et al., 2003).
- **IBEA**: The *Indicator-Based Evolutionary Algorithm* is an algorithmic framework that allows the incorporation of any performance indicator in the selection mechanism of a MOEA (Zitzler et al., 2004). It was originally tested using the hypervolume and the binary ϵ indicator.

Indicator-based Selection

Another important MOEA within this class is the following:

- **SMS-EMOA**: Emmerich et al. (2005) proposed an approach based on NSGA-II and the archiving techniques proposed by Knowles, Corne and Fleischer. This approach was called *S Metric Selection Evolutionary Multiobjective Algorithm*. SMS-EMOA creates an initial population and generates a single solution per iteration (i.e., it uses steady state selection) using the crossover and mutation operators from NSGA-II. Then, it applies Pareto ranking. When the last nondominated front has more than one solution, SMS-EMOA uses hypervolume to decide which solution should be removed.

Indicator-based Selection

In 2007, Beume et al. proposed a new version of SMS-EMOA in which the hypervolume contribution is not used when, in the Pareto ranking process, we obtain more than one front. In this case, they use the number of solutions that dominate to a certain individual (i.e., the solution that is dominated by the largest number of solutions is removed). The authors of this approach indicate that their motivation to use the hypervolume is to improve the distribution of solutions along the Pareto front (in other words, hypervolume is used only as a density estimator).

Indicator-based Selection

- **MO-CMA-ES**: This is a multi-objective version of the *covariance matrix adaptation evolution strategy* (CMA-ES) proposed by Igel et al. (2007). Its selection mechanism is based on a nondominated sorting that adopts as its second selection criterion either the crowding distance or the hypervolume contribution (two versions of the algorithm were tested, and the one based on the hypervolume has the best overall performance). This MOEA is rotation invariant, as its original single-objective optimizer.
- **SPAM**: The *Set Preference Algorithm for Multiobjective optimization* is a generalization of IBEA which allows to adopt any set preference relation in its selection mechanism (Zitzler et al., 2008).

Indicator-based Selection

- **HyPE**: The *hypervolume estimation algorithm for multi-objective optimization*, was proposed by Bader (2011). In this case, the author proposes a quick search algorithm that uses Monte Carlo simulations to approximate the hypervolume contributions. The core idea is that the actual hypervolume contribution value is not that important, but only the actual ranking that is produced with it. Although this proposal is quite interesting, in practice its performance is rather poor with respect that of MOEAs that use the exact hypervolume contributions.

Indicator-based Selection

The **hypervolume** (also known as the S metric or the Lebesgue measure) of a set of solutions, measures the size of the portion of objective space that is dominated by such solutions, collectively.

The hypervolume is the only performance indicator that is known to be monotonic with respect to Pareto dominance. This guarantees that the true Pareto front achieves the maximum possible hypervolume value, and any other set will produce a lower value for this indicator.

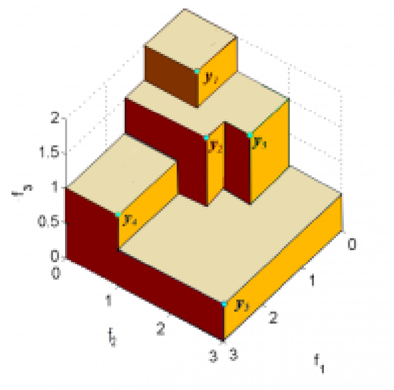
Indicator-based Selection

Fleischer (2003) proved that, given a finite search space and a reference point, maximizing the hypervolume is equivalent to obtaining the Pareto optimal set. Therefore, a bounded set that contains the maximum possible hypervolume value for a certain population size, will only consist of Pareto optimal solutions. This has been experimentally validated (Knowles, 2003; Emmerich, 2005), and it has been observed that such solutions also have a good distribution along the Pareto front.

Indicator-based Selection

The computation of the hypervolume depends on the reference point that we adopt, and this point can have a significant influence on the results. Some researchers have proposed to use the worst objective function values available in the current population, but this requires a scaling of the objectives.

Indicator-based Selection



However, the main drawback of using the hypervolume is its high computational cost. The best known algorithms currently available to compute the hypervolume have a complexity that is polynominal on the number of points, but such a complexity grows exponentially with the number of objectives.

Indicator-based Selection

It is worth noting that the use of the hypervolume to select solutions is not straightforward. This indicator operates on a set of solutions, and the selection operator considers only one solution at a time. Therefore, when using the hypervolume to select solutions, a fitness assignment strategy is required.

Indicator-based Selection

The strategy that has been most commonly adopted in the specialized literature consists of performing first a nondominated sorting procedure and then ranking the solutions within each front based on the hypervolume loss that results from removing a particular solution (Knowles and Corne, 2003; Emmerich et al., 2005; Igel et al., 2007; Bader et al., 2010).

Indicator-based Selection

The main motivation for using indicators in the selection mechanism is scalability (in objective function space). However, the high computational cost of the hypervolume has motivated the exploration of alternative performance indicators, such as Δ_p. See:

Oliver Schütze, Xavier Esquivel, Adriana Lara and Carlos A. Coello Coello, **Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multi-Objective Optimization**, *IEEE Transactions on Evolutionary Computation*, Vol. 16, No. 4, pp. 504–522, August 2012.

Indicator-based Selection

The Δ_p indicator can be seen as an "averaged Hausdorff distance" between our approximation and the true Pareto front. Δ_p combines some slight variations of two well-known performance indicators: generational distance (Van Veldhuizen, 1999) and inverted generational distance (Coello & Cruz, 2005).

Indicator-based Selection

Δ_p is a pseudo-metric that simultaneously evaluates proximity to the true Pareto front and the distribution of solutions along it. Although it is not a Pareto compliant indicator, in practice, it seems to work reasonably well, being able to deal with outliers. This makes it attractive as a performance indicator. Additionally, its computational cost is very low.

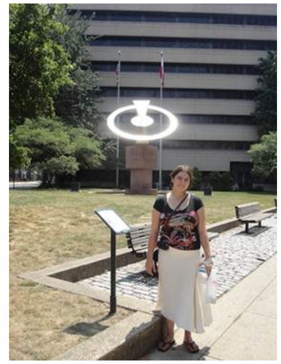
Indicator-based Selection

Nevertheless, it is worth mentioning that in order to incorporate Δ_p in the selection mechanism of a MOEA, it is necessary to have an approximation of the true Pareto front at all times. This has motivated the development of techniques that can produce such an approximation in an efficient and effective manner. For example, Gerst et al. (2011) lineralized the nondominated front produced by the current population and used that information in the so-called Δ_p-EMOA, which was used to solve bi-objective problems. This algorithm is inspired on the SMS-EMOA and adopts an external archive.

Indicator-based Selection

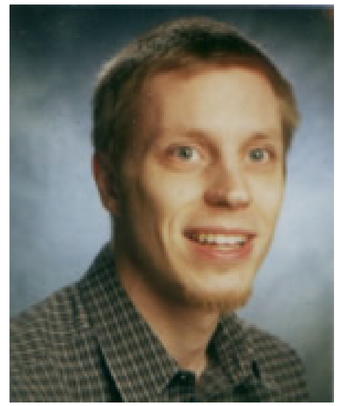
There was a further extension of this MOEA for dealing with problems having three objectives (Trautmann, 2012). In this case, the algorithm requires some prior steps, which include reducing the dimensionality of the nondominated solutions and computing their convex hull. This version of the Δ_p-EMOA generates solutions with a better distribution, but requires more parameters and has a high computational cost when is used for solving many-objective optimization problems.

Indicator-based Selection



Another possible way of incorporating Δ_p into a MOEA is to use the currently available nondominated solutions in a stepped way, in order to build an approximation of the true Pareto front. This was the approach adopted by the Δ_p-DDE (Rodríguez & Coello, 2012), which uses differential evolution as its search engine. This MOEA provides results of similar quality to those generated by SMS-EMOA, but at a much lower computational cost (in high dimensionality). Its main limitation is that its solutions are normally not well-distributed in many-objective problems. Additionally, it has difficulties to deal with disconnected Pareto fronts.

Indicator-based Selection



Recently, some researchers have recommended the use of the $R2$ indicator, which was originally proposed by Hansen (1998) for comparing sets of solutions using utility functions (Brockhoff, 2012). A utility function is a model of the decision maker preferences that maps each point from the objective function space to a utility value.

Indicato-based Selection

It is worth indicating that $R2$ is weakly monotonic and that it's correlated to the hypervolume, but has a much lower computational cost. Due to these properties, its use is recommended for dealing with many-objective problems. Nevertheless, the utility functions that are required to compute this indicator have to be properly scaled.

Indicator-based Selection

Currently, there are already several MOEAs based on $R2$:

Raquel Hernández Gómez and Carlos A. Coello Coello, **MOMBI: A New Metaheuristic for Many-Objective Optimization Based on the R2 Indicator**, in *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pp. 2488–2495, IEEE Press, Cancún, México, 20-23 June, 2013, ISBN 978-1-4799-0454-9.

Dimo Brockhoff, Tobias Wagner and Heike Trautmann, $R2$ **Indicator-Based Multiobjective Search**, *Evolutionary Computation*, Vol. 23, No. 3, pp. 369–395, Fall 2015.

Indicator-based Selection

Alan Díaz-Manríquez, Gregorio Toscano-Pulido, Carlos A. Coello Coello and Ricardo Landa-Becerra, **A Ranking Method Based on the R2 Indicator for Many-Objective Optimization**, in *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pp. 1523–1530, IEEE Press, Cancún, México, 20-23 June, 2013, ISBN 978-1-4799-0454-9.

Dúng H. Phan and Junichi Suzuki, **R2-IBEA: R2 Indicator Based Evolutionary Algorithm for Multiobjective Optimization**, in *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pp. 1836–1845, IEEE Press, Cancún, México, 20-23 June, 2013, ISBN 978-1-4799-0454-9.

NSGA-III

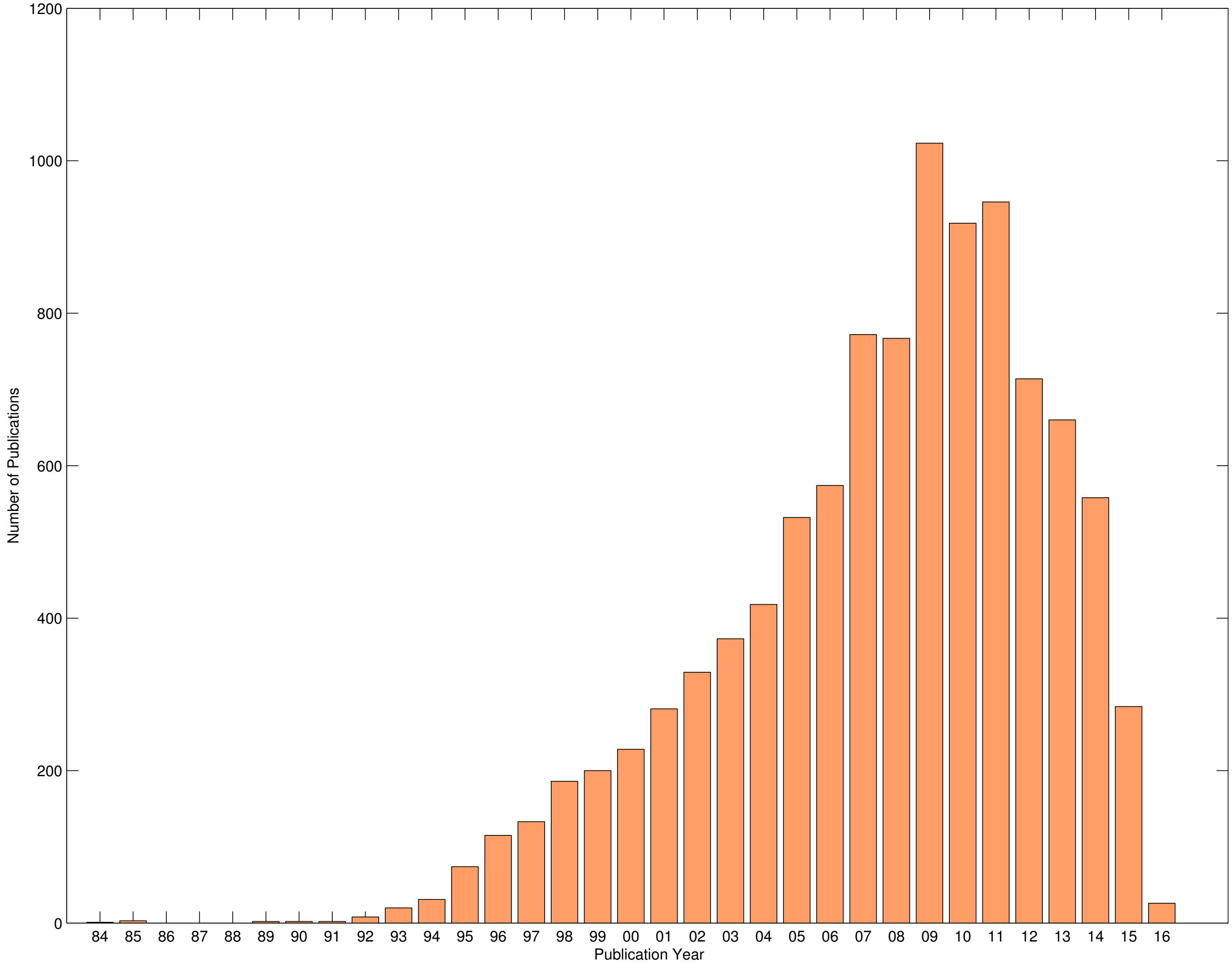
The *Nondominated Sorting Genetic Algorithm III* (NSGA-III) was proposed by Deb and Jain (2014) as an extension of NSGA-II specifically designed to deal with many-objective problems (i.e., multi-objective optimization problems having 4 or more objectives). NSGA-III still uses nondominated sorting (producing different levels), but in this case, the density estimation is done through adaptively updating a number of well-spread reference points.

To Know More About Multi-Objective Evolutionary Algorithms

Kalyanmoy Deb, **Multi-Objective Optimization using Evolutionary Algorithms**, John Wiley & Sons, Chichester, UK, 2001, ISBN 0-471-87339-X.

Carlos A. Coello Coello, Gary B. Lamont and David A. Van Veldhuizen, **Evolutionary Algorithms for Solving Multi-Objective Problems**, Second Edition, Springer, New York, ISBN 978-0-387-33254-3, September 2007.

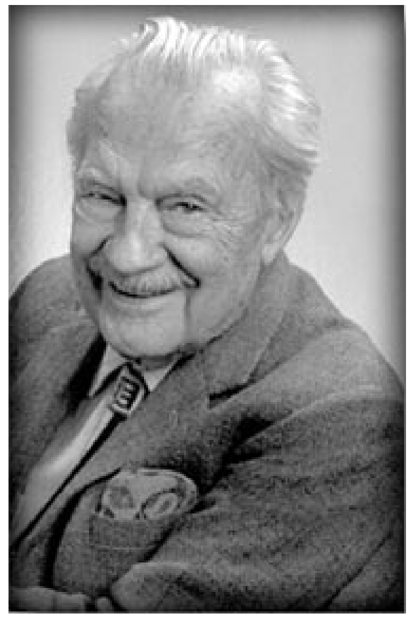
Current state of the literature (mid 2016)



Alternative Heuristics

- Simulated Annealing
- Tabu Search
- Ant System
- Particle Swarm Optimization
- Artificial Immune Systems
- Differential Evolution

Simulated Annealing



Based on an algorithm originally proposed by Metropolis et al. (1953) to simulate the evolution of a solid in a heat bath until it reaches its thermal equilibrium.

Simulated Annealing

Kirkpatrick et al. (1983) and Černy (1985) independently pointed out the analogy between the “annealing” process proposed by Metropolis and combinatorial optimization and proposed the so-called “simulated annealing algorithm”.

Simulated Annealing

1. Select an initial (feasible) solution s_0
2. Select an initial temperature $t_0 > 0$
3. Select a cooling schedule CS
4. Repeat

   Repeat

   Randomly select $s \in N(s_0)$ (N = neighborhood structure)

   $\delta = f(s) - f(s_0)$ (f = objective function)

   If $\delta < 0$ then $s_0 \leftarrow s$

   Else

   Generate random x (uniform distribution in the range $(0, 1)$)

   If $x < \exp(-\delta/t)$ then $s_0 \leftarrow s$

   Until max. number of iterations $ITER$ reached

   $t \leftarrow CS(t)$
5. Until stopping condition is met

Simulated Annealing

SA generates local movements in the neighborhood of the current state, and accepts a new state based on a function depending on the current "temperature" t. The two main parameters of the algorithm are $ITER$ (the number of iterations to apply the algorithm) and CS (the cooling schedule), since they have the most serious impact on the algorithm's performance.

Despite the fact that it was originally intended for combinatorial optimization, other variations of simulated annealing have been proposed to deal with continuous search spaces.

Simulated Annealing

The key in extending simulated annealing to handle multiple objectives lies in determining how to compute the probability of accepting an individual $\vec{x}'$ where $f(\vec{x}')$ is dominated with respect to $f(\vec{x})$.

Simulated Annealing

Some multiobjective versions of SA are the following:

- Serafini (1994): Uses a target-vector approach to solve a bi-objective optimization problem (several possible transition rules are proposed).
- Ulungu (1993): Uses an L_∞-Tchebycheff norm and a weighted sum for the acceptance probability.
- Czyzak & Jaszkiewicz (1997,1998): Population-based approach that also uses a weighted sum.
- Ruiz-Torres et al. (1997): Uses Pareto dominance as the selection criterion.

Simulated Annealing

- Suppapitnarm et al. (1999,2000): Uses Pareto dominance plus a secondary population.
- Baykasoğlu (2005): Uses preemptive goal programming (the most important goals are optimized first, followed by the secondary goals).
- Suman (2002,2003): Uses Pareto dominance, an external archive and a scheme that handles constraints within the expression used to determine the probability of moving to a certain state.
- Bandyopadhyay et al. (2008): It selects individuals with a probability that depends on the amount of domination measures in terms of the hypervolume measure. It uses an external archive

Some Applications of Multiobjective Simulated Annealing

- Design of a cellular manufacturing system (Czyzak, 1997).
- Nurse scheduling (Jaszkiewicz, 1997).
- Portfolio optimization (Chang, 1998).
- Aircrew rostering (Lučić & Teodorović, 1999).
- Ship design (Ray, 1995).
- Optimization of bicycle frames (Suppapitnarm, 1999).
- Parallel machine scheduling (Ruiz-Torres, 1997)
- Analog Filter Tuning (Thompson, 2001)

To Know More About Multiobjective Simulated Annealing

B. Suman and P. Kumar, **A survey of simulated annealing as a tool for single and multiobjective optimization**, *Journal of the Operational Research Society*, Vol. 57, No. 10, pp. 1143–1160, October 2006.

Carlos A. Coello Coello, Gary B. Lamont and David A. Van Veldhuizen, **Evolutionary Algorithms for Solving Multi-Objective Problems**, Second Edition, Springer, New York, ISBN 978-0-387-33254-3, September 2007.

Tabu Search

Tabu search was proposed by Fred Glover in the mid-1980s. In general terms, tabu search has the three following components (Glover & Laguna, 1997):

- A short-term memory to avoid cycling.
- An intermediate-term memory to intensify the search.
- A long-term memory to diversify the search.

Tabu Search

1. Select $x \in \mathcal{F}$ ($\mathcal{F}$ represents feasible solutions)
2. $x^* = x$ (x^* is the best solution found so far)
3. $c = 0$ (iteration counter)
4. $T = \emptyset$ (T set of "tabu" movements)
5. If $\mathcal{N}(x) - T = \emptyset$, goto step 4 ($\mathcal{N}(x)$ is the neighborhood function)
6. Otherwise, $c \leftarrow c + 1$
   Select $n_c \in \mathcal{N}(x) - T$ such that: $n_c(x) = \mathit{opt}(n(x) : n \in \mathcal{N}(x) - T)$
   $\mathit{opt}()$ is an evaluation function defined by the user
7. $x \leftarrow n_c(x)$
   If $f(x) < f(x^*)$ then $x^* \leftarrow x$
8. Check stopping conditions:
   Maximum number of iterations has been reached
   $\mathcal{N}(x) - T = \emptyset$ after reaching this
   step directly from step 2.
9. If stopping conditions are not met, update T
   and return to step 2

Tabu Search

The basic idea of tabu search is to create a subset T of $\mathcal{N}$, whose elements are called "tabu moves" (historical information of the search process is used to create T). Membership in T is conferred either by a historical list of moves previously detected as improductive, or by a set of tabu conditions (e.g., constraints that need to be satisfied). Therefore, the subset T constrains the search and keeps tabu search from becoming a simple hillclimber. At each step of the algorithm, a "best" movement (defined in terms of the evaluation function $opt()$) is chosen. Note that this approach is more aggressive than the gradual descent of simulated annealing.

Tabu Search

Tabu search tends to generate moves that are in the area surrounding a candidate solution. Therefore, the main problem when extending this technique to deal with multiple objectives is how to maintain diversity so that the entire Pareto front can be generated. The proper use of the historial information stored is another issue that deserves attention.

Tabu Search

Some multiobjective versions of tabu search are the following:

- Hansen (1997): MOTS*, which uses a λ-weighted Tchebycheff metric.
- Gandibleux et al. (1997): MOTS, which is based on the use of an utopian reference point.
- Hertz et al. (1994): Proposed 3 approaches (weighted sum of objectives, lexicographic ordering and the ε-constraint method).
- Baykasoğlu (1999,2001): MOTS, which uses 2 lists: the *Pareto list* (stores the nondominated solutions found during the search), and the *candidate list* (stores all the solutions which are not globally nondominated, but were locally nondominated at some stage of the search). Elements from the candidate list are used to diversify the search.

Tabu Search

- Ho et al. (2002): Uses Pareto ranking (as in MOGA), an external archive (which is bounded in size), fitness sharing and a neighborhood generation based on the construction of concentric "hypercrowns".
- Jaeggi et al. (2004): Proposes a multi-objective parallel tabu search approach that operates on continuous search spaces. The search engine is based on a multi-objective version of the Hooke and Jeeves method coupled with short, medium and long term memories.

Tabu Search

- Kulturel-Konak (2006): Proposes the multinomial tabu search (MTS) algorithm for multi-objective combinatorial optimization problems. The idea is to use a multinomial probability mass function to select an objective (considered "active") at each iteration. The approach uses an external archive in which solutions are added based on Pareto dominance. The approach also performs neighborhood moves, and uses a diversification scheme based on restarts.
- Xu et al. (2006): Uses an aggregating function. However, a set of rules based on Pareto dominance are used when evaluating neighborhood moves, so that some moves during the search may be based on Pareto dominance.

Some Applications of Multiobjective Tabu Search

- Resource constrained project scheduling (Viana and Pinho de Sousa, 2000).
- Flowshop scheduling (Marett and Wright, 1996).
- Cell formation problems (Hertz et al., 1994).
- Flight instructor scheduling problems (Xu et al., 2006).
- Aerodynamic shape optimization problems (Jaeggi et al., 2004).

To Know More About Multiobjective Tabu Search

Fred Glover and Manuel Laguna, **Tabu Search**, Kluwer Academic Publishers, Boston, Massachusetts, 1997.

Carlos A. Coello Coello, Gary B. Lamont and David A. Van Veldhuizen, **Evolutionary Algorithms for Solving Multi-Objective Problems**, Second Edition, Springer, New York, ISBN 978-0-387-33254-3, September 2007.

Ant System

The Ant System (AS) is a meta-heuristic inspired by colonies of real ants, which deposit a chemical substance on the ground called *pheromone* and was proposed by Marco Dorigo in the mid-1990s. The pheromone influences the behavior of the ants: paths with more pheromone are followed more often. From a computer science perspective, the AS is a multi-agent system where low level interactions between single agents (i.e., artificial ants) result in a complex behavior of the entire ant colony.

Ant System

The AS was originally proposed for the traveling salesman problem (TSP), and most of the current applications of the algorithm require the problem to be reformulated as one in which the goal is to find the optimal path of a graph. A way to measure the distances between nodes is also required in order to apply the algorithm.

Ant-Q

Gambardella and Dorigo (1995) realized the AS can be interpreted as a particular kind of distributed learning technique and proposed a family of algorithms called Ant-Q. This family of algorithms is really a hybrid between Q-learning and the AS. The algorithm is basically a reinforcement learning approach with some aspects incrementing its exploratory capabilities.

Ant System and Ant-Q

Some multiobjective versions of AS and Ant-Q are the following:

- Mariano and Morales (1999): proposed Multi-Objective Ant-Q (MOAQ), which uses lexicographic ordering.
- Gambardella et al. (1999): proposed the use of two ant colonies (one for each objective), and applied lexicographic ordering.
- Iredi et al. (2001): proposed a multi colony approach to handle the two objectives of a single machine total tardiness problem.
- Gagné et al. (2001): proposed an approach in which the heuristic values used to decide the movements of an ant take into consideration several objectives.

Ant System and Ant-Q

- T'kindt et al. (2002): proposed SACO, which adopts lexicographic ordering, and incorporates local search.
- Shelokar et al. (2000,2002): proposed a version of SPEA in which the search engine is an ant system. The approach adopts strength Pareto fitness assignment, an external archive, thermodynamic clustering for pruning the contents of the external archive, mutation, crossover, and a local search mechanism.
- Barán and Schaerer (2003): extends the MAC-VRPTW algorithm using a Pareto-based approach. All the objectives share the same pheromone trails, so that the knowledge of good solutions is equally important for every objective function. The approach maintains a list of Pareto optimal solutions, and each new generated solution is compared with respect to the contents of this list.

Ant System and Ant-Q

- Cardoso et al. (2003): proposed MONACO, which uses a multi-pheromone trail (the number of trails corresponds to the number of objectives to be optimized) and performs a local search over a partially built solution.
- Doerner et al. (2001,2004): proposed P-ACO, which uses a quadtree data structure for identifying, storing and retrieving nondominated solutions. Pheromone updates are done using two ants: the best and the second best values generated in the current iteration for each objective function.

Ant System and Ant-Q

- Guntsch and Middendorf (2003): proposed PACO, in which the population is formed with a subset of the nondominated solutions found so far. First, one solution is selected at random, but then the remainder solutions are chosen so that they are closest to this initial solution with respect to some distance measure. An average-rank-weight method is adopted to construct a selection probability distribution for the ants and the new derivation of the active population to determine the pheromone matrices.

Ant System and Ant-Q

- Doerner et al. (2003): proposed COMPETants, which was specifically designed for a bi-objective optimization problem and it consists of two ant populations with different priority rules. The first of these colonies uses a priority rule that emphasizes one of the objectives , and the second one emphasizes the other objective. The idea is to combine the best solutions from these two populations as to find good trade-offs.

Some Applications of Multiobjective Ant System or Ant-Q

- Optimization of a water distribution irrigation network (Mariano and Morales, 1999).
- Vehicle routing problems (Gambardella et al., 1999).
- Single machine total tardiness problem (Iredi et al., 2001).
- Industrial scheduling (Gravel et al. 2001).
- Reliability optimization (Shelokar et al., 2002).
- Portfolio selection problems (Doerner et al., 2004).
- Network optimization (Cardoso et al., 2003).

To Know More About Multiobjective Ant System or Ant-Q

C. García-Martínez, O. Cordón and F. Herrera, **A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP**, *European Journal of Operational Research*, Vol. 180, No. 1, pp. 116–148, July 1, 2007.

Carlos A. Coello Coello, Gary B. Lamont and David A. Van Veldhuizen, **Evolutionary Algorithms for Solving Multi-Objective Problems**, Second Edition, Springer, New York, ISBN 978-0-387-33254-3, September 2007.

Particle Swarm Optimization

James Kennedy and Russell Eberhart (1995) proposed an approach called "particle swarm optimization" (PSO) inspired by the choreography of a bird flock. This approach can be seen as a distributed behavioral algorithm that performs (in its more general version) multidimensional search. In the simulation, the behavior of each individual is affected by either the best local (i.e., within a certain neighborhood) or the best global individual.

Particle Swarm Optimization

It is worth mentioning that PSO is an unconstrained search technique. Therefore, it is also necessary to develop an additional mechanism to deal with constrained multiobjective optimization problems. The design of such a mechanism is also a matter of current research even in single-objective optimization (see for example (Ray, 2001)).

Particle Swarm Optimization

1. For i = 1 to M (M = population size)
   Initialize $P[i]$ randomly
   (P is the population of particles)
   Initialize $V[i] = 0$ (V = speed of each particle)
   Evaluate $P[i]$
   $GBEST$ = Best particle found in $P[i]$
2. End For
3. For i = 1 to M
   $PBESTS[i] = P[i]$
   (Initialize the "memory" of each particle)
4. End For

Particle Swarm Optimization

5. Repeat

   For i = 1 to M

   $V[i] = w \times V[i] + C_1 \times R_1 \times (PBESTS[i] - P[i])$

   $+C_2 \times R_2 \times (PBESTS[GBEST] - P[i])$

   (Calculate speed of each particle)

   (W = Inertia weight, C_1 & C_2 are positive constants)

   (R_1 & R_2 are random numbers in the range [0..1])

   $POP[i] = P[i] + V[i]$

   If a particle gets outside the pre-defined hypercube
   then it is reintegrated to its boundaries

   Evaluate $P[i]$

   If new position is better then $PBESTS[i] = P[i]$

   $GBEST$ = Best particle found in $P[i]$

   End For

6. Until stopping condition is reached

Particle Swarm Optimization

To extend PSO for multiobjective optimization, it is necessary to modify the guidance mechanism of the algorithm such that nondominated solutions are considered as leaders. Note however, that it's important to have a diversity maintenance mechanism. Also, an additional exploration mechanism (e.g., a mutation operator) may be necessary to generate all portions of the Pareto front (mainly in disconnected fronts).

Particle Swarm Optimization

Some multiobjective versions of particle swarm optimization are the following:

- Moore & Chapman (1999): Based on Pareto dominance. The authors emphasize the importance of performing both an individual and a group search (a cognitive component and a social component). No scheme to maintain diversity is adopted.
- Ray & Liew (2002): Uses Pareto dominance and combines concepts of evolutionary techniques with the particle swarm. The approach uses crowding to maintain diversity and a multilevel sieve to handle constraints.

Particle Swarm Optimization

- Coello & Lechuga (2002,2004): Uses Pareto dominance and a secondary population to retain the nondominated vectors found along the search process. The approach is very fast and has performed well compared to other techniques considered representative of the state-of-the-art in evolutionary multiobjective optimization.
- Fieldsend & Singh (2002): Also uses Pareto dominance and a secondary population. However, in this case, a data structure called "dominated trees" is used to handle an unconstrained archive, as to avoid the truncation traditionally adopted with MOEAs. A mutation operator (called "turbulence") is also adopted.

Particle Swarm Optimization

- Mostaghim and Teich (2003): proposed a sigma method (similar to compromise programming) in which the best local guides for each particle are adopted to improve the convergence and diversity of a PSO approach used for multiobjective optimization. They also use a "turbulence" operator, but applied on decision variable space. In further work, the authors study the influence of ϵ-dominance on MOPSO methods. In more recent work, Mostaghim and Teich (2004) proposed a new method called *covering*MOPSO (cvMOPSO), which works in two phases. In phase 1, a MOPSO algorithm is run with a restricted archive size and the goal is to obtain a good approximation of the Pareto-front. In phase 2, the nondominated solutions obtained from phase 1 are considered as the input archive of the cvMOPSO, and the aim is to cover the gaps left.

Particle Swarm Optimization

- Li (2003): proposed an approach that incorporates the main mechanisms of the NSGA-II into the PSO algorithm. In more recent work, Li (2004) proposed the *maximinPSO*, which uses a fitness function derived from the maximin strategy (Balling, 2003) to determine Pareto domination.

- Chow and Tsui (2004): A modified PSO called "Multi-Species PSO" is introduced by considering each objective function as a species swarm. A communication channel is established between the neighboring swarms for transmitting the information of the best particles, in order to provide guidance for improving their objective values.

Particle Swarm Optimization

- Alvarez-Benitez et al. (2005): proposed PSO-based methods based exclusively on Pareto dominance for selecting guides from an unconstrained nondominated archive. Three different tecnniques are presented: *Rounds* which explicitly promotes diversity, *Random* which promotes convergence and *Prob* which is a weighted probabilistic method and forms a compromise between *Random* and *Rounds*.

- Santana-Quintero et al. (2006): proposed a hybrid algorithm, in which particle swarm optimization is used to generate a few solutions on the Pareto front (or very close to it), and rough sets are adopted as a local search mechanism to generate the rest of the front.

Particle Swarm Optimization

- Peng and Zhang (2008) proposed the multi-objective particle swarm optimizer based on decomposition (MOPSO/D). This approach uses the framework adopted by MOEA/D but replaces the genetic operators (crossover and mutation) by the inertia flight equations used in traditional PSO. MOPSO/D uses a turbulence (or mutation) operator and adopts an archiving strategy (which is based on ϵ-dominance (Laumanns, 2002)) to store the nondominated solutions found during the search.

Particle Swarm Optimization

- Fuentes-Cabrera and Coello (2010) proposed a micro-MOPSO, which uses a population containing only five individuals and performs a relatively low number of evaluations (only 3000). This approach first selects the leader and then selects the neighborhood for integrating the swarm. It also performs a reinitialization process for preserving diversity and uses two external archives: one for storing the solutions that the algorithm finds during the search process and another for storing the final solutions obtained.

Particle Swarm Optimization

- Padhye et al. (2009) proposed the use of hypervolume contributions for guiding selection in a MOPSO. Jiang and Cai (2009) proposed the use of hypervolume for pruning archive solutions in the context of an ϵ-MOPSO. Chaman and Coello (2014) proposed the use of the hypervolume contribution of archived solutions for selecting each particle's global and personal leaders, and also as a mechanism for undaping the external archive when inserting new nondominated solutions into it.

Particle Swarm Optimization

- Nebro et al. (2009) proposed the Speed-constrained Multi-objective PSO (SMPSO), which adopts a constriction coefficient (Clerc & Kennedy, 2002) to limit the velocity. This approach is able to reach the true Pareto front of several test problems in which most MOPSOs fail.

Particle Swarm Optimization

- Today, a wide variety of MOPSOs exist, based on aggregating functions, decomposition, lexicographic ordering, quantum computing, speciation, co-evolution, sub-populations, Pareto ranking, and hybrids with other approaches (e.g., mathematical programming techniques).

Some Applications of Multiobjective Particle Swarm Optimization

- Optimal groundwater management (El-Ghandour & Elbeltagi, 2014).
- Optimize the design of efficient Automatic Train Operation (ATO) speed profiles in railway systems (Domínguez et al., 2014).
- Planning of electrical distribution systems incorporating distributed generation (Ganguly et al., 2013).
- Complex network clustering (Gong et al., 2014).
- Partial classification for accident severity analysis (Qiu et al., 2014).

To Know More About Multiobjective Particle Swarm Optimization

Margarita Reyes-Sierra and Carlos A. Coello Coello, **Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art**, *International Journal of Computational Intelligence Research*, Vol. 2, No. 3, pp. 287–308, 2006.

Andries P. Engelbrecht, **Fundamentals of Computational Swarm Intelligence**, John Wiley & Sons, Ltd, 2005, ISBN 978-0-470-09191-3.

Konstantinos E. Parsopoulos and Michael N. Vrahatis, **Multi-Objective Particles Swarm Optimization Approaches**, in Lam Thu Bui and Sameer Alam (editors), *Multi-Objective Optimization in Computational Intelligence: Theory and Practice*, pp. 20–42, Information Science Reference, Hershey, PA, USA, 2008.

Artificial Immune System

Computationally speaking, the immune system is a highly parallel intelligent system that is able to learn and retrieve previous knowledge (i.e., it has "memory") to solve recognition and classification tasks. Due to these interesting features, several researchers have developed computational models of the immune system and have used it for a variety of tasks.

Artificial Immune System

There are several computational models of the immune system, from which the main ones are the following:

- Immune network theory
- Negative selection
- Clonal selection theory

Artificial Immune System

The main issues to extend an artificial immune system to deal with multiple objectives are how to influence the propagation of antibodies (i.e., how to couple the Pareto selection mechanism) and how to maintain diversity. The use of a secondary population may also be useful, if possible in the model adopted.

Artificial Immune System (fitness scoring)

Repeat

1. Select an antigen $\mathcal{A}$ from PA
   (PA = Population of Antigens)
2. Take (randomly) R antibodies from PS
   (PS = Population of Antibodies)
3. For each antibody $r \in R$, match it against
   the selected antigen $\mathcal{A}$
   Compute its match score (e.g., using Hamming distance)
4. Find the antibody with the highest match score
   Break ties at random
5. Add match score of winning antibody to its fitness

Until maximum number of cycles is reached

Artificial Immune System

Some examples are the following:

- Yoo & Hajela (1999): Use of a linear aggregating function combined with the fitness scoring function previously indicated.
- Cui et al. (2001): Another hybrid approach that uses entropy to maintain diversity.
- Anchor et al. (2002): Adopt both lexicographic ordering and Pareto-based selection in an evolutionary programming algorithm used to detect attacks with an artificial immune system for virus and computer intrusion detection.

Artificial Immune System

- Luh et al. (2003): proposed the multi-objective immune algorithm (MOIA) which adopts several biologically inspired concepts. This is a fairly elaborate approach which adopts a binary encoding. Affinity of the antibodies is measured in such a way that the best antibodies are the feasible nondominated solutions. The approach has a germinal center where the nondominated solutions are cloned and hypermutated.
- Campelo et al. (2004): proposed the Multiobjective Clonal Selection Algorithm (MOCSA). This approach combines ideas from both CLONALG and opt-AINet. MOCSA uses real-numbers encoding, nondominated sorting, cloning, maturation (i.e., Gaussian mutation) and replacement (based on nondominated sorting).

Artificial Immune System

- Cutello et al. (2005): extended PAES with a different representation (*ad hoc* to the protein structure prediction problem of their interest) and with immune inspired operators. The original mutation stage of PAES, which consists of two steps (mutate and evaluate) is replaced by four steps: (1) a clonal expansion phase, (2) an affinity maturation phase, (3) an evaluation phase, and (4) a selection phase (the best solution is chosen).
- Coello and Cruz (2002,2005): extended a clonal selection algorithm to handle multiple objectives. A secondary population is adopted.

Artificial Immune System

- Jiao et al. (2005): proposed the Immune Dominance Clonal Multiobjective Algorithm (IDCMA), which is based on clonal selection and adopts Pareto dominance. The antigens are the objective functions and constraints that must be satisfied. The antibodies are the candidate solutions. The affinity antibody-antigen is based on the objective function values and the feasibility of the candidate solutions. The authors also determine an antibody-antibody affinity using Hamming distances. It adopts the "immune differential degree", which is a value that denotes the relative distribution of nondominated solutions in the population (similar to fitness sharing).

Artificial Immune System

- Lu et al. (2005): proposed the Immune Forgetting Multiobjective Optimization Algorithm (IFMOA), which adopts the fitness assignment scheme of SPEA, a clonal selection operator, and an Immune Forgetting Operator. The clonal selection operator implements clonal proliferation, affinity maturation, and clonal selection on the antibody population (the antibodies are the possible solutions to the problem).

- Freschi and Repetto (2005): proposed the Vector Artificial Immune System (VAIS). which is based on opt-aiNet. VAIS assigns fitness using the *strength* value of SPEA. After assigning fitness to an initially random population, the approach clones each solution and mutates them. Then, it applies a Pareto-based selection and the nondominated individuals are stored in an external memory.

Some Applications of Multiobjective Artificial Immune Systems

- Structural optimization (Yoo & Hajela, 1999).
- Computer security (Anchor et al., 2002).
- Multidisciplinary design optimization (Kurapati & Azarm, 2000).
- Unsupervised feature selection (Lu et al., 2005).
- Protein structure prediction problem (Cutello et al., 2005).
- Electromagnetic design (Campelo et al., 2004).

To Know More About Multiobjective Artificial Immune Systems

Fabio Freschi, Carlos A. Coello Coello and Maurizio Repetto, **Multiobjective Optimization and Artificial Immune Systems: A Review**, in Hongwei Mo (editor), *Handbook of Research on Artificial Immune Systems and Natural Computing: Applying Complex Adaptive Technologies*, pp. 1–21, Medical Information Science Reference, Hershey, New York, 2009, ISBN 978-1-60566-310-4.

Felipe Campelo, Frederico G. Guimaraes and Hajime Igarashi, **Overview of Artificial Immune Systems for Multi-Objective Optimization**, in Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu and Tadahiko Murata (editors), *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pp. 937–951, Springer. Lecture Notes in Computer Science Vol. 4403, Matshushima, Japan, March 2007.

Differential Evolution

Differential Evolution (DE) is a relatively recent heuristic (it was created in the mid-1990s) proposed by Kenneth Price and Rainer Storn which was designed to optimize problems over continuous domains. This approach originated from Kenneth's Price attempts to solve the Chebychev Polynomial fitting Problem that had been posed to him by Rainer Storn. In one of the different attempts to solve this problem, Price came up with the idea of using vector differences for perturbing the vector population.

Differential Evolution

DE is an evolutionary (direct-search) algorithm which has been mainly used to solve continuous optimization problems. DE shares similarities with traditional EAs. DE performs mutations based on the distribution of the solutions in the current population. In this way, search directions and possible stepsizes depend on the location of the individuals selected to calculate the mutation values.

Differential Evolution

There is a nomenclature scheme developed to reference the different DE variants. The most popular is called *"DE/rand/1/bin"*, where "DE" means Differential Evolution, the word "rand" indicates that individuals selected to compute the mutation values are chosen at random, "1" is the number of pairs of solutions chosen and finally "bin" means that a binomial recombination is used. This algorithm is shown in the following slide.

Differential Evolution

Begin
G=0. Create a random initial population $\vec{x}_{i,G}$ $\forall i$, $i = 1, \ldots, NP$
Evaluate $f(\vec{x}_{i,G})$ $\forall i$, $i = 1, \ldots, NP$
For G=1 to MAX_GEN **Do**
  For i=1 to NP **Do**
$\Rightarrow$ Select randomly $r_1 \neq r_2 \neq r_3$:
$\Rightarrow$ $j_{rand} = \mathrm{randint}(1, D)$
$\Rightarrow$ **For** j=1 to D **Do**
$\Rightarrow$ **If** $(rand_j[0,1) < CR$ **or** $j = j_{rand})$ **Then**
$\Rightarrow$ $u_{i,j,G+1} = x_{r_3,j,G} + F(x_{r_1,j,G} - x_{r_2,j,G})$
$\Rightarrow$ **Else** $u_{i,j,G+1} = x_{i,j,G}$
$\Rightarrow$ **End If**
$\Rightarrow$ **End For**
    If $(f(\vec{u}_{i,G+1}) \leq f(\vec{x}_{i,G}))$ **Then**
      $\vec{x}_{i,G+1} = \vec{u}_{i,G+1}$
    Else $\vec{x}_{i,G+1} = \vec{x}_{i,G}$ **End If**
  End For
  $G = G + 1$. **End For End**

Differential Evolution

The "CR" parameter controls the influence of the parent in the generation of the offspring. Higher values mean less influence of the parent. The "F" parameter scales the influence of the set of pairs of solutions selected to calculate the mutation value (one pair in the case of the algorithm shown in the previous slide).

Differential Evolution

Some multi-objective extensions of differential evolution are the following:

- Chang et al. (1999): Adopt an external archive to store the nondominated solutions obtained during the search. It incorporates fitness sharing to maintain diversity. The selection mechanism is modified in order to enforce that the members of the new generation are both nondominated and at a certain minimum distance from the previously found nondominated solutions.
- Abbass et al. (2001,2002): proposed the Pareto-frontier Differential Evolution (PDE) approach. It uses Pareto dominance, and enforces that only the nondominated individuals are retained in the population and recombined. A form of niching is also adopted. In a further paper, a self-adaptive version (SPDE) is proposed (crossover and mutation rates are self-adapted).

Differential Evolution

- Madavan (2002): proposed the Pareto-Based Differential Evolution approach which incorporates the nondominated sorting and ranking selection procedure from the NSGA-II. Once the new candidate is obtained using DE operators, the new population is combined with the existing parents population and then the best members of the combined population (parents plus offspring) are chosen.
- Xue et al. (2003,2004): proposed the Multi-Objective Differential Evolution (MODE) approach, in which the best individual is adopted to create the offspring. A Pareto-based approach is introduced to implement the selection of the best individual. If a solution is dominated, a set of nondominated individuals can be identified and the "best" turns out to be any individual (randomly picked) from this set.

Differential Evolution

- Iorio and Li (2004): proposed the Nondominated Sorting Differential Evolution (NSDE), which is a simple modification of the NSGA-II. In further work, Iorio and Li (2006) proposed a variation of NSDE that incorporates directional information regarding both convergence and spread. For convergence, the authors modify NSDE so that offpsring are generated in the direction of the previously generated solutions with better rank. For spread, the authors modify NSDE so that it favors the selection of individuals from different regions of decision variable space.

Differential Evolution

- Kukkonen and Lampinen (2004): proposed a revised version of Generalized Differential Evolution (GDE). The basic idea in this selection rule is that the trial vector is required to dominate the old population member used as a reference either in constraint violation space or in objective function space. If both vectors are feasible and nondominated with respect to each other, the one residing in a less crowded region is chosen to become part of the population of the next generation.

Differential Evolution

- Kukkonen and Lampinen (2005) introduced GDE3, which is a new version of Generalized Differential Evolution that can handle both single- and multi-objective optimization problems (either constrained or unconstrained). The selection mechanism in GDE3 considers Pareto dominance (in objective function space) when comparing feasible solutions, and weak dominance (in constraint violation space) when comparing infeasible solutions. Feasible solutions are always preferred over infeasible ones, regardless of Pareto dominance.

Differential Evolution

- Robič and Filipič (2005): proposed an approach called Differential Evolution for Multi-Objective Optimization (DEMO), which combines the advantages of DE with the mechanisms of Pareto-based ranking and crowding distance sorting (from the NSGA-II). DEMO only maintains one population and it is extended when newly created candidates take part immediately in the creation of the subsequent candidates.
- Santana-Quintero and Coello Coello (2005): proposed the ϵ-MyDE, which uses Pareto ranking and ϵ-dominance. In a further paper, Hernandez et al. (2006), hybridize ϵ-MyDE with rough sets.

Differential Evolution

- Landa Becerra and Coello Coello (2006): proposed the use of the ε-constraint technique hybridized with a single-objective evolutionary optimizer: the cultured differential evolution.
- Li and Zhang (2006): proposed a multi-objective differential evolution algorithm based on decomposition (MODE/D) for continuous multi-objective optimization problems with variable linkages. The authors use the weighted Tchebycheff approach to decompose a multi-objective optimization problem into several scalar optimization subproblems. The differential evolution operator is used for generating new trail solutions, and a neighborhood relationship among all the subproblems generated is defined, such that they all have similar optimal solutions.

Some Applications of Multiobjective Differential Evolution

- Concurrent design of pinion–rack continuously variable transmission (Portilla-Flores, 2006).
- Classification (Abbass, 2001).
- Fine-tuning of the fuzzy automatic train operation (ATO) for a typical mass transit system (Chang et al., 1999).

To Know More About Multiobjective Differential Evolution

Efrén Mezura-Montes, Margarita Reyes-Sierra and Carlos A. Coello Coello, **Multi-Objective Optimization using Differential Evolution: A Survey of the State-of-the-Art**, in Uday K. Chakraborty (Editor), *Advances in Differential Evolution*, pp. 173–196, Springer, Berlin, 2008, ISBN 978-3-540-68827-3.

Carlos A. Coello Coello, Gary B. Lamont and David A. Van Veldhuizen, **Evolutionary Algorithms for Solving Multi-Objective Problems**, Second Edition, Springer, New York, ISBN 978-0-387-33254-3, September 2007.

Promising areas of future research

- Can we produce MOEAs that perform a very low number of objective function evaluations and can handle problems of large dimensionality?
- How to deal with expensive objective functions (e.g., surrogates)?
- Can we provide a solid theoretical foundation to this field?

Promising areas of future research

There are plenty of fundamental questions that remain unanswered. For example:

- What are the sources of difficulty of a multi-objective optimization problem for a MOEA?
- Can we benefit from hybridizing multi-objective metaheuristics with mathematical programming techniques.
- Can we use alternative mechanisms into an evolutionary algorithms to generate nondominated solutions without relying on Pareto ranking?

Promising areas of future research

- How to deal with uncertainty?
- How to deal with dynamic multi-objective optimization problems?
- What about incorporating preferences from the user?
- What is the most appropriate type of algorithm for a particular problem?

To know more about evolutionary multiobjective optimization

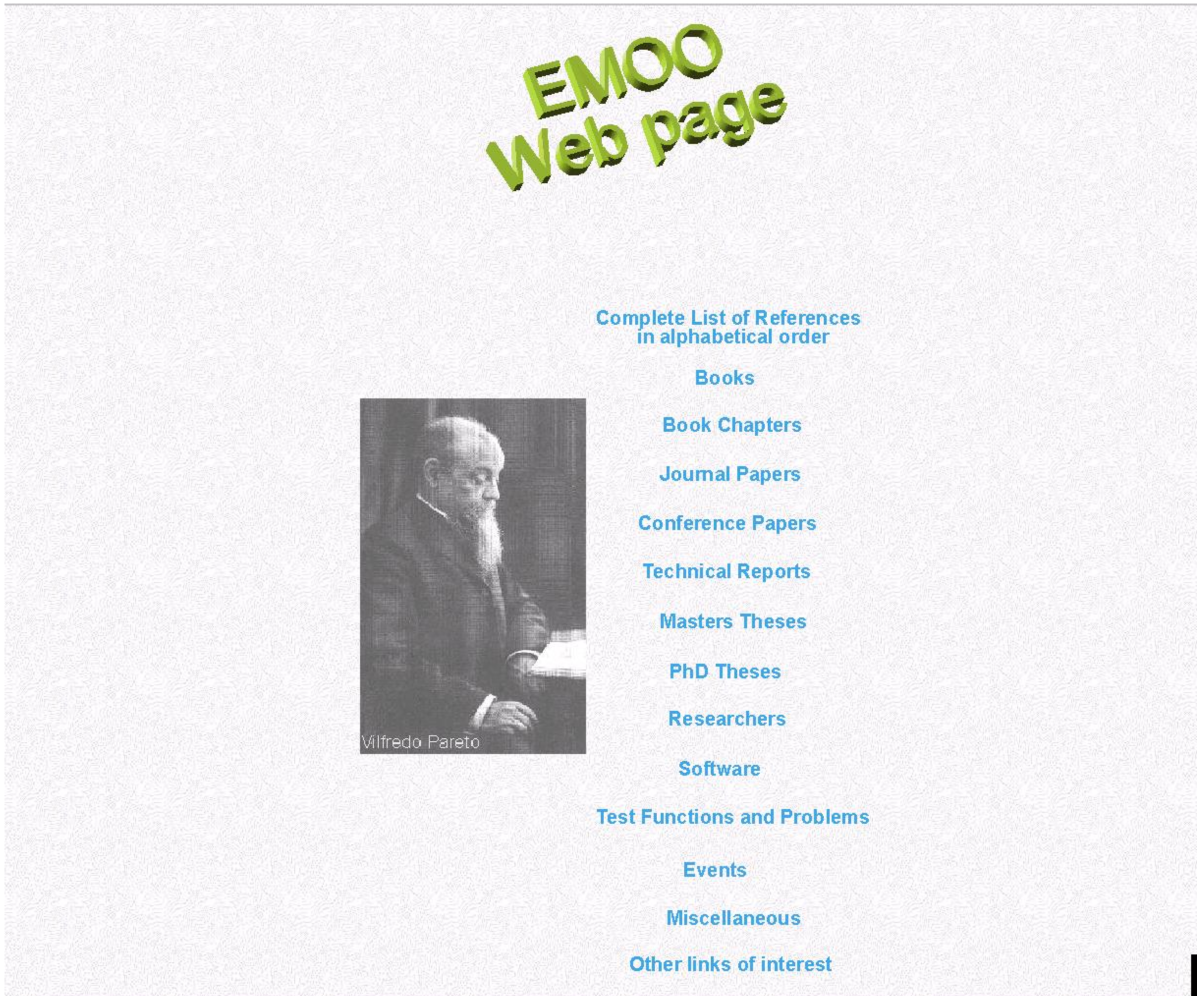
Please visit our EMOO repository located at:

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 10180 bibliographic references including 291 PhD theses, over 4685 journal papers and over 3790 conference papers.
- Contact info of 79 EMOO researchers
- Public domain implementations of SPEA, NSGA-II, the microGA, MOPSO, MISA, AMOSA and PAES, among others.