Evolutionary Multiobjective Optimization

- Particularities of multiobjective optimization
- Methods for multiobjective optimization
- Evolutionary algorithms for multiobjective optimization

Particularities

Multiobjective optimization = simultaneous optimization of several criteria

Examples:

1. Find the parameters of a product which simultaneously maximize the reliability and minimize the cost.

2. Solve a routing problem in a communication network such that both the costs and the network congestion are minimized.

3. Identify clusters in data such that the similarity between data in the same cluster is maximized while the similarity between data belonging to different clusters is minimized

Particularities

Problem: $f:R^n \rightarrow R^r$, $f(x)=(f_1(x),\ldots,f_r(x))$

Find x* which satisfies:

(i) Inequality constraints: $g_i(x^*) \geq 0$, $i=1..p$
(ii) Equality constraints: $h_i(x^*)=0$, $i=1..q$
(iii) Optimize (maximize or minimize each criterion)

Rmk: the criteria can be conflicting (e.g.: quality and price of a product: a higher quality usually implies a higher price) → solutions optimizing all of them might not exist

Particularities

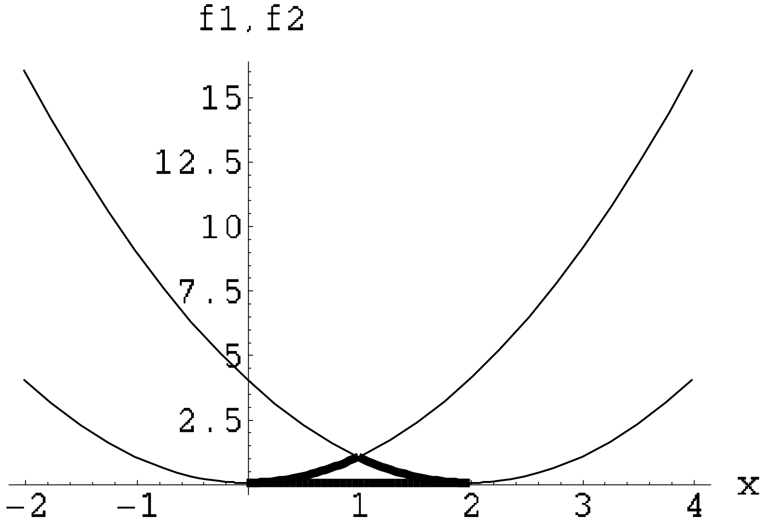
Example: r=2, $f_1(x)=x^2$, $f_2(x)=(x-2)^2$ for x in [-2,4]

Problem: Minimize both functions.

Remark: It does not exist x* which minimizes simultaneously both functions (the minimum of f_1 is in 0 while the minimum of f_2 is in 2)

Idea: Find trade-off solutions (no Better configurations can be found)

e.g. x which satisfies:
x in [0,2] – does not exist x'
such that
$f_1(x')<f_1(x)$ and $f_2(x')<f_2(x)$



Such a trade-off solution
Is called Pareto solution

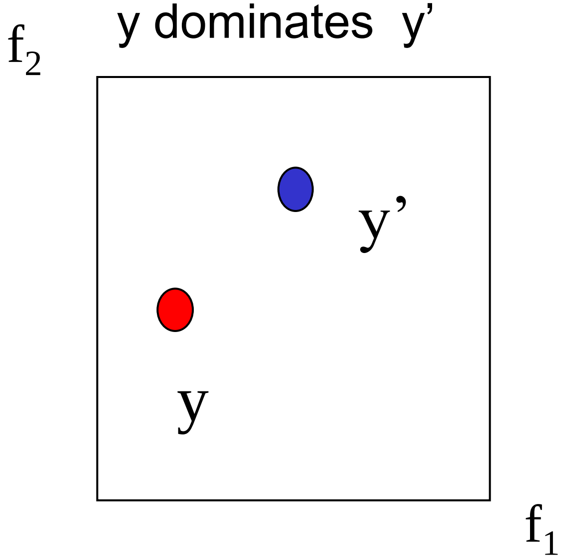
Basic notions in Pareto optimization

1. Domination:

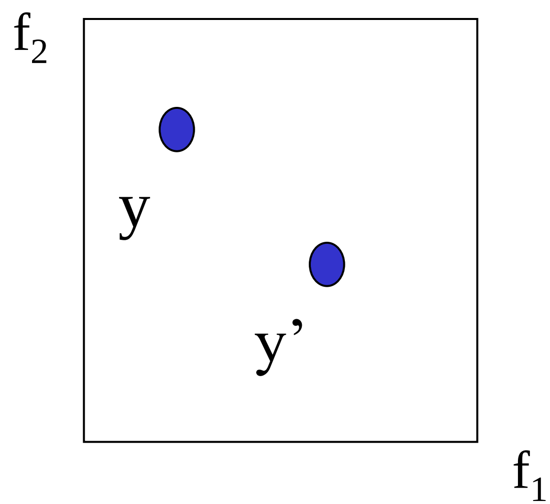
A vector y dominates another vector y' (for a minimization problem)

if y_i<=y'_i for each i and the inequality is strict for at least one component

y dominates y'



y does not dominate y' and
y' does not dominate y
(they are reciprocally non-dominated

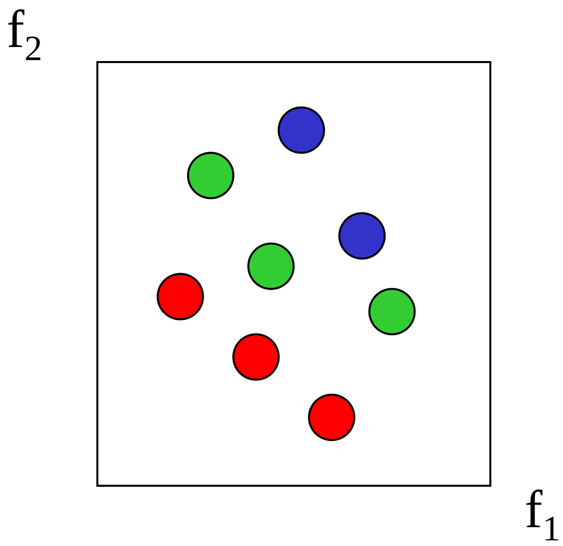


The domination relation is a partial order relation

Basic notions in Pareto optimization

2. Non-dominated element (with respect to a set):

y is non-dominated with respect to V if there is no element in V which dominates y



The red elements are globally non-dominated

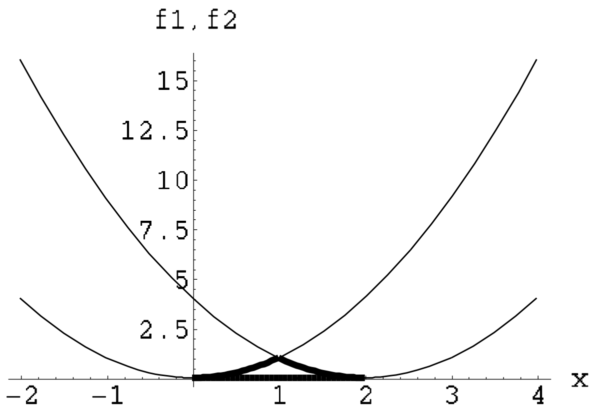
The green elements are non-dominated with respect to the set of green and blue elements but are dominated by all red elements

Basic notions in Pareto optimization

3. Pareto optimal solution

An element x is called Pareto optimal solution if there is no element x' such that f(x') dominates f(x)

The set of all Pareto optimal elements for a multiobjective optimization problem is called Pareto optimal set.

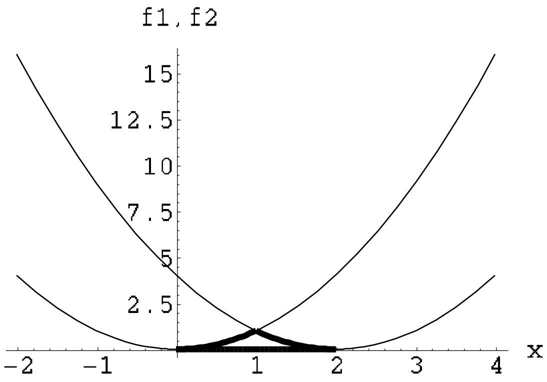


In this case the Pareto optimal set is the interval [0,2] (any element of this interval is equally good when both objective functions are taken into account)

Basic notions in Pareto optimization

4. Pareto Front

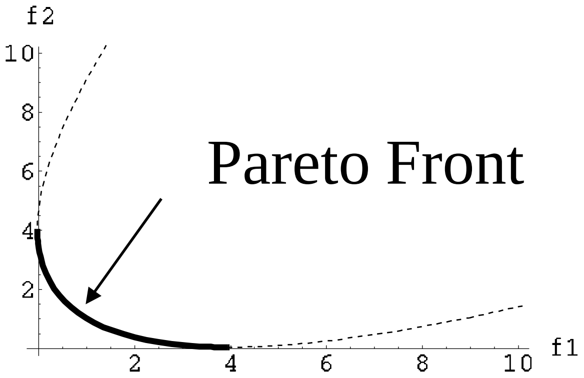
The set of values of the objective functions for all elements in the Pareto optimal set is called Pareto front



$\{(f_1(x),f_2(x))|$ x in the Pareto optimal set$\}$

Example:

$\{(x^2,(x-2)^2)$ | x in $[0,2]\}$



Traditional approaches

1. Transform the multi-objective optimization problem in a single-objective optimization problem : combine all criteria in just one criterion

- Aggregation method

$$\overline{f}(x) = \sum_{i=1}^{r} w_i f_i(x), \quad w_i \in (0,1), \quad \sum_{i=1}^{r} w_i = 1$$

Advantages: the problem to be solved is easier

Disadvantages:

- For a set of parameters w one obtains just one solution – in order to obtain an approximation of the Pareto set several problems should be solved (for various values of w)
- The user has to specify appropriate values for w

Traditional approaches

1. Transform the multi-objective optimization problem in a single-objective optimization problem : combine all criteria in just one criterion

- Goal attainment method:

$$\overline{f}(x) = (\sum_{i=1}^{r} |f_i(x) - y_i^*|^p)^{1/p}, \quad y_i^* - \text{goal values}$$

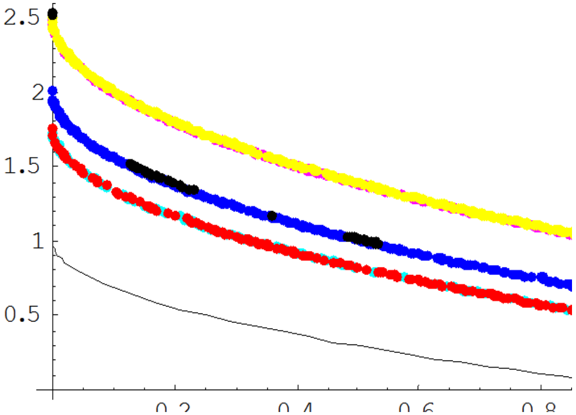
Advantages: the problem to be solved is easier

Disadvantages: the goal values should be known

Pareto optimization

2. Approximate the Pareto optimal set by using a population of candidates

- Apply an EA in order to generate elements approaching the Pareto optimal set (and the corresponding Pareto front)
- The Pareto front approximation should satisfy the properties:
  - Closeness to the real Pareto front
  - Uniform spread of points
  - Good covering of the Pareto front



Pareto Optimization – specific mechanisms

There are several techniques which can be used in order to obtain Pareto fronts satisfying the previous conditions:

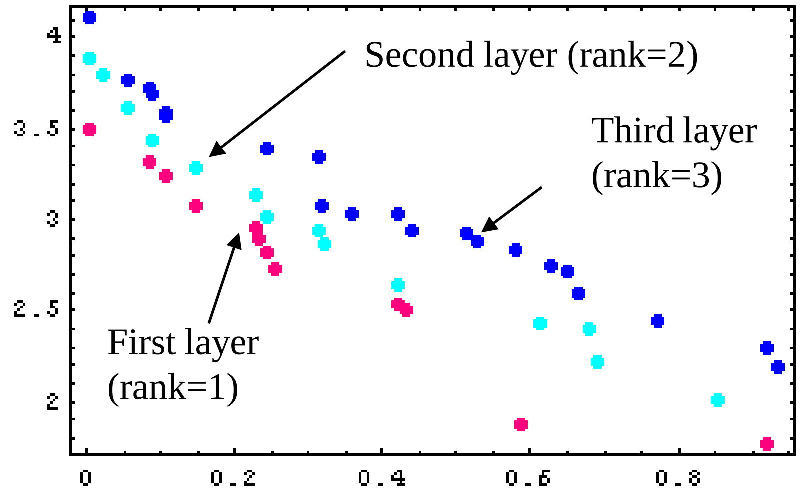
- Use a selection process based on the non-dominance relation
- Use a crowding factor to stimulate the elements in less crowded regions
- Change the fitness function by using a sharing mechanism
- Mating restriction
- Ensure the elitism (e.g. use an archive)

Pareto Optimization – specific mechanisms

Selection criteria:

- Use a nondomination rank (ex: NSGA – Nondominated Sorting GA)
  - The population is organized in nondomination layers:
    - The first layer consists of nondominated elements
    - The nondominated elements in the set obtained by removing the first layer represent the second layer etc.

- An element is better if its nondomination rank is smaller
- During the selection process the parents population is joined with offspring population and this set is increasingly sorted by rank.



Pareto Optimization – specific mechanisms

Selection criteria: use a fitness value based on the non-dominance relationship

- The fitness value of an element will depend on:
  - The number of elements which it dominates (direct relationship)
  - The number of elements which dominate it (inverse relationship)

Ex: SPEA – Strength Pareto EA

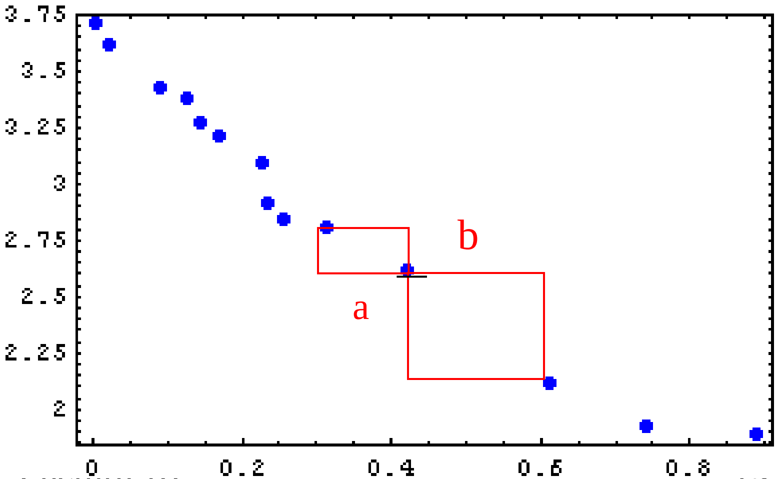
Pareto Optimization – specific mechanisms

Using a crowding factor

- Aim: stimulate the diversity of the Pareto front

- Idea: between two elements belonging to the same non-domination layer the one having a smaller crowding factor is preferred
- The crowding factor for an element is computed by using the distance between that element and the closest neigbours.

Value of the crowding factor:
(a+b)/2



Pareto Optimization – specific mechanisms

Sharing mechanism:

- Idea: if a group of individuals share the same resource their survival chance is increasing with the amount of the resource and is decreasing with the size of the group

- The fitness of an element is adjusted by dividing its value by a function (called sharing function) which depends on the distances between the elements in the group.

$$fit_i^{(s)} = \frac{fit_i}{\sum_{j=1}^{m} s(d(x_i, x_j))}, \quad s(d) = \begin{cases} 1-(d/\sigma_s)^{\alpha} & d < \sigma_s \\ 0 & d \geq \sigma_s \end{cases}$$

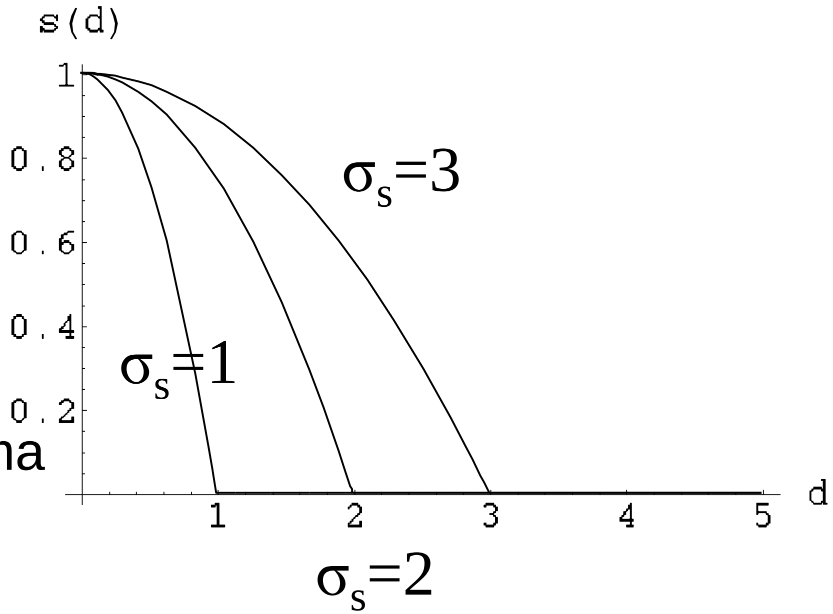
Pareto Optimization – specific mechanisms

Sharing mechanism:

- Allows to differentiate elements which are mutually non-dominated
- The main disadvantage is the need to specify the niche radius (σ_s)

$$s(d) = \begin{cases} 1-(d/\sigma_s)^\alpha & d < \sigma_s \\ 0 & d \geq \sigma_s \end{cases}$$

It is also beneficial in the case of multimodal optimization (when the goal is to approximate several optima or even all local optima



Pareto Optimization – specific mechanisms

Restricted mating:

- Idea: the crossover is accepted only for elements which satisfy some restrictions

- Goal: avoid the generation of low fitness configurations and/or stimulates the diversity (of the Pareto set)

Examples:

1. Accept as parents only elements which are similar enough

2. Accept as parents only nondominated elements

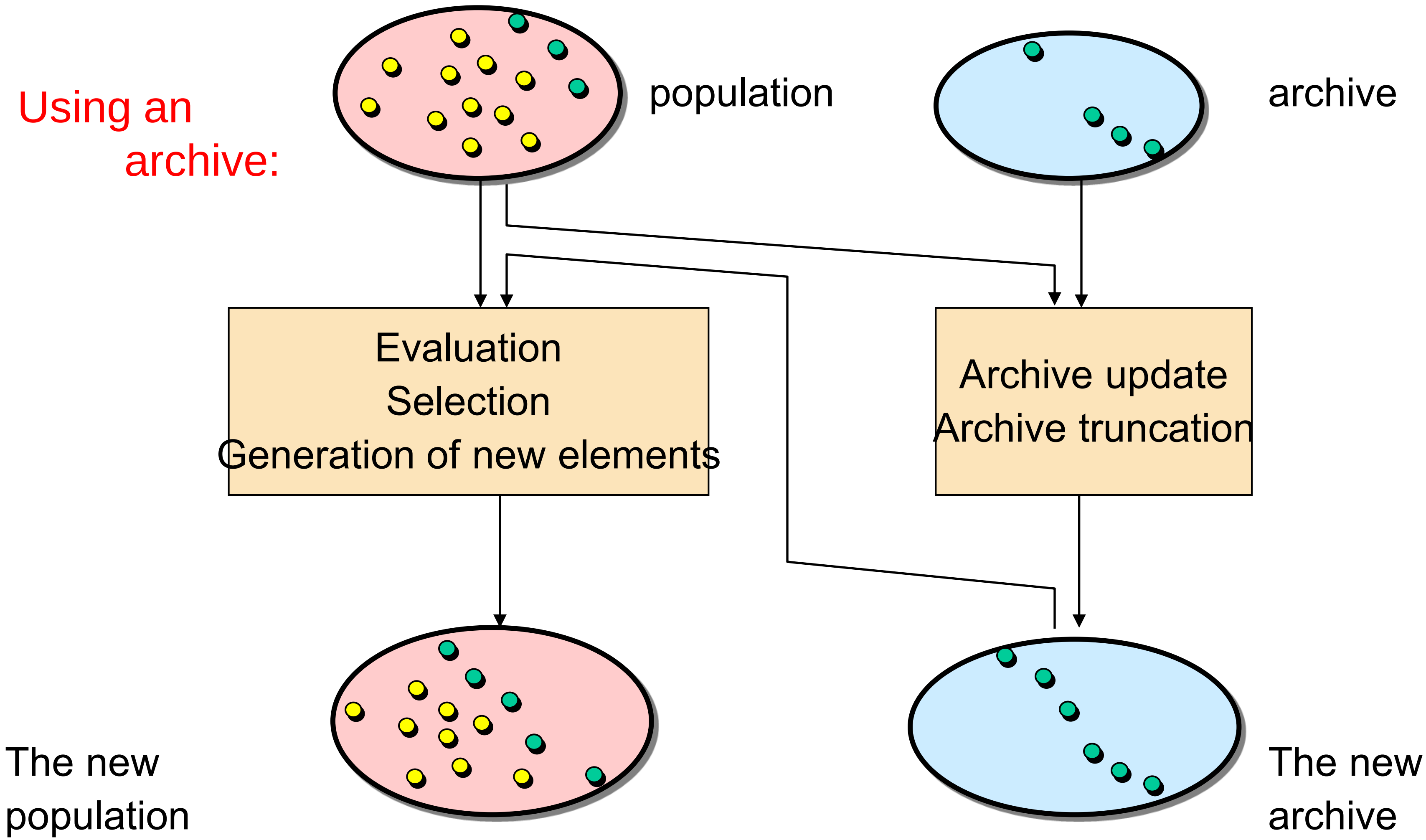
Pareto Optimization – specific mechanisms

Archiving:

- Aim: elitism (preserve the nondominated elements discovered during the evolution)
- The archive will be the approximation of the Pareto optimal set

- Disadvantage: managing the archive requires additional processing steps:
  - A new element is accepted in the archive only if it is nondominated
  - If the new element dominates some elements in the archive then these dominated elements are eliminated
  - In order to avoid the unlimited increase of the archive it should be periodically reorganized (e.g. by clustering or by eliminating some of the elements which are too similar to other elements)

Archive usage



Example: VEGA (Vector Evaluating Genetic Algorithm)

- First multiobjective evolutionary algorithm (1985) which does not use aggregation

- The general structure and crossover and mutation operators are similar to those used for single objective optimization

- The only specific element is the selection process:
  - In the case of r criteria the selection operator (e.g. Proportional selection) is applied for r times, each time with respect to another optimization function; in this way there are constructed r subpopulations (each one with m/r elements, m being the population size)

- Advantage: easy to implement
- Disadvantage: not effective for non-convex Pareto fronts

Example: Lexicographic Ordering

- A priority is associated to each criterion
- The single objective problems corresponding to all criteria are independently solved in decreasing order of priorities
- Advantage: easy to implement
- Disadvantage: it requires priorities
- Remarks: if priorities are not available the criteria can be selected randomly

Example: Multiobjective GA (MOGA)

- Proposed in 1993
- Each element has a rank wich is proportional with the number of other elements in the population which dominate it. The fitness is higher if the rank is smaller (all nondominated elements will be considered equally good)
- It uses a sharing function and restricted mating.

Example: Nondominating Sorting GA (NSGA)

- Variants: NSGA-I (1993), NSGA-II (2000), NSGA-III (2014)

- Each element has a nondomination rank which is computed based on the identification of some nondomination layers:
  - The nondominated elements from the current population represent the first nondomination layer
  - The nondominated elements from the set obtained after removing the elements of the first layer represent the the second layer etc.
- For population diversity stimulation:
  - NSGA-I uses a sharing function
  - NSGA-II uses a crowding factor
- NSGA-III : adapted for many objectives optimization (based on the idea of using reference points)

Example: Niched Pareto GA (NPGA)

- Proposed in: 1994

- It uses a tournament selection based on checking the nondomination property with respect to a population sample:
  - Random selection of two elements from the current population (x_1 and x_2)
  - Random selection of a sample with q elements from the current population
  - Check if x_1 and x_2 are dominated by elements from the sample
    - If x_1 is nondominated but x_2 is dominated then select x_1
    - If x_2 is nondominated but x_1 is dominated then select x_2
    - If both of them are dominated or both of them are non-dominated then is used a sharing function to decide which element should be selected.

Example: Strength Pareto EA (SPEA)

- First version (SPEA 1) was proposed in: 1999

- It uses an archive of nondominated elements which is updated during the evolutionary process
  - For each element in the archive is computed a so-called strength factor which is proportional to the number of elements in the population which are dominated by the archive element
  - The fitness of a population element is inverse proportional with the sum of the strength factors of all archive elements which dominate it
- SPEA 2 has some improvements with respect to SPEA 1:
  - When an element is evaluated both the dominating and the dominated elements are counted.
  - It uses a technique for estimating the Pareto front density (based on the distance to the nearest neighbor).
  - It uses an archive truncation technique

Example: decomposition based techniques (MOEA/D)

- Particularity: the multi-objective optimization problem is transformed in several single objective optimization problems

- MOEA/D has been proposed in 2007 and currently is one of the most effective multi-objective optimization algorithms (especially in the case of more than two criteria)
- MOEA/D idea:
  - It uses N weight vectors $(w_1,w_2,...,w_N)$; each vector contains r values (r=number of criteria)
  - For each of the r criteria there is known a reference value (the reference value for criteria i is denoted y_j^*)
  - For each vector w_i is solved a single-objective problem:
    - minimize $\max_j\{w_{ij} |f_j(x)-y_j^*|\}$ (Cebisev criterion)

Example: decomposition based techniques (MOEA/D)

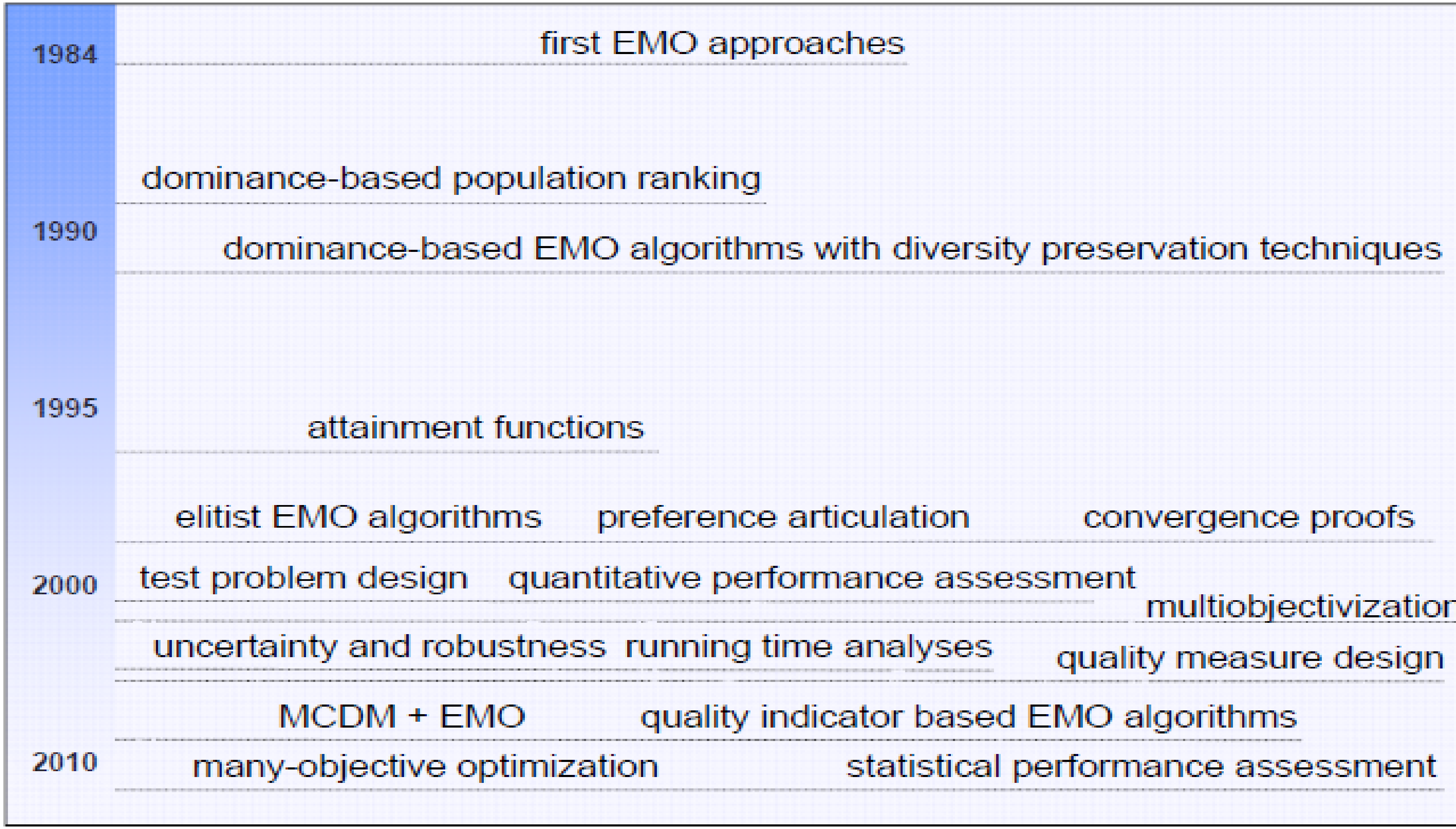
- Elements to be specified in MOEA/D:
  - N = number of subproblems
  - N weight vectors (they should be distributed as uniformly as possible)
  - T = number of neighbors of a weight vector

Example: decomposition based techniques (MOEA/D)

MOEA/D general structure:

- Initialize N populations; initialize a vector with reference values $z=(z_1,\ldots,z_r)$; initialize an archive (empty set)
- At each generation, for each subproblem i:
  - Select x_k and x_q from the populations corresponding to the subproblems which are in the neighborhood of problem i (the neighborhood is defined based on the distances between the weight vectors)
  - Combine x_k with x_q and construct y
  - Update the vector of reference values z (if it is necessary)
  - Replace the neighbors of y with y (if it is better)
  - Remove the dominated elements from the archive and add y to the archive (if it is non-dominated)

Summary



D.Brockhoff - Tutorial MOEA – GECCO 2013

Summary

Bibliographical resource:

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

(2940 references – september 2007)

(3519 references – november 2008)

(4388 references – october 2009)

(4861 references – february 2010) – no updates

(7806 references – march 2013) – [source: tutorial on MOEA – GECCO 2013]