Università di Catania
Dipartimento di Ingegneria Informatica e delle Telecomunicazioni

A Brief Introduction to Multiobjective Optimization Techniques

Maurizio Palesi

Introduction

- Most real-world engineering optimization problems are multiobjective in nature
- Objectives are often conflicting
  - Performance vs. Silicon area
  - Quality vs. Cost
  - Efficiency vs. Portability
  - …
- The notion of ”*optimum*” has to be re-defined

Statement of the Problem

- Multiobjective optimization (multicriteria, multiperformance, vector optimization)
  - Find a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions
  - Objectives are usually in conflict with each other
  - **Optimize**: finding solutions which would give the values of all the objective functions acceptable to the designer

Mathematical Formulation

- Find the vector

$$\bar{x}^* = \left[x_1^*, x_2^*, \ldots, x_n^*\right]^T$$

- Which will satisfy the *m* inequality constraints

$$g_i(\bar{x}) \geq 0 \quad i = 1,2,\ldots,m$$

- The *p* equality constraints

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

- And optimizes the vector function

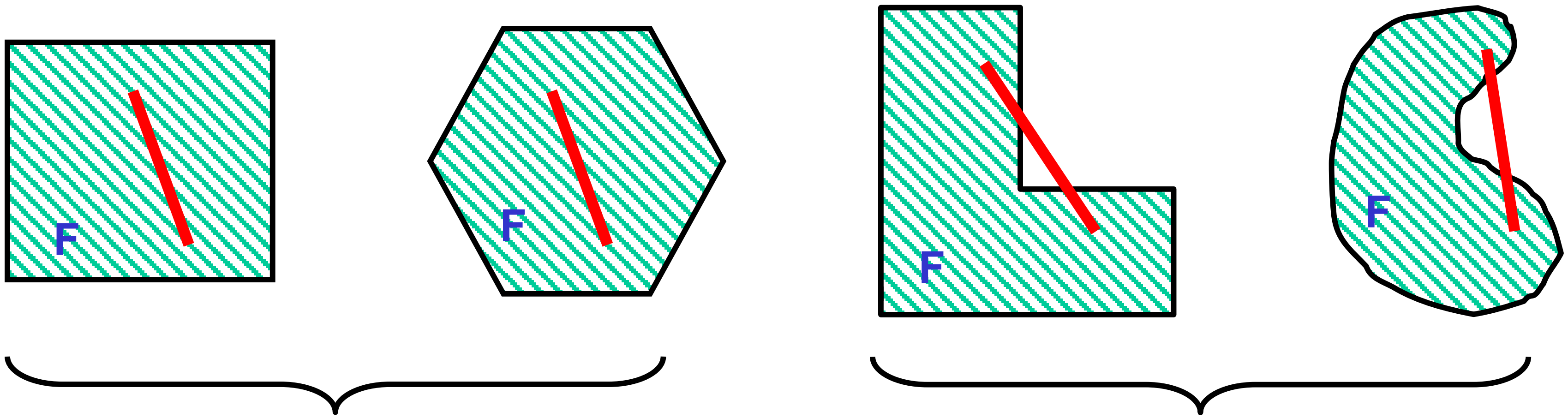
$$\bar{f}(\bar{x}) = \left[f_1(\bar{x}), f_2(\bar{x}), \ldots, f_k(\bar{x})\right]^T$$

Feasible Region

$$g_i(\overline{x}) \geq 0 \quad i = 1,2,\ldots,m$$

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

Define the *feasible region F*



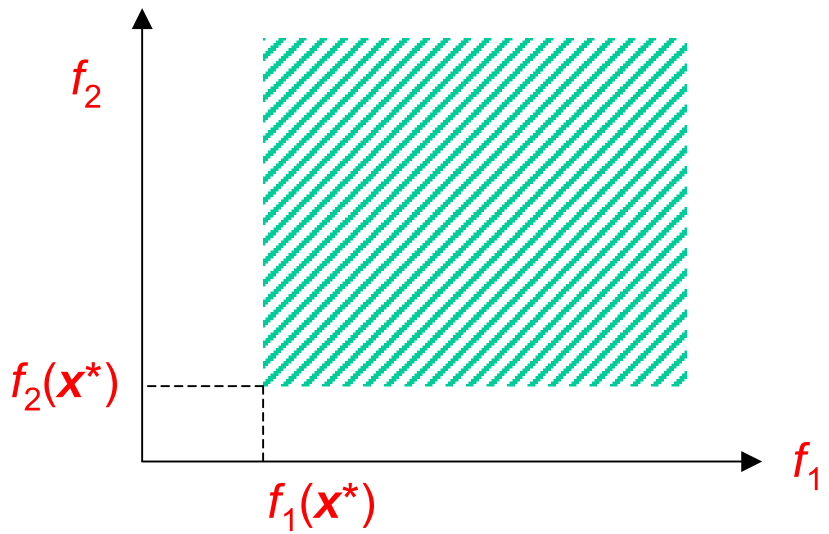
Convex sets

Non-convex sets

Meaning of Optimum

- We rarely have an **x*** such that

$$f_i\left(\bar{x}^*\right) \leq f_i\left(\bar{x}\right) \quad \forall\, \bar{x} \in F, i = 1, 2, \ldots, k$$



- We have to estabilish a certain criteria to determine what would be considered as an optimal solution

Pareto Optimum

- Fornulated by Vilfredo Pareto in the XIX century
- A point $\overline{x}^* \in F$ is *Pareto optimal* if for every $\overline{x} \in F$ either

$$f_i(\overline{x}) = f_i(\overline{x}^*), \quad i = 1,2,\ldots,k$$

or, there is at least one $i \in \{1,2,\ldots,k\}$ such that

$$f_i(\overline{x}) > f_i(\overline{x}^*)$$

Vilfredo Pareto 1848-1923

Pareto Optimum

- In words, this definition says that $\overline{x}^*$ is *Pareto optimal* if there exists no feasible vector of decision variables $\overline{x}^* \in F$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion


Vilfredo Pareto 1848-1923

Pareto Optimum

Vilfredo Pareto 1848-1923

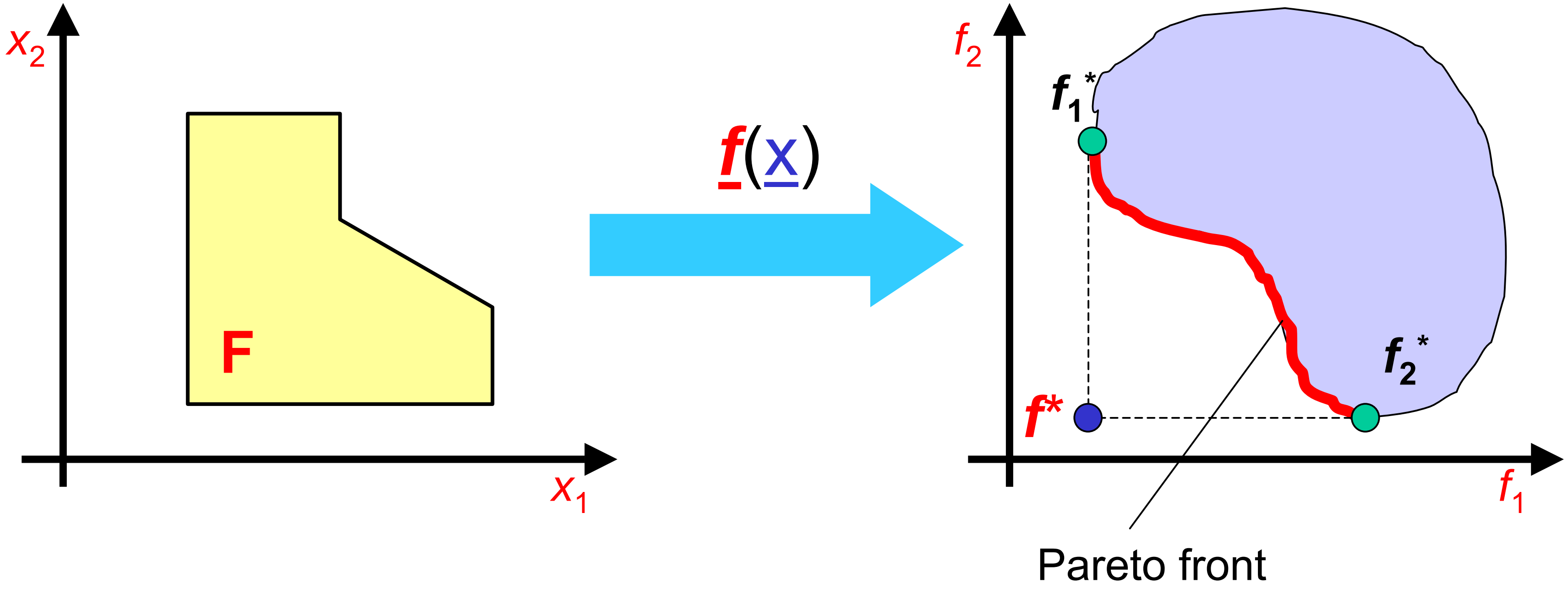
- A solution $x \in F$ is said to dominate $y \in F$ if
  - ➔ x is better or equal to y in all attributes
  - ➔ x is strictly better than y in at least one attribute
- Formally, x dominate y ($x \succ y$)

$$f_i(\bar{x}) \leq f_i(\bar{y}), \quad i = 1,2,\ldots,k$$

$$\exists\, j \in \{1,2,\ldots,k\} : f_j(\bar{x}) < f_j(\bar{y})$$

- The *Pareto set* consists of solutions that are not dominated by any other solutions

Pareto Front



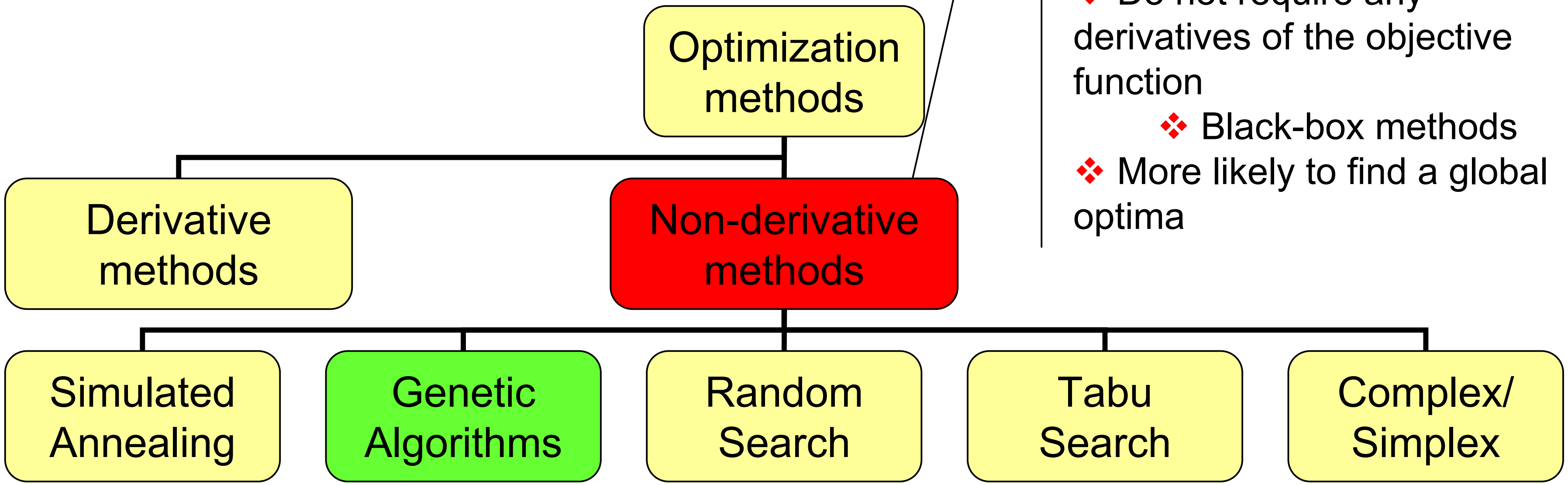
Current State of the Area

- 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)

General Optimization Methods



- More suitable for general engineering design problems
- Do not require any derivatives of the objective function
  - Black-box methods
- More likely to find a global optima

Why Evolutionary Algorithms?

- Most suitable to handle multi modal function landscapes and to identify multiple optima in a robust manner
- They are an adaptive approach
  - They are of general application and do not require detailed knowledge of the problem
  - A GA can be said to build itself a representation of the problem and solve it
- They learn by experience
  - They solve a problem by successive refinement. Starting from a set of random solutions, the process of evolution generates a learning process
- They are efficient at solving complex problems
  - Growing interest from researchers with various backgrounds to solve problems of all kinds and levels of complexity
- Parallelization
  - The complexity of the approach often lies in evaluation of the fitness functions of the individuals in each generation. This procedure can be parallelised quite simply, as it is possible to assign individual fitness values independently, so concurrent execution of this operation does not cause conflict
  - Simple to execute them parallely even on a loosely coupled NOW (Network Of Workstations) without much communication overhead

Multiobjective Optimization Based on GA

- Naïve approaches
  - ➔Weighted Sum Approach
  - ➔Goal Programming
  - ➔Goal Attainment
  - ➔The ε-constraint Method
- Non-Aggregating Approaches that are not Pareto-Based
  - ➔VEGA
  - ➔Lexicographic Ordering
  - ➔Game Theory
- Pareto-Based Approaches
  - ➔SPEA2

Naïve Approaches

- A GA needs scalar fitness information to work
- Combine all the objectives into a single one
  - ➔ Provide accurate scalar information about the range of the objectives
    - ✓ Avoid having one of them to dominate the others
    - ✓ Behaviour of each of the objective functions
      - – Very expensive process!
  - ➔ No further interaction with the decision maker is required

Weighted Sum Approach

- The MOO problem is transformed into a scalar optimization problem

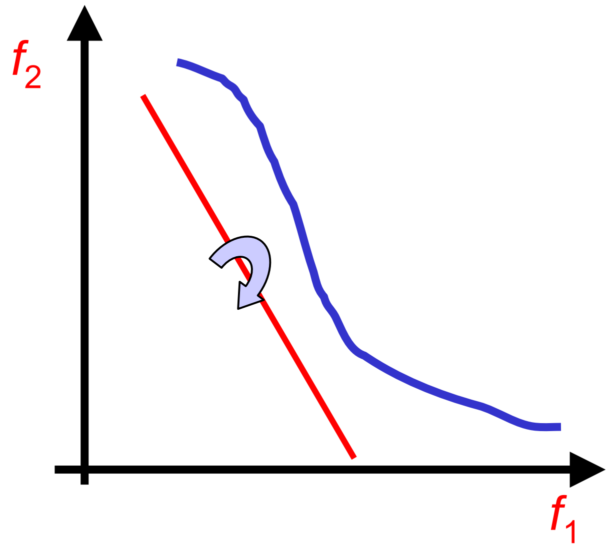
$$\min \sum_{i=1}^{k} w_i f_i(\overline{x})$$

- Where $w_i \geq 0$ are the weighting coefficients representing the relative importance of the objectives
  - ➔It is usually assumed that $\sum_{i=1}^{k} w_i = 1$

Weighted Sum Approach (cnt'd)

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

- Results can vary significantly as the weighting coefficients change
- How to choose these coefficients?
  - ➔ Solve the same problem for many different values of w_i
    - ✓ Interaction with the decision maker is required



Weighted Sum Approach (cnt'd)

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

- w_i do not reflect the importance of the objectives
  - ➔All functions should be expressed in units of approximately the same numerical values

$$\min \sum_{i=1}^{k} w_i f_i(\overline{x}) c_i$$

  - ➔The best results are usually obtained if $c_i = 1/ f_i^0$

Weighted Sum Approach (cnt'd)

- Very efficient (computationally speaking)
- Generates a strongly non-dominated solution that can be used as an initial solution for other techniques
- Problems
  - ➔ Solution quite often appear only in some parts of the Pareto front, while no solutions are obtained in other parts
  - ➔ Cannot find solution on non-convex parts of the Pareto front
    - ✓ Convex combination of objectives where the sum of all weights is costant and negative weights are not allowed

Goal Programming

- The DM has to assign targets or goal (*T*) that he/she wishes to achieve for each objective
- Minimize the absolute deviations from the targets to the objectives

$$\min \sum_{i=1}^{k} \left| f_i(\bar{x}) - T_i \right|$$

Goal Programming (cnt'd)

$$\min \sum_{i=1}^{k} \left| f_i(\bar{x}) - T_i \right|$$

- Very efficient (computationally speaking) because it does not require any non-dominance comparison
- Problems
  - ➔ Definition of the goals
    - ✓ Will yield a dominated solution if the goal point is chosen in the feasible domain
  - ➔ Not being able to deal with non-convex search spaces

Goal Attainment

- In addition to the goal vector ***b***, a vector of weights ***w*** has to be elicited from the DM

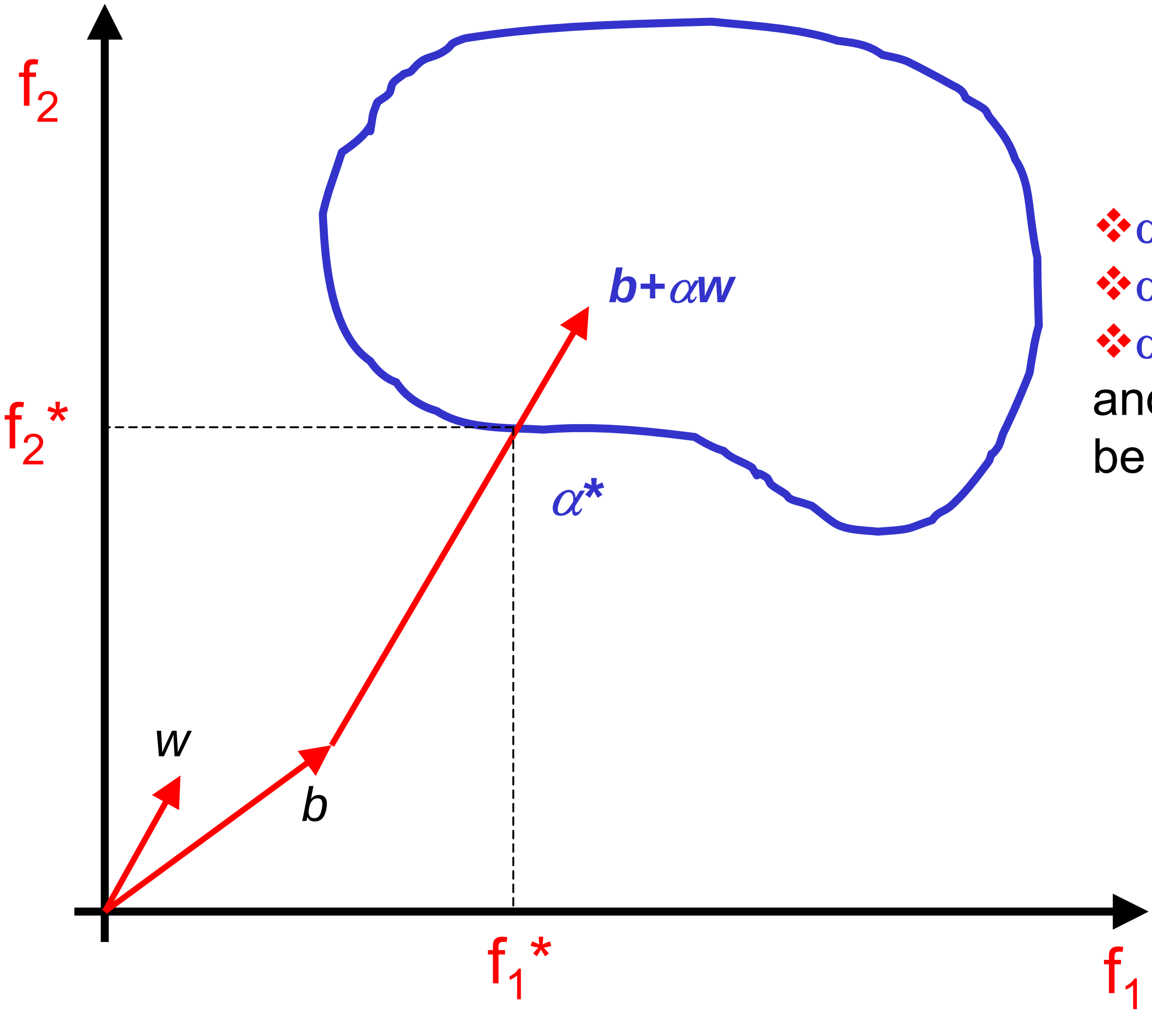
$$\min \alpha$$

subject to:

$$g_j(\bar{x}) \leq 0, \quad j = 1,2,\ldots,m$$

$$b_i + \alpha \cdot w_i \geq f_i(\bar{x}), \quad i = 1,2,\ldots,k$$

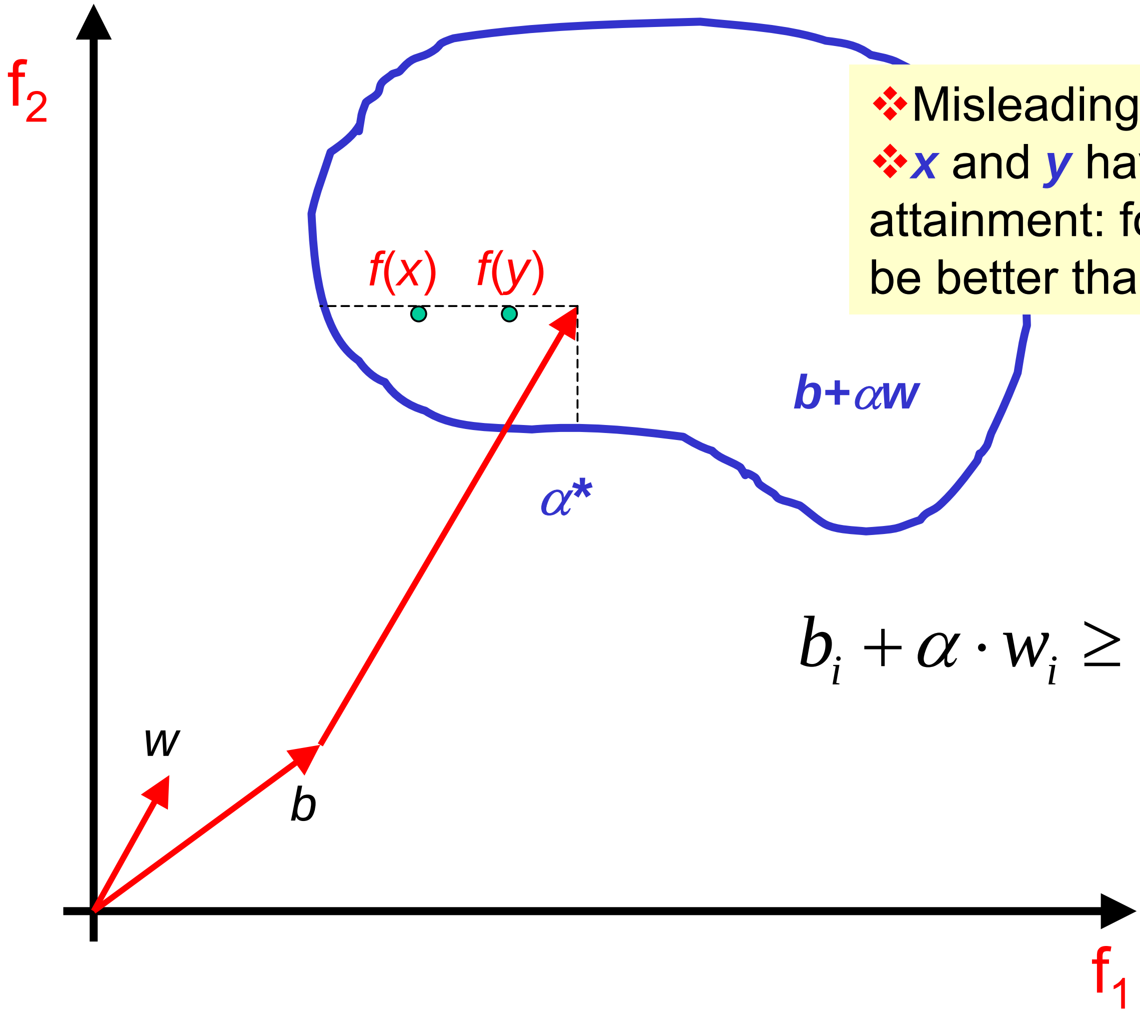
Goal Attainment (cnt'd)



- α is not limited in sign
- α **> 0** the goal is unattainable
- α **< 0** the goal is attainable

and an improved solution will be obtained

Goal Attainment (cnt'd)



- Misleading selection pressure
- ***x*** and ***y*** have the same goal-attainment: for the GA none of them will be better than the other

$$b_i + \alpha \cdot w_i \geq f_i(\bar{x}), \quad i = 1,2,\ldots,k$$

The ε-Constraint Method

- Minimization of one (primary) objective function, and considering the other objectives as constraints bound by some allowable levels ε_i

$$f_r\left(\bar{x}^*\right) = \min_{\bar{x} \in F} f_r\left(\bar{x}\right)$$

subject to additional constraints of the form

$$f_i\left(\bar{x}\right) \leq \varepsilon_i \quad i = 1,2,\ldots,k \text{ and } i \neq r$$

- Repeat with different values of ε_i

The ε-Constraint Method (cnt'd)

- Several heuristics to set the ε_i
  - ➔ Using upper and lower bound

$$\varepsilon_i \geq f_i\left(\overline{x}_i^*\right) \quad i = 1,2,\ldots,r-1,r+1,\ldots,k$$
$$\varepsilon_i \leq f_i\left(\overline{x}_r^*\right) \quad i = 1,2,\ldots,r-1,r+1,\ldots,k$$

- Problems
  - ➔ Very time-consuming approach

Non-Aggregating Approaches that are not Pareto-Based

- Born to overcome the difficulties involved in the aggregating approaches
- Based on
  - ➔ Population policies
  - ➔ Special handling of objectives

VEGA

- Vector Evaluated Genetic Algorithm
  - ➔ Extended the Grefenstette's GENESIS
- Selection
  - ➔ For a problem with *K* objectives
    - ✓ K sub-populations of size N/K would be generated
    - ✓ Sub-population will be shuffled together to obtain a new population of size N
    - ✓ Crossover and mutation operators are applied in the usual way

VEGA (cnt'd)

- "*Speciation*" problem
  - ➔ This technique selects individual who excel in one dimension (without looking at the other dimensions)
  - ➔ Loss of many compromise solutions
- Has been proved that shuffling sub-population corresponds to averaging the fitness components associated to each objective
  - ➔ Linear combination of the objectives
    - ✓ Weights depend on the distribution of the population at each generation

Lexicographic Ordering

- Objectives are ranked in order of importance
- Minimize the objective functions starting from the most important and proceeding to the assigned order of importance

Lexicographic Ordering (cnt'd)

- Let $f_1(\boldsymbol{x})$ and $f_k(\boldsymbol{x})$ denote the most and least important objective functions, respectively

$$\min f_1(\bar{x})$$

Subject to

$$g_j(\bar{x}) \leq 0 \quad j = 1,2,\ldots,m$$

And its solution $\boldsymbol{x}_1^*$ and $f_1^* = f_1(\boldsymbol{x}_1^*)$ is obtained. Then the second problem is formulated as

$$\min f_2(\bar{x})$$

Subject to

$$g_j(\bar{x}) \leq 0 \quad j = 1,2,\ldots,m$$

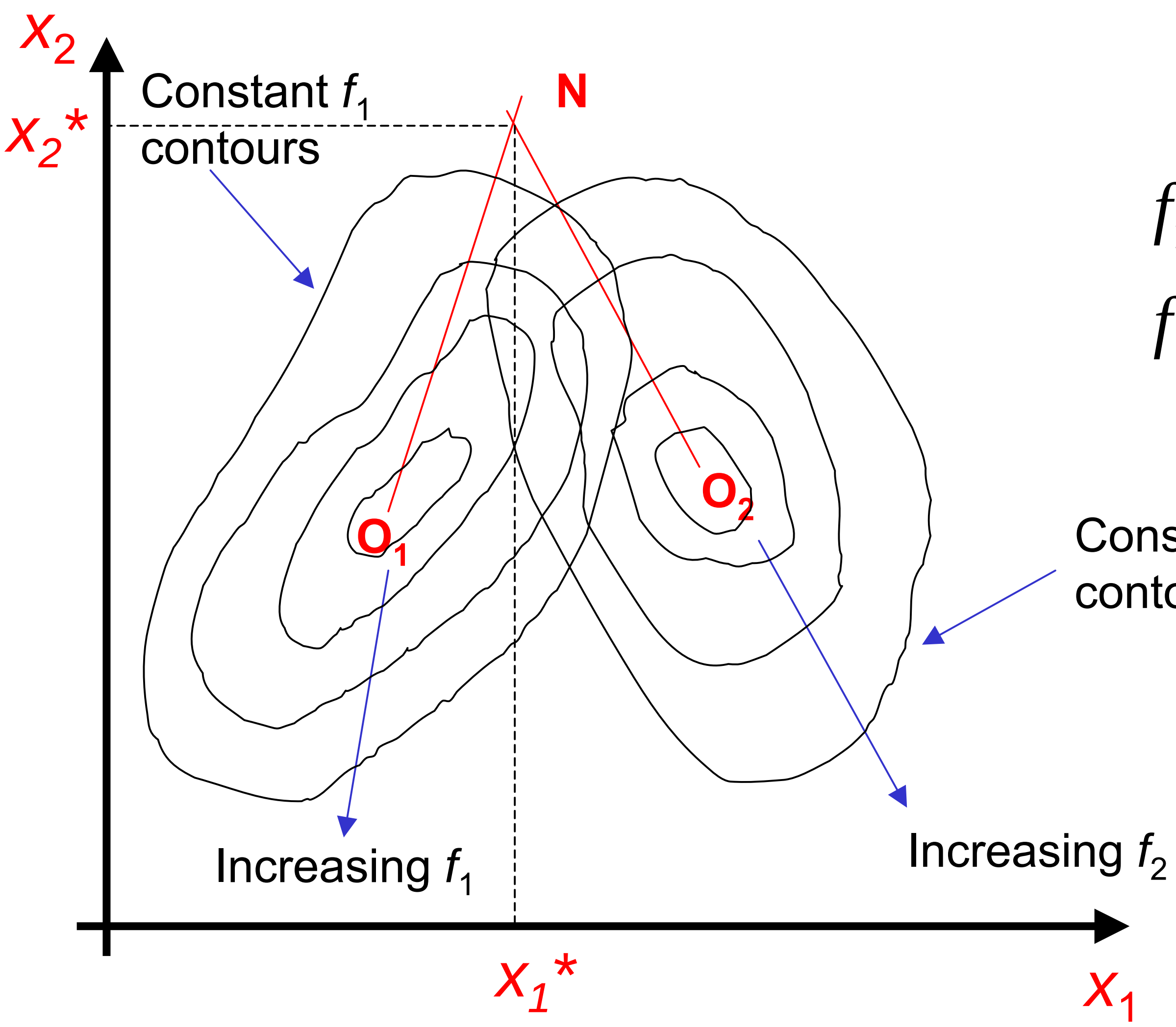
$$f_1(\bar{x}) = f_1^*$$

And the solution of the problem is obtained as $\boldsymbol{x}_2^*$ and $f_2^* = f_2(\boldsymbol{x}_2^*)$. This procedure is repeated until all k objectives have been considered

Lexicographic Ordering (cnt'd)

- Often the population converge to a particular part of Pareto front

Use of Game Theory



$$f_1\left(x_1^*, x_2^*\right) \le f_1\left(x_1, x_2^*\right)$$
$$f_2\left(x_1^*, x_2^*\right) \le f_2\left(x_1^*, x_2\right)$$

No player can deviate unilaterally from this point for further improvement his own criterion

Pareto-Based Approaches

- Calculating an individual's fitness on the basis of Pareto dominance
  - Dominance Rank: Number of individuals by which an individual is dominated
  - Dominance Depth: Population is divided in into several fronts and the depth reflects to which front an individual belongs to
  - Dominance Count: Number of individuals dominated by a certain individual
- The fitness is related to the whole population
  - In contrast to aggregation-based methods which calculate an individual's raw fitness value indipendently of other individuals

SPEA2

- **S**trength **P**areto **E**volutionary **A**lgorithm
  - P_t: Population at generation t
  - A_t: Archive (external set) at generation t
- Fitness Assignment
- Environmental Selection

SPEA2 Fitness Assignment

- Each individual *i* in A_t and P_t is assigned a strength value $S(i)$
  - ➔ $S(i)$: number of solutions it dominates

$$S(i) = \left|\left\{ j \mid j \in P_t \cup A_t \wedge i \succ j \right\}\right|$$

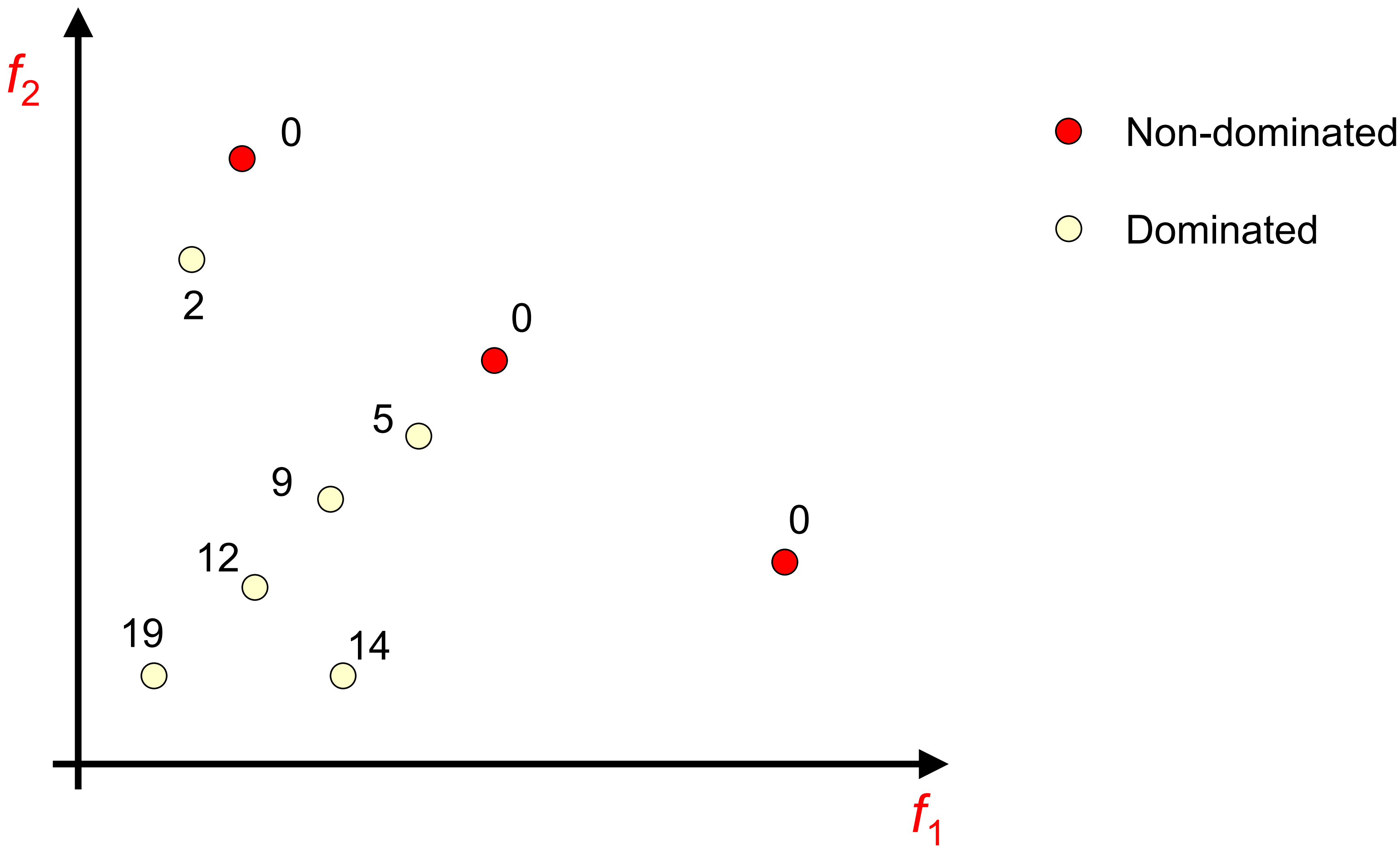
SPEA2 Fitness Assignment (cnt'd)

- On the basis of S values, the raw fitness $R(i)$ of an individual i is calculated

$$R(i) = \sum_{\substack{j \in P_t \cup A_t \\ j \succ i}} S(j)$$

- That is the raw fitness is determined by the strengths of its dominators in both archive and population

SPEA2 Fitness Assignment (cnt'd)



SPEA2 Fitness Assignment (cnt'd)

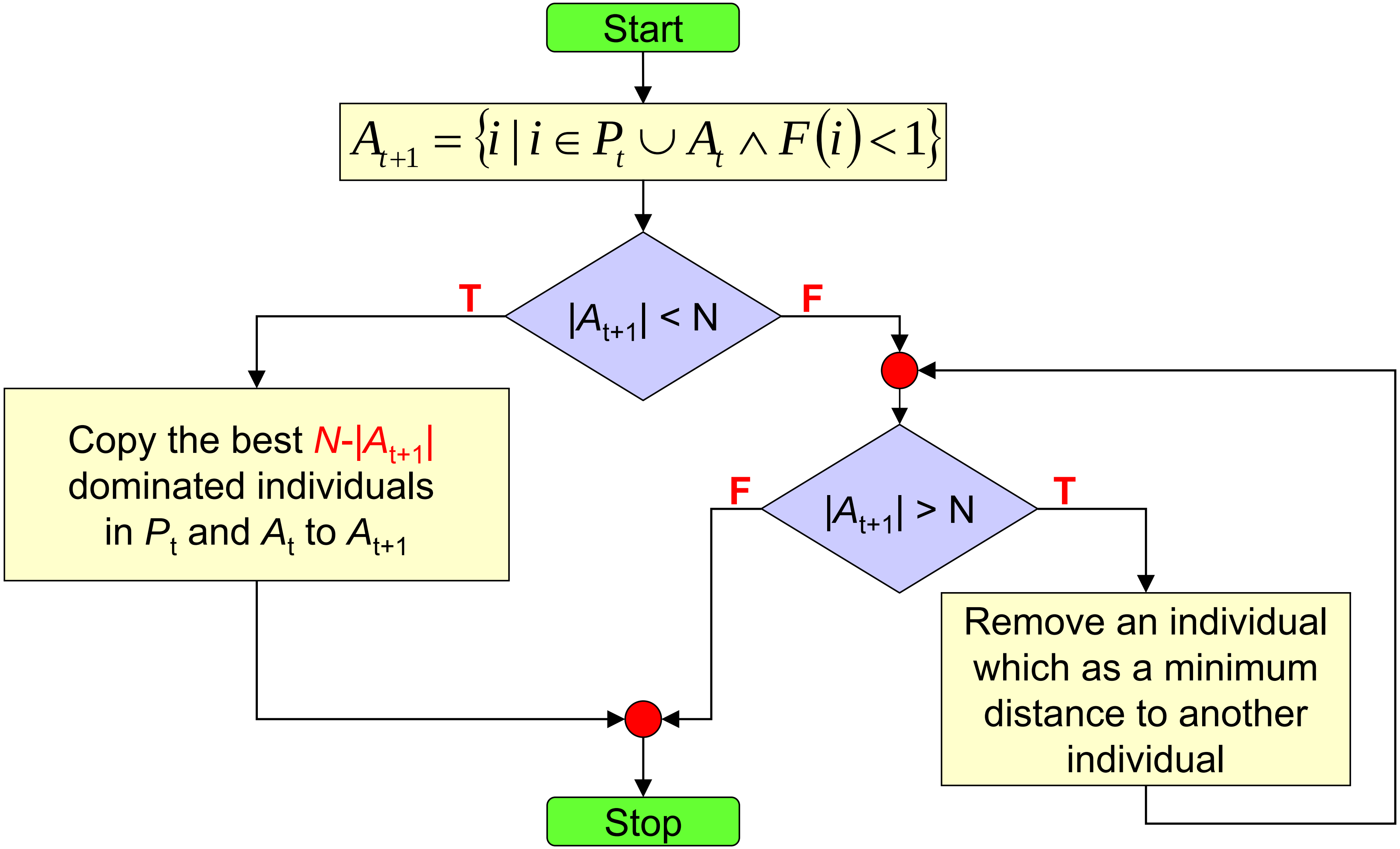
- Problems when most individuals do not dominate each other
  - ➔ Adding *density* information to discriminate between individuals having the same raw fitness
  - ➔ The density at any point is a decreasing function of the distance to the *k*-th nearest data point

$$D(i) = \frac{1}{\sigma_i^k + 2} \qquad i \in P_t \cup A_t$$

where σ_i^k denotes the distance of *i* to its *k*-th nearest neighbor

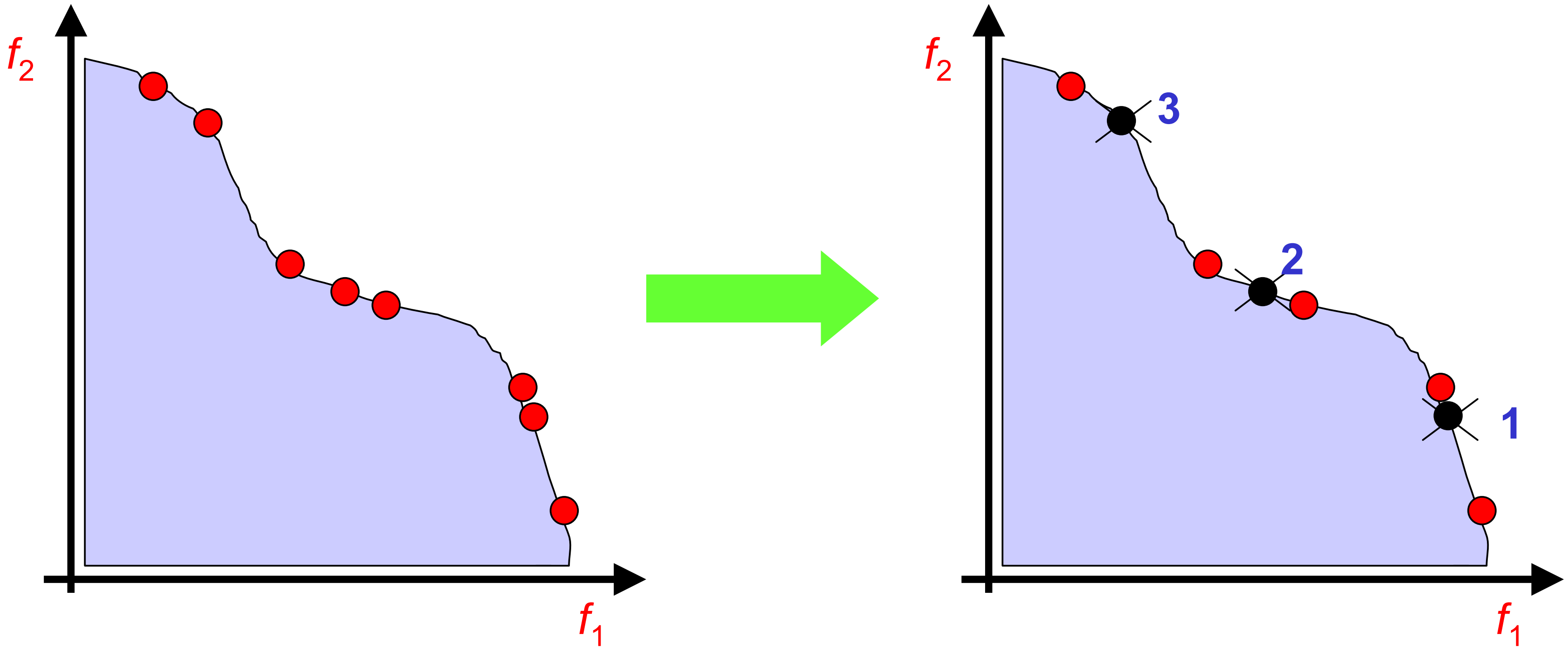
$$F(i) = R(i) + D(i) \qquad i \in P_t \cup A_t$$

SPEA2 Environmental Selection



Truncation Technique

$$i \le_d j :\Leftrightarrow \forall 0 < k < |A_{t+1}| : \sigma_i^k = \sigma_j^k \vee$$
$$\exists 0 < k < |A_{t+1}| : \left[\left(\forall 0 < l < k : \sigma_i^l = \sigma_j^l\right) \wedge \sigma_i^k = \sigma_j^k\right]$$



SPEA2 Main Loop

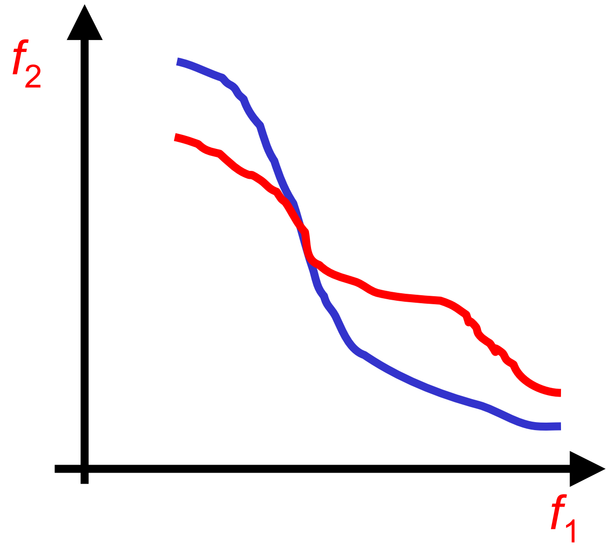
- Initialization: generate an initial population P_0 and create the empty archive $A_0=\emptyset$. Set $t=0$.
- Fitness assignment: Calculate fitness values of individuals in P_t and A_t.
- Environmental selection: Copy all nondominate individuals in P_t and A_t to A_{t+1}. If size of A_{t+1} exceeds N then reduce A_{t+1} by means of the truncation operator, otherwise if size of A_{t+1} is less then N then fill A_{t+1} with dominated individual in P_t and A_t.
- Termination: If $t \geq T$ or another stopping criterion is satisfied then set A^* to the set of decision vectors represented by the nondominated individuals in A_{t+1}. Stop.
- Mating selection: Perform binary tournament selection with replacement on A_{t+1} in order to fill the mating pool
- Variation: Apply recombination and mutation operators to the mating pool and set P_{t+1} to the resulting population. Increment generation counter ($t=t+1$) and go Step 2.

Quality Assessment

- Notion of performance
  - Computational resources needed
    - Number of fitness evaluations
    - Overall run-time
      - No difference between single- and multi-objective optimization
  - Quality of the solutions
    - Single-objective: the smaller (or larger) the value of the objective function, the better the solution
    - Multi-objective: ?

Quality Assessment (cnt'd)

- Compare 2 solutions x_1 and x_2
  - ➔ x_1 is better then x_2 if $x_1 \succ x_2$
- Compare two sets of solutions
  - ➔ Closeness to the optimal Pareto surface?
  - ➔ Coverage of a wide range of diverse solutions?
  - ➔ Other properties?

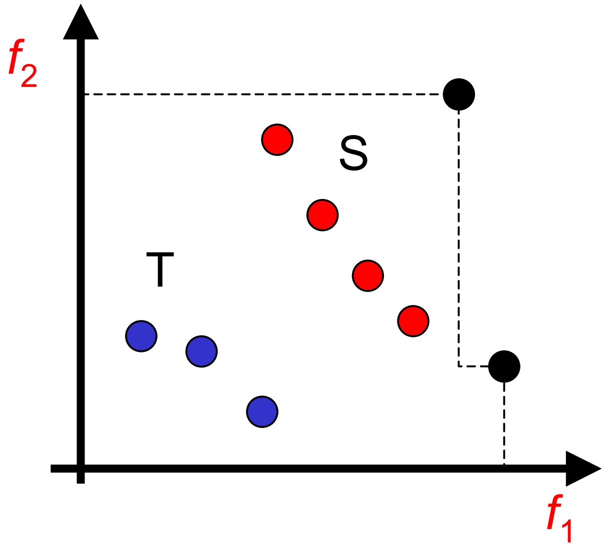


Quality Assessment (cnt'd)

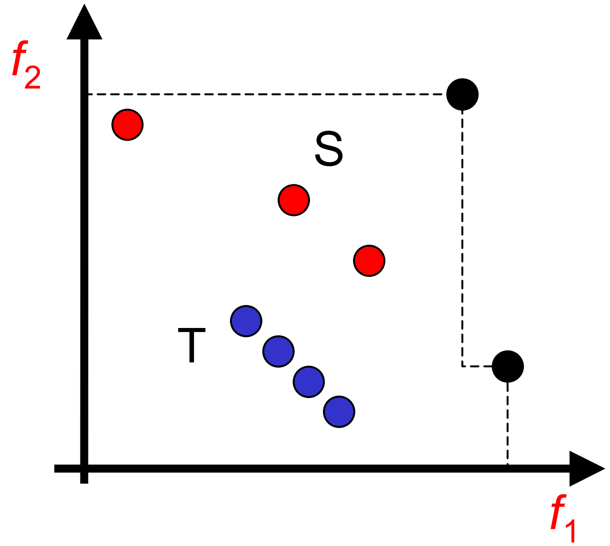
- How to best summarize Pareto set approximations by means of a few characteristic numbers
  - ➔ Crucial point: does not lose the information one is interested in
  - ➔ Some quality measures
    - ✓ Average distance from Pareto-optimal front
    - ✓ Hypervolume measure
    - ✓ Diversity
    - ✓ Spread
    - ✓ Cardinality

Quality Assessment (cnt'd)

- Quality of a Pareto set approximation cannot be completely described by a (finite) set of distinct criteria
  - ➔ Average distance, Diversity, Cardinality



S $\succ$ **T**
S is better then **T**



S $\succ$ **T**
S is worst then **T**

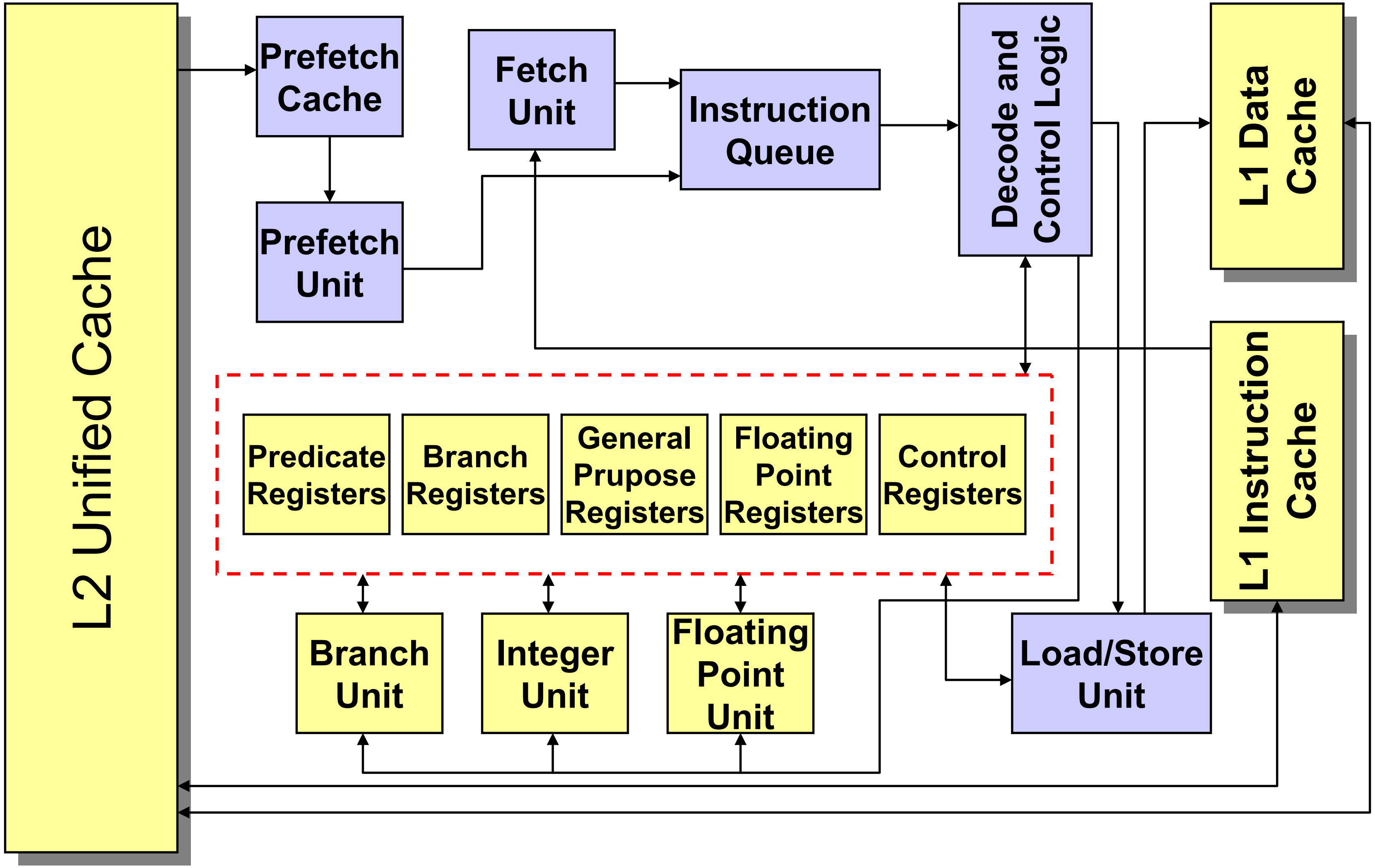
Binary ε-Quality Measure

- Let *S* and *T* two Pareto-set
- Binary ε-quality measure is the minimum $\varepsilon \in \Re$ such that any $b \in T$ is ε-dominated by at least one $a \in S$

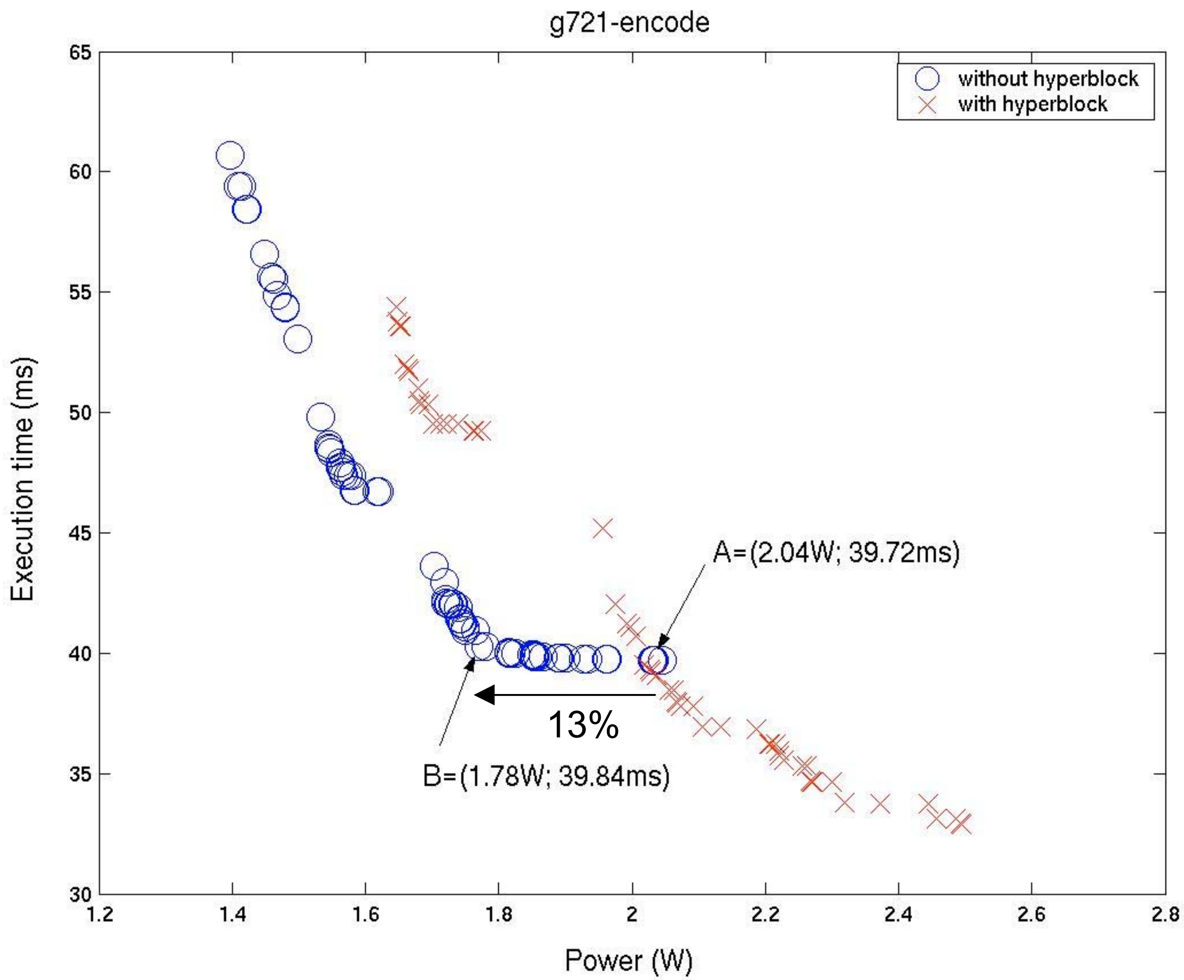
$$I_\varepsilon(S,T) = \min\{\varepsilon \in \Re \mid \forall b \in T\, \exists a \in S : \varepsilon \cdot a \succ b\}$$

- $I_\varepsilon(S,T)<1$: All solutions in *T* are dominated by a solution in *S*
- $I_\varepsilon(S,T)=1 \wedge I_\varepsilon(T,S)=1$: *T* and *S* represent the same Pareto front approximation
- $I_\varepsilon(S,T)>1 \wedge I_\varepsilon(S,T)>1$: *T* and *S* are incomparable

Reference architecture (HPL-PD)



Pareto Set (G721 encode)



Resources

- Programmin libraries
  - GALibs (http://lancet.mit.edu/ga/)
  - MOMHLib++ (http://www-idss.cs.put.poznan.pl/~jaszkiewicz/MOMHLib/)
  - PISA (http://www.tik.ee.ethz.ch/pisa/)
- References
  - EMOO (http://www.lania.mx/~ccoello/EMOO/)