Lecture 9: Multi-Objective Optimization

Suggested reading: K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Inc., 2001

Multi-Objective Optimization Problems (MOOP)

- Involve more than one objective function that are to be minimized or maximized
- Answer is set of solutions that define the best tradeoff between competing objectives

General Form of MOOP

- Mathematically

$$
\begin{aligned}
\text{min/max} \quad & f_m(\boldsymbol{x}), && m = 1, 2, \cdots, M \\
\text{subject to} \quad & g_j(\boldsymbol{x}) \geq 0, && j = 1, 2, \cdots, J \\
& h_k(\boldsymbol{x}) = 0, && k = 1, 2, \cdots, K \\
& \underbrace{x_i^{(L)}}_{\text{lower bound}} \leq x_i \leq \underbrace{x_i^{(U)}}_{\text{upper bound}}, && i = 1, 2, \cdots, n
\end{aligned}
$$

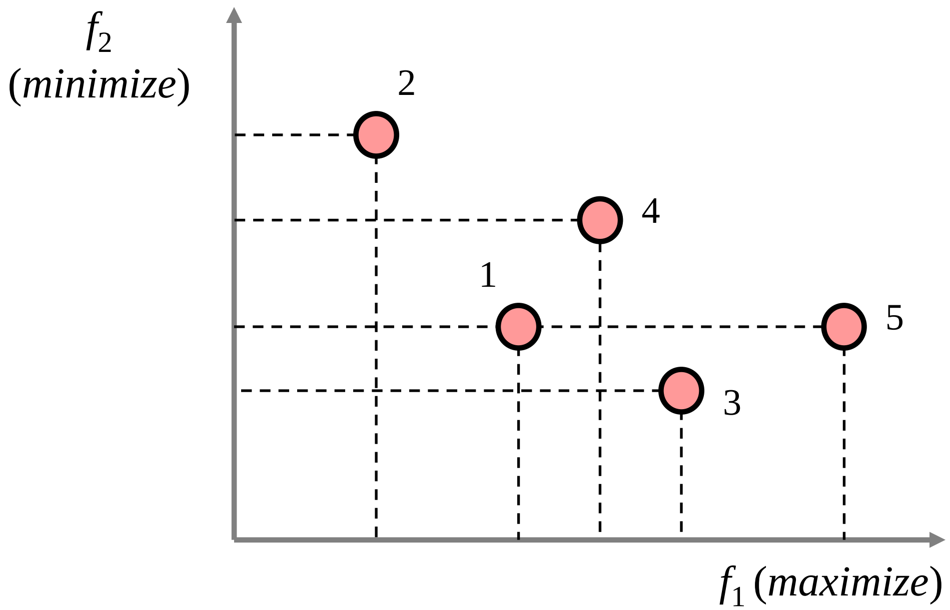
Dominance

- In the single-objective optimization problem, the superiority of a solution over other solutions is easily determined by comparing their objective function values
- In multi-objective optimization problem, the goodness of a solution is determined by the **dominance**

Definition of Dominance

- **Dominance Test**
  - $\boldsymbol{x}_1$ dominates $\boldsymbol{x}_2$, if
    - Solution $\boldsymbol{x}_1$ is no worse than $\boldsymbol{x}_2$ in all objectives
    - Solution $\boldsymbol{x}_1$ is strictly better than $\boldsymbol{x}_2$ in at least one objective
  - $\boldsymbol{x}_1$ dominates $\boldsymbol{x}_2$ $\Longleftrightarrow$ $\boldsymbol{x}_2$ is dominated by $\boldsymbol{x}_1$

Example Dominance Test

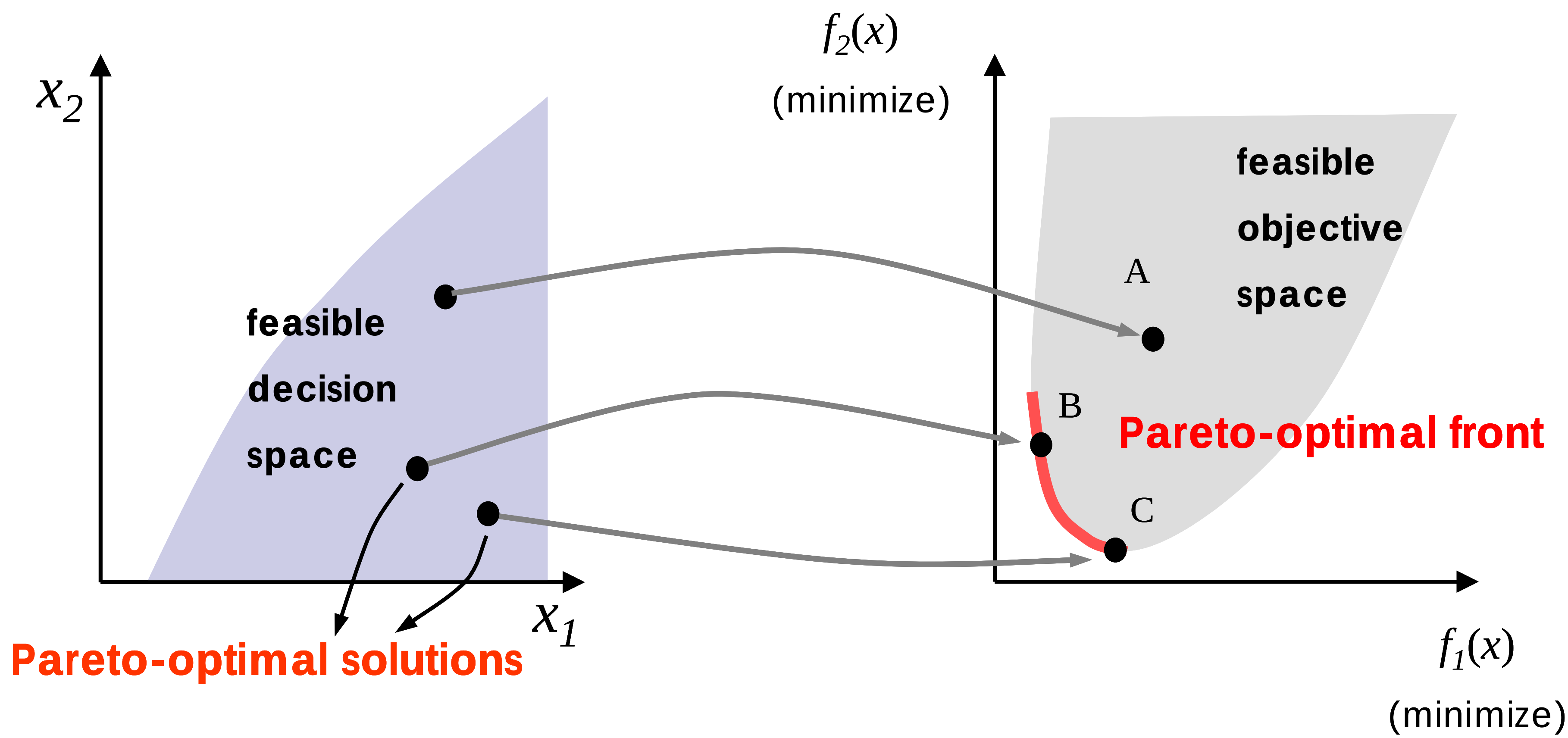


- 1 Vs 2: 1 dominates 2
- 1 Vs 5: 5 dominates 1
- 1 Vs 4: Neither solution dominates

Pareto Optimal Solution

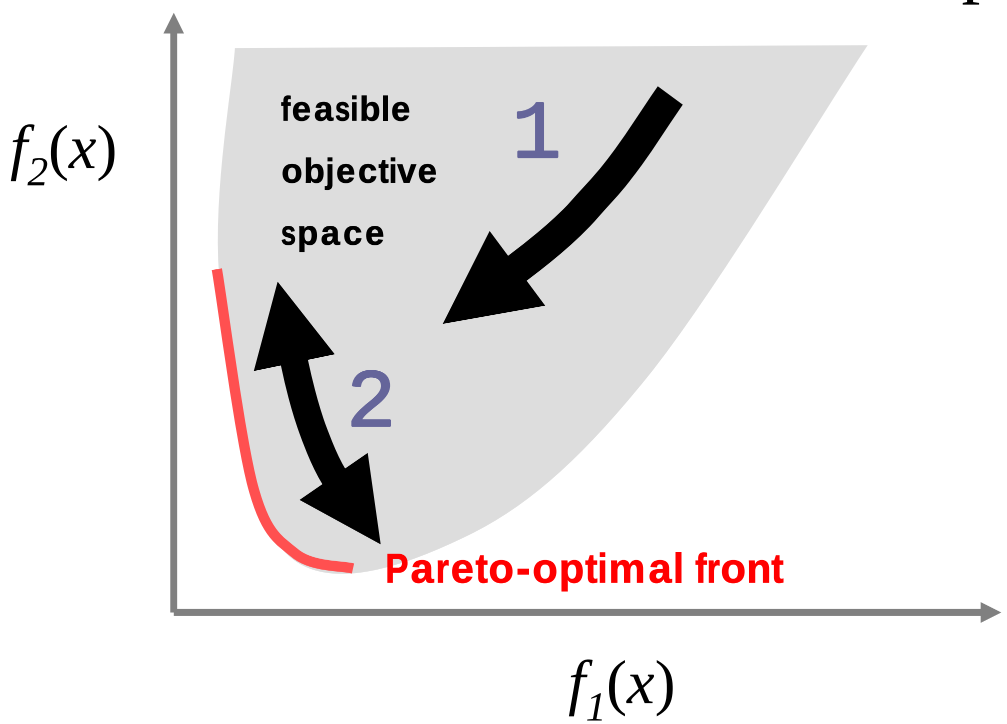
- **Non-dominated solution set**
  - Given a set of solutions, the non-dominated solution set is a set of all the solutions that are not dominated by any member of the solution set
- The non-dominated set of the entire feasible decision space is called the **Pareto-optimal set**
- The boundary defined by the set of all point mapped from the Pareto optimal set is called the **Pareto-optimal front**

Graphical Depiction of Pareto Optimal Solution



Goals in MOO

- Find set of solutions as close as possible to Pareto-optimal front
- To find a set of solutions as diverse as possible



Classic MOO Methods

Weighted Sum Method

- Scalarize a set of objectives into a single objective by adding each objective pre-multiplied by a user-supplied weight

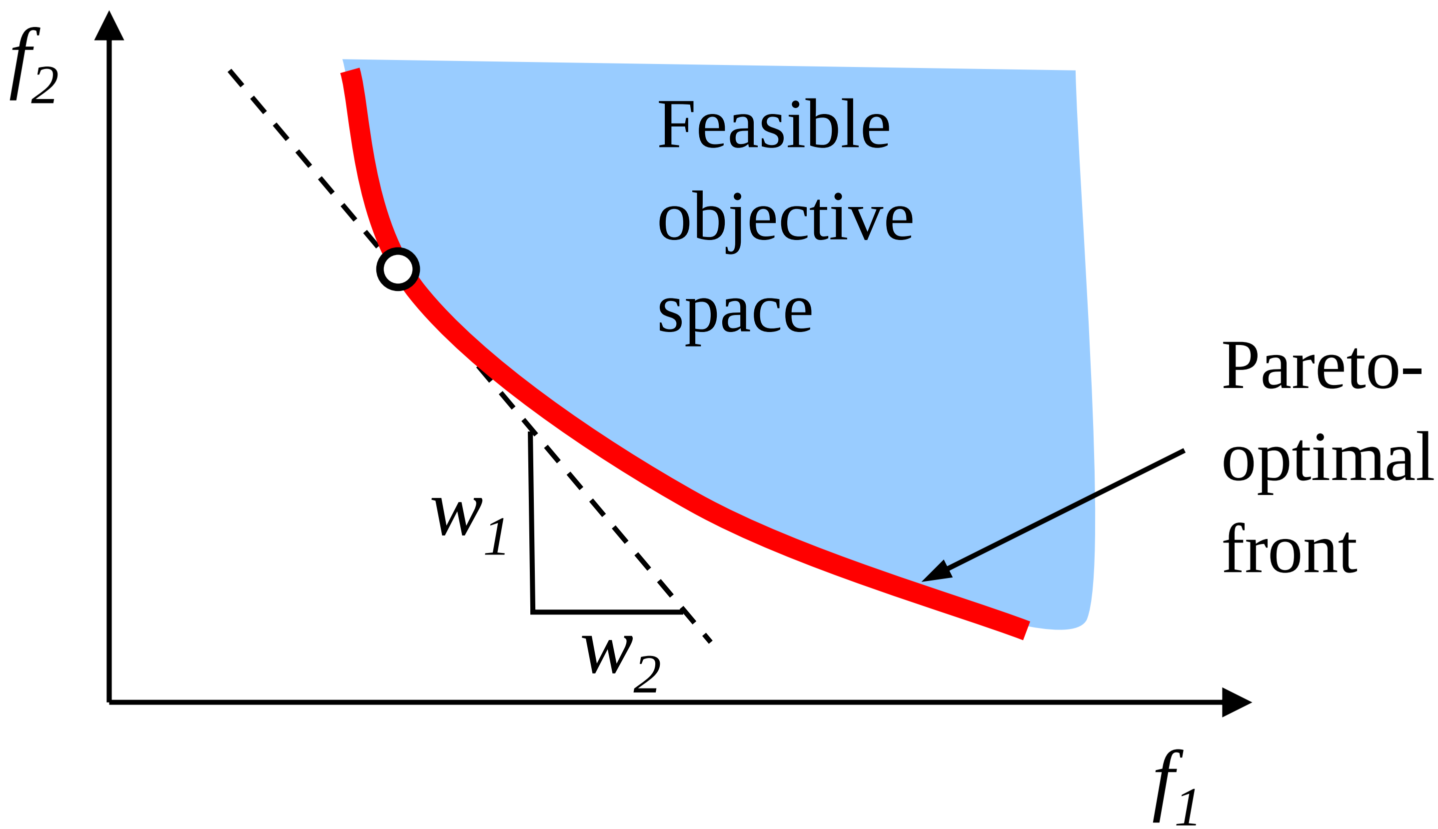
$$
\begin{aligned}
\textbf{minimize} \quad & F(\boldsymbol{x}) = \sum\nolimits_{m=1}^{M} w_m f_m(\boldsymbol{x}), & \\
\textbf{subject to} \quad & g_j(\boldsymbol{x}) \geq 0, & j = 1, 2, \cdots, J \\
& h_k(\boldsymbol{x}) = 0, & k = 1, 2, \cdots, K \\
& x_i^{(L)} \leq x_i \leq x_i^{(U)}, & i = 1, 2, \cdots, n
\end{aligned}
$$

- Weight of an objective is chosen in proportion to the relative importance of the objective

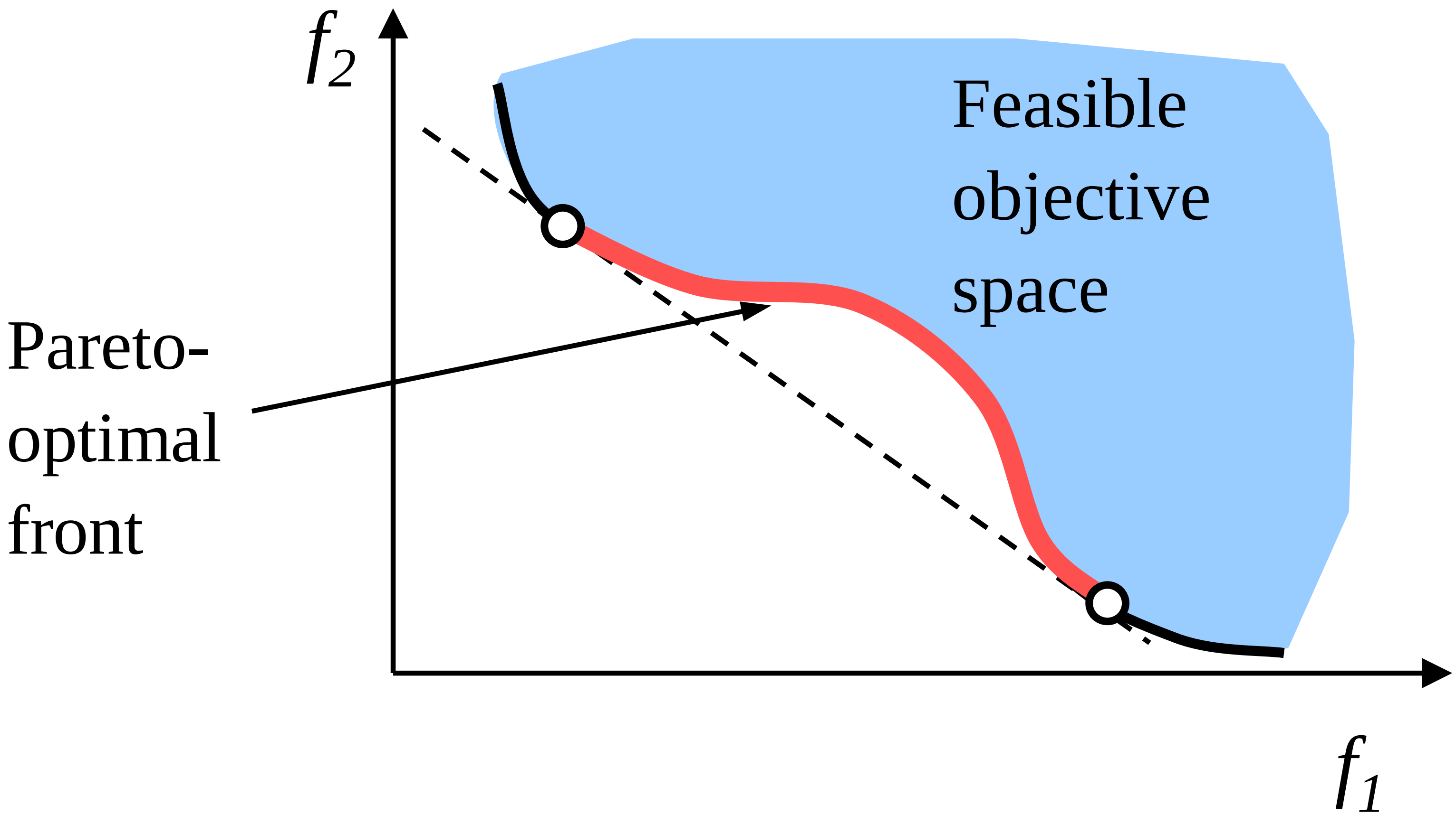
Weighted Sum Method

- Advantage
  - Simple
- Disadvantage
  - It is difficult to set the weight vectors to obtain a Pareto-optimal solution in a desired region in the objective space
  - It cannot find certain Pareto-optimal solutions in the case of a nonconvex objective space

Weighted Sum Method (Convex Case)



Weighted Sum Method (Non-Convex Case)



ε-Constraint Method

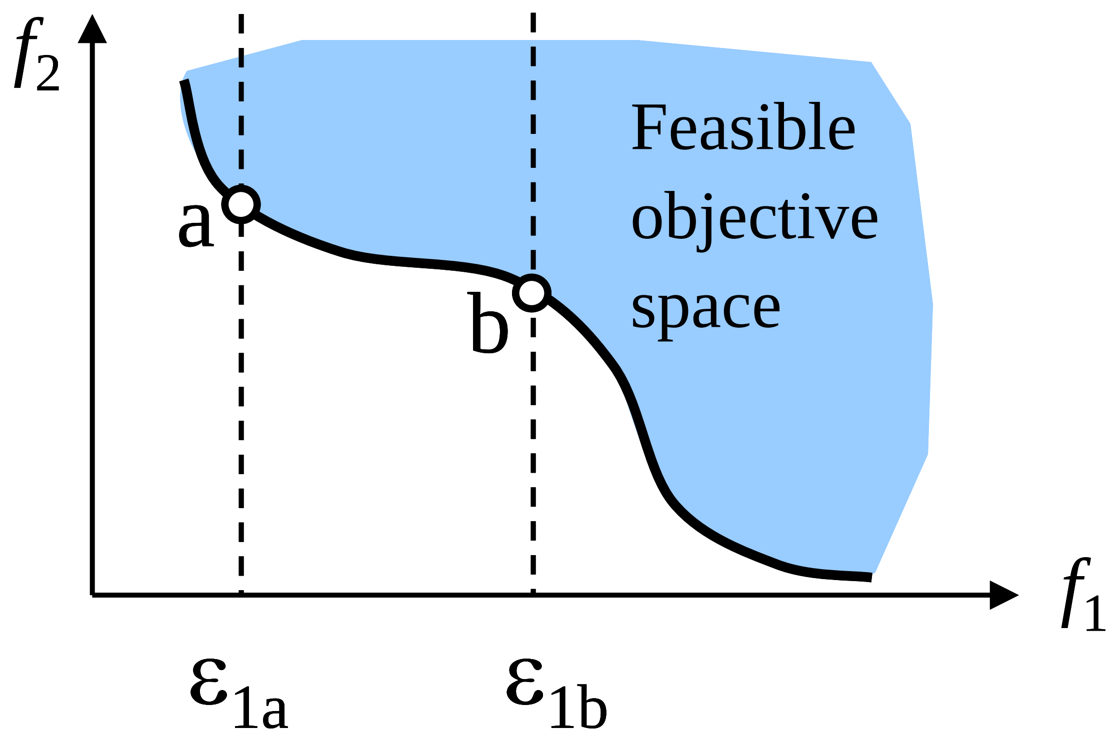
- Haimes et. al. 1971
- Keep just one of the objective and restricting the rest of the objectives within user-specific values

$$
\begin{aligned}
\textbf{minimize} \quad & f_\mu(\boldsymbol{x}), & \\
\textbf{subject to} \quad & f_m(\boldsymbol{x}) \le \varepsilon_m, & m = 1, 2, \cdots, M \text{ and } m \ne \mu \\
& g_j(\boldsymbol{x}) \ge 0, & j = 1, 2, \cdots, J \\
& h_k(\boldsymbol{x}) = 0, & k = 1, 2, \cdots, K \\
& x_i^{(L)} \le x_i \le x_i^{(U)}, & i = 1, 2, \cdots, n
\end{aligned}
$$

ε-Constraint Method

Keep f_2 as an objective **Minimize** $f_2(\boldsymbol{x})$

Treat f_1 as a constraint $f_1(\boldsymbol{x}) \leq \varepsilon_1$



ε-Constraint Method

- Advantage
  - Applicable to either convex or non-convex problems

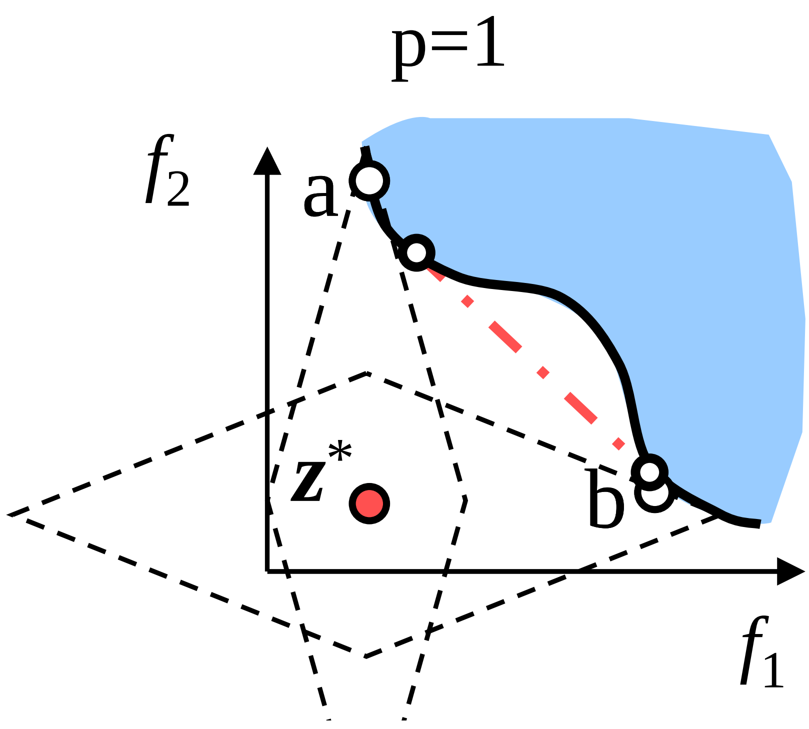
- Disadvantage
  - The ε vector has to be chosen carefully so that it is within the minimum or maximum values of the individual objective function

Weighted Metric Method

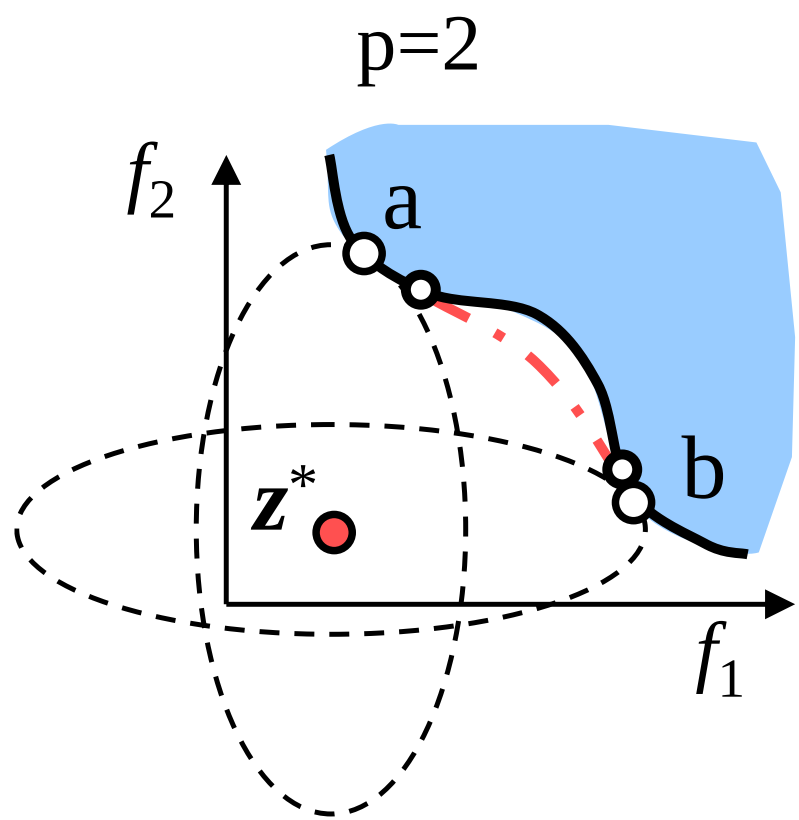
- Combine multiple objectives using the weighted distance metric of any solution from the ideal solution z^*

$$
\begin{array}{lll}
\textbf{minimize} & l_{\mathbf{p}}(\boldsymbol{x}) = \left(\sum_{m=1}^{M} w_m \left| f_m(\boldsymbol{x}) - z_m^* \right|^p \right)^{1/p}, & \\
\textbf{subject to} & g_j(\boldsymbol{x}) \geq 0, & j = 1, 2, \cdots, J \\
 & h_k(\boldsymbol{x}) = 0, & k = 1, 2, \cdots, K \\
 & x_i^{(L)} \leq x_i \leq x_i^{(U)}, & i = 1, 2, \cdots, n
\end{array}
$$

Weighted Metric Method

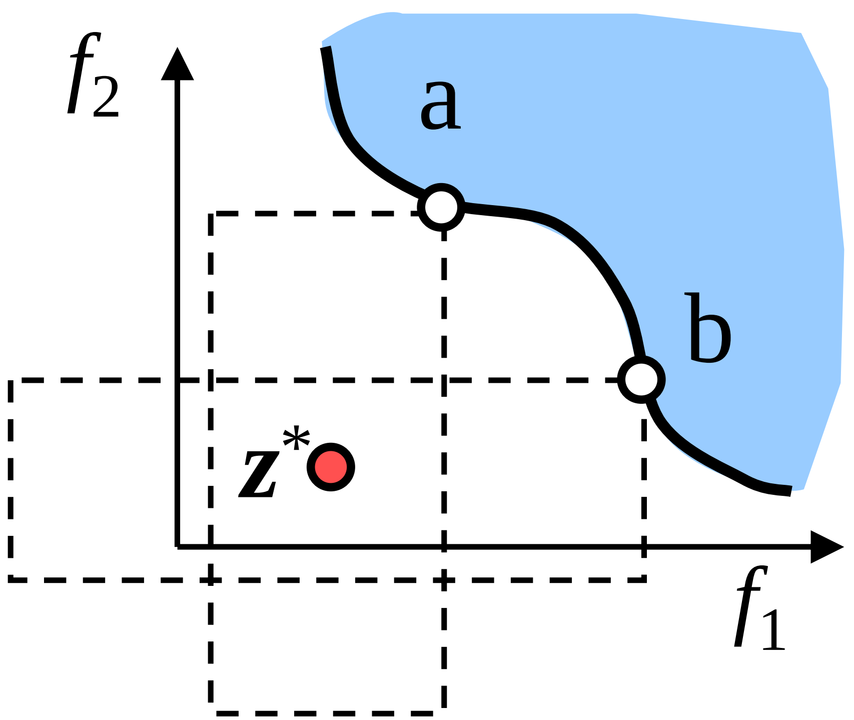


(Weighted sum approach)



Weighted Metric Method

p=∞



(Weighted Tchebycheff problem)

Weighted Metric Method

- Advantage
  - Weighted Tchebycheff metric guarantees finding all Pareto-optimal solution with ideal solution z^*
- Disadvantage
  - Requires knowledge of minimum and maximum objective values
  - Requires z^* which can be found by independently optimizing each objective functions
  - For small p, not all Pareto-optimal solutions are obtained
  - As p increases, the problem becomes non-differentiable

Multi-Objective Genetic Algorithms

Advantages of GAs over Traditional Methods

- Our desired answer: a set of solutions
- Traditional optimization methods operate on **a** candidate solution
- Genetic algorithms fundamentally operate on a set of candidate solutions

Multi-Objective EAs (MOEAs)

- There are several different multi-objective evolutionary algorithms
- Depending on the usage of elitism, there are two types of multi-objective EAs

Multi-Objective MOEAs

Non-Elitist MOEAs

- Vector evaluated GA
- Vector optimized ES
- Weight based GA
- Random weighted GA
- Multiple-objective GA
- Non-dominated Sorting GA
- Niched Pareto GA

Elitist MOEAs

- Elitist Non-dominated Sorting GA
- Distance-based Pareto GA
- Strength Pareto GA
- Pareto-archived ES

Identifying the Non-Dominated Set

- Critical Step in Elitist Strategies
- Kung's et. al. Method
  - Computational the most efficient method known
  - Recursive

Kung's et. al. Method: Step 1

- Step 1: Sort population in descending order of importance of the first objective function and name population as P
- Step 2: Call recursive function Front(P)

Front(P)

IF |P| = 1,

  Return P

ELSE

  T = **Front** (P(1: [|P|/2]))

  B = **Front** (P([|P|/2 + 1] : |P|))

  IF the *i*-th non-dominated solution of B is not dominated by any non-dominated solution of T, M=T $\cup$ {*i*}

    Return M

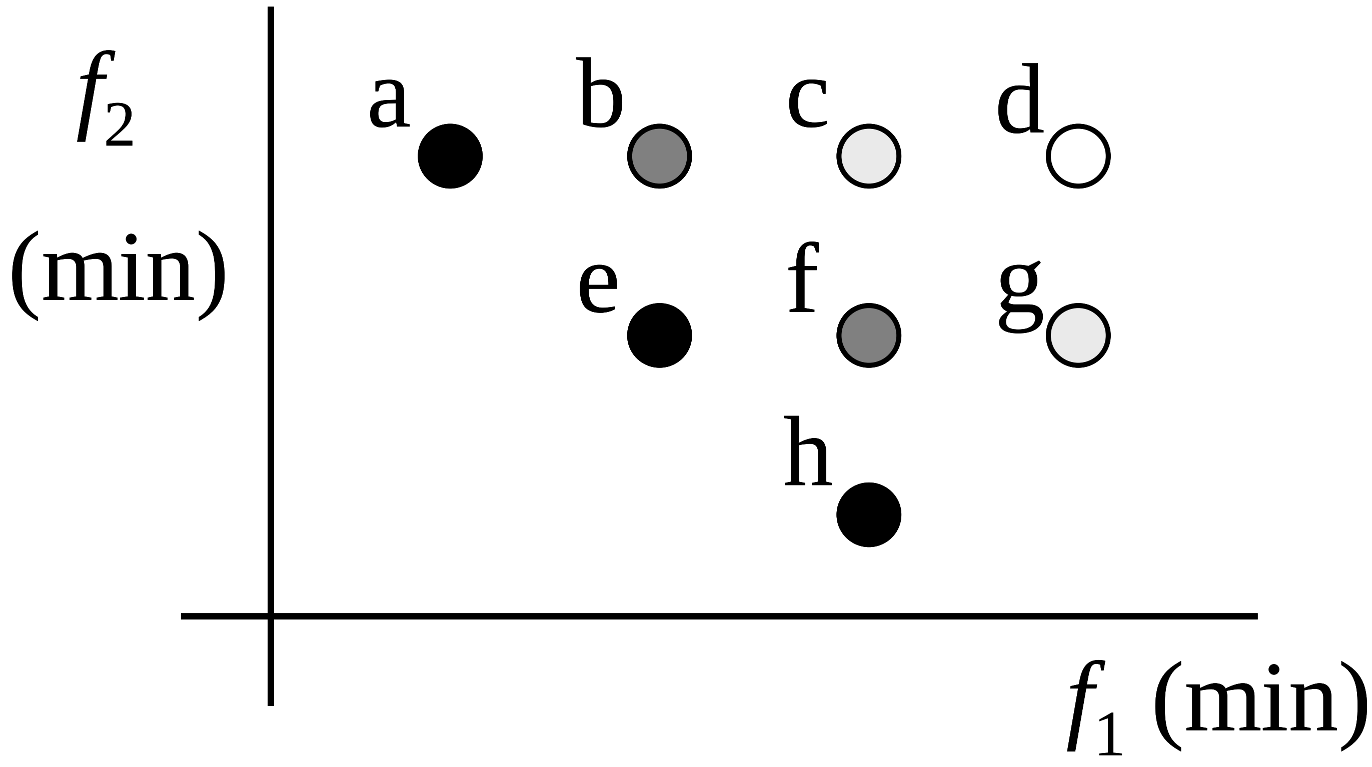
  END

END

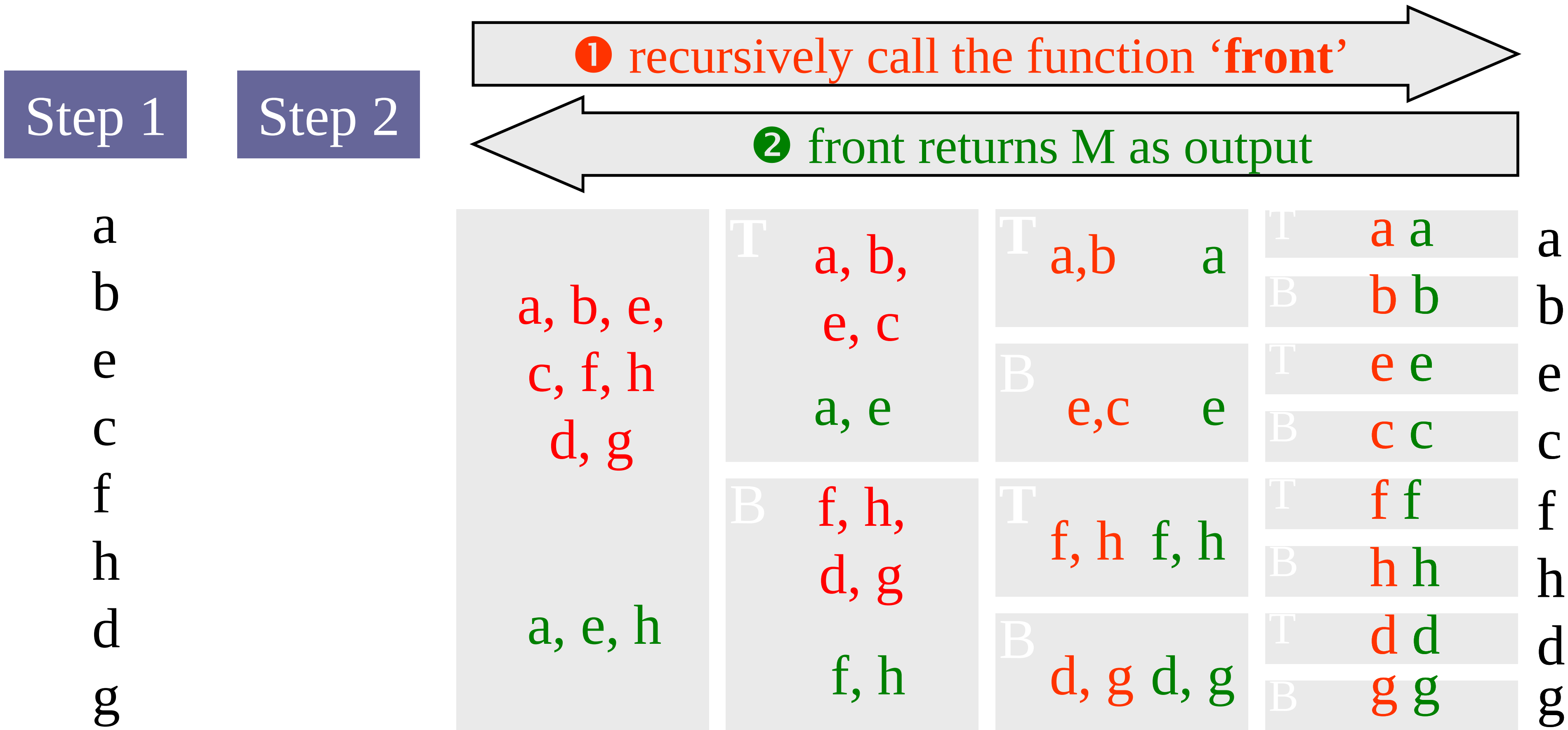
Notes on Front (P)

- |•| is the number of the elements
- P(a : b) means all the elements of P from index a to b,
- [•] is an operator gives the nearest smaller integer value

Example of Kung's Method



Example of Kung's Method



Elitist MOEAs

- Elite-preserving operator carries elites of a population to the next generation
- Rudolph(1996) proved GAs converge to the global optimal solution of some functions in the presence of elitism
- Elitist MOEAs two methods are often used
  - Elitist Non-dominated Sorting GA (NSGA II)
  - Strength Pareto EA

* **Reference**: G. Rudolph, *Convergence of evolutionary algorithms in general search spaces*, In Proceedings of the Third IEEE conference of Evolutionary Computation, 1996, p.50-54

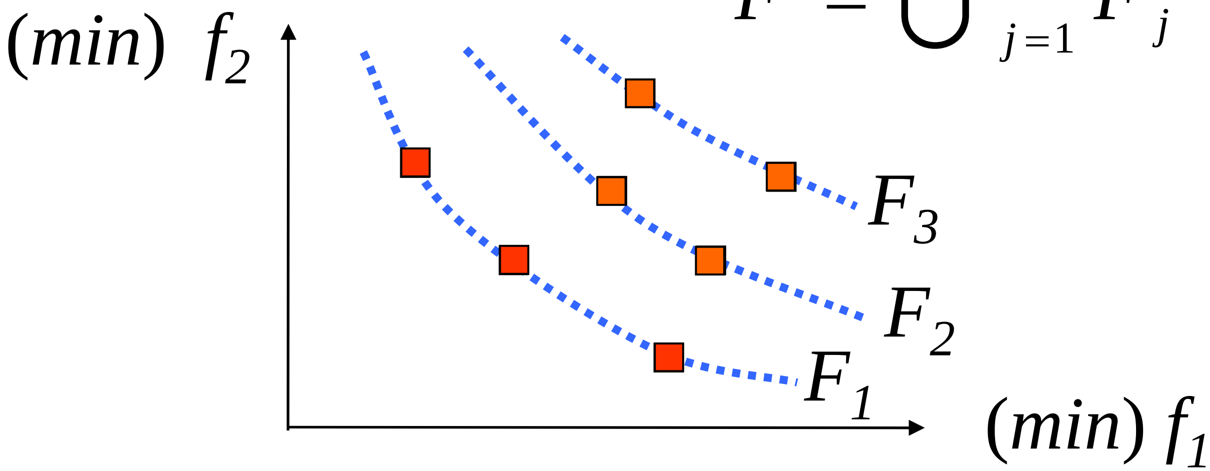
Elitist Non-Dominated Sorting GA
(Deb et al., 2000)

- Use an explicit diversity-preserving strategy together with an elite-preservation strategy

Elitist Non-Dominated Sorting GA

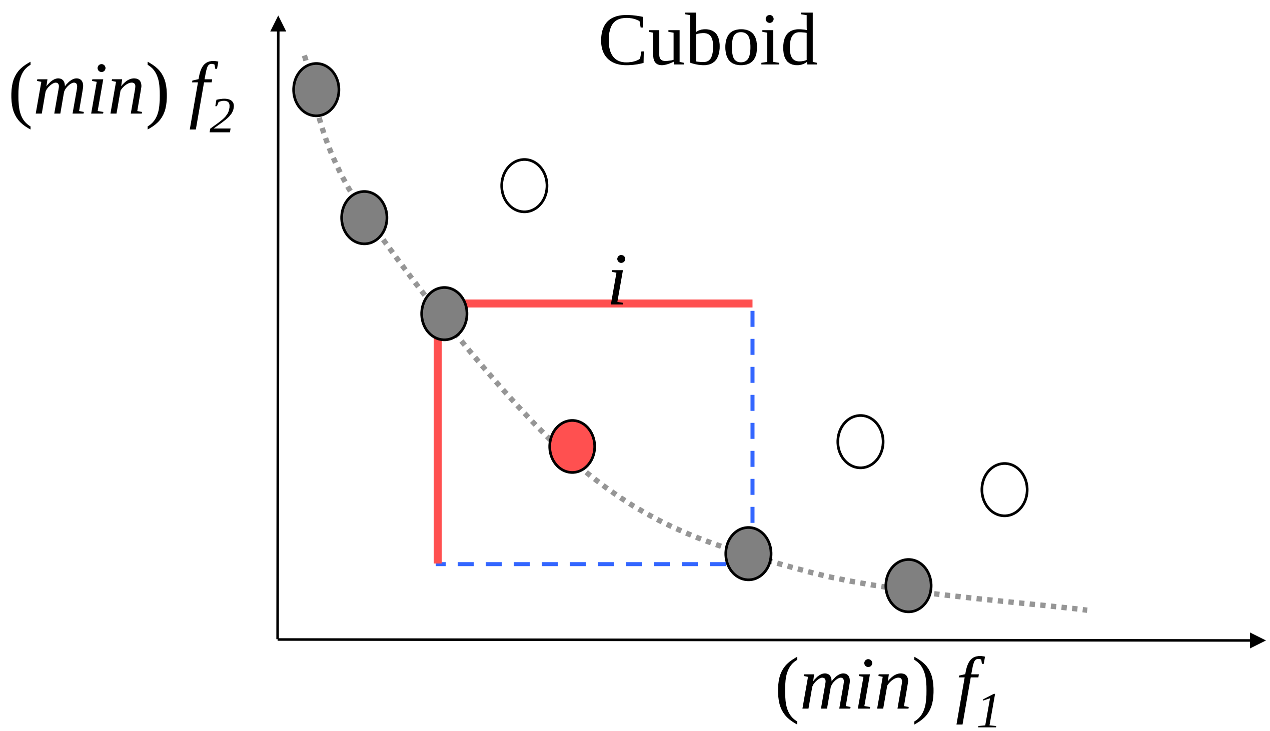
- Non-Dominated Sorting
  - Classify the solutions into a number of mutually exclusive equivalent non-dominated sets

$$F = \bigcup_{j=1}^{\rho} F_j$$



Elitist Non-Dominated Sorting GA

- Determine Crowding Distance
  - Denotes half of the perimeter of the enclosing cuboid with the nearest neighboring solutions in the same front



Elitist Non-Dominated Sorting GA

- Crowding tournament selection
  - Assume that every solution has a non-domination rank and a local crowding distance
  - A solution *i* wins a tournament with another solution *j*
    1. if the solution *i* has a better rank
    2. They have the same rank but solution *i* has a better crowing distance than solution *j*

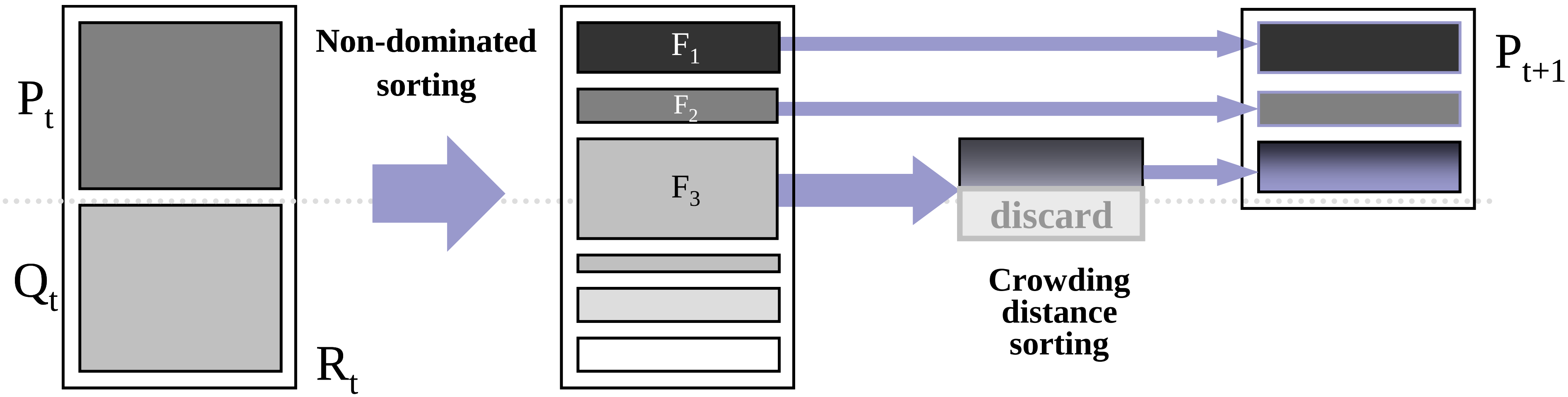
Elitist Non-Dominated Sorting GA

- **Step 1**
  - Combine parent P_t and offspring Q_t populations $R_t = P_t \cup Q_t$
  - Perform a **non-dominated sorting** to R_t and find different fronts F_i
- **Step 2**
  - Set new population $P_{t+1} = \varnothing$ and set $i = 1$
  - Until $|P_{t+1}| + |F_i| < N$, perform $P_{t+1} = P_{t+1} \cup F_i$ and increase i

Elitist Non-dominated Sorting GA

- **Step 3**
  - Include the most widely spread solutions ($N-|P_{t+1}|$) of F_i in P_{t+1} using the **crowding distance** values
- **Step 4**
  - Create offspring population Q_{t+1} from P_{t+1} by using the **crowded tournament selection**, crossover and mutation operators

Elitist Non-Dominated Sorting GA

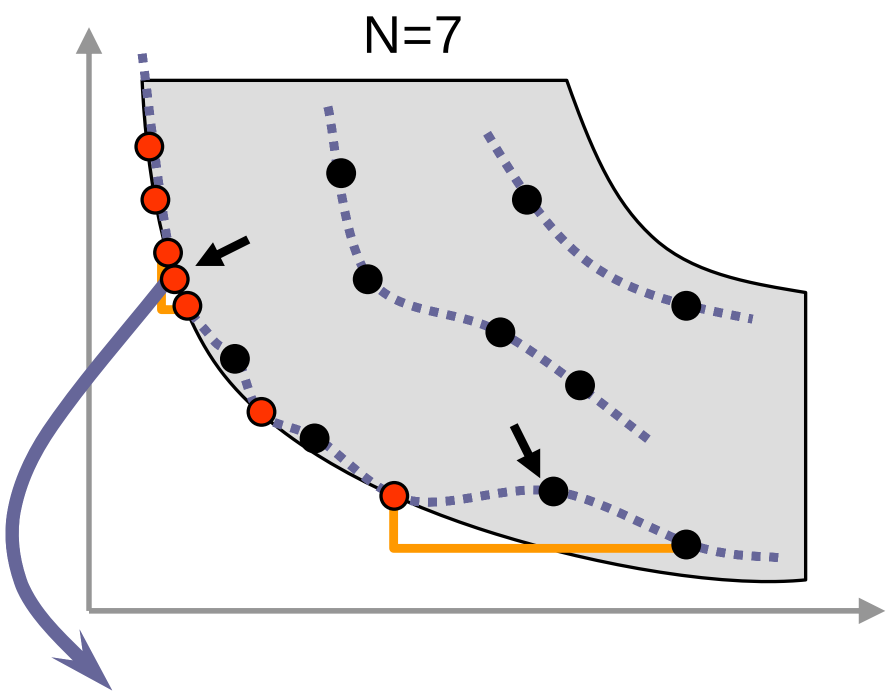


Elitist Non-dominated Sorting GA

- Advantages
  - The diversity among non-dominated solutions is maintained using the crowding procedure: No extra diversity control is needed
  - Elitism protects an already found Pareto-optimal solution from being deleted

Elitist Non-dominated Sorting GA

- Disadvantage
  - When there are more than N members in the first non-dominated set, some Pareto-optimal solutions may give their places to other non-Pareto-optimal solutions



A Pareto-optimal solution is discarded

Strength Pareto EA (SPEA)

- Zitzler & Thiele., 1998
- Use external population P
  - Store fixed number of non-dominated solutions
  - Newly found non-dominated solutions are compared with the existing external population and the resulting non-dominated solutions are preserved

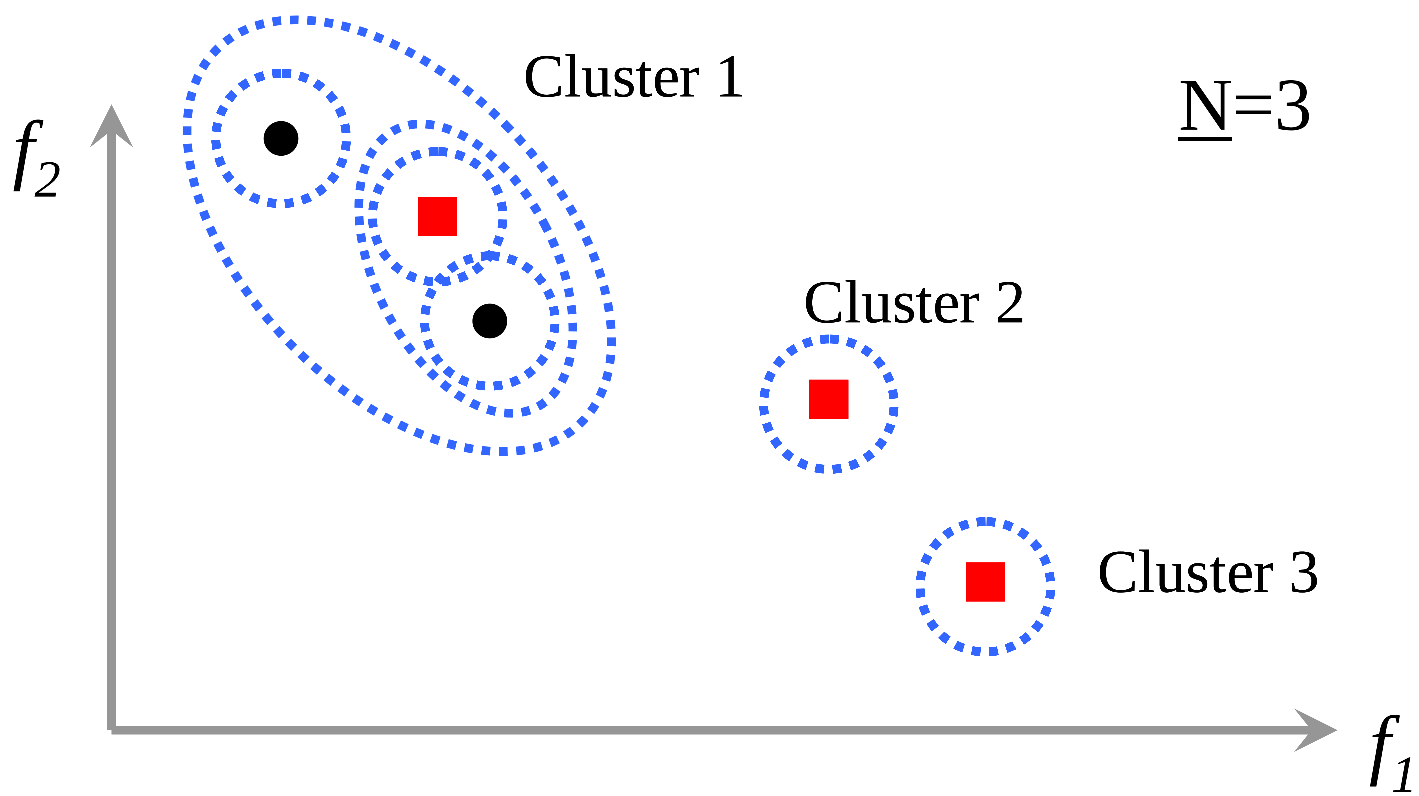
SPEA Clustering Algorithm

1. Initially, each solution belongs to a distinct cluster C_i
2. If number of clusters is less than or equal to N, go to 5
3. For each pair of clusters, calculate the cluster distance d_{ij} and find the pair with minimum cluster-distance

$$d_{12} = \frac{1}{|C_1||C_2|} \sum_{i \in C_1, j \in C_2} d(i, j)$$

4. Merge the two clusters and go to 2
5. Choose only one solution from each cluster and remove the other (The solution having minimum average distance from other solutions in the cluster can be chosen)

SPEA Clustering Algorithm



SPEA Algorithm

- Step 1. Create initial population P_0 of size N randomly and an empty external population $\underline{P}_0$ with maximum capacity of $\underline{N}$
- Step 2. Find the non-dominated solutions of P_t and copy (add) these to $\underline{P}_t$
- Step 3. Find the non-dominated solutions of $\underline{P}_t$ and delete all dominated solutions
- Step 4. If $|\underline{P}_t| > \underline{N}$ then use the **clustering technique** to reduce the size to $\underline{N}$

- Step 5 Fitness evaluation
  - Elites: assign fitness to each elite solution *i* by using

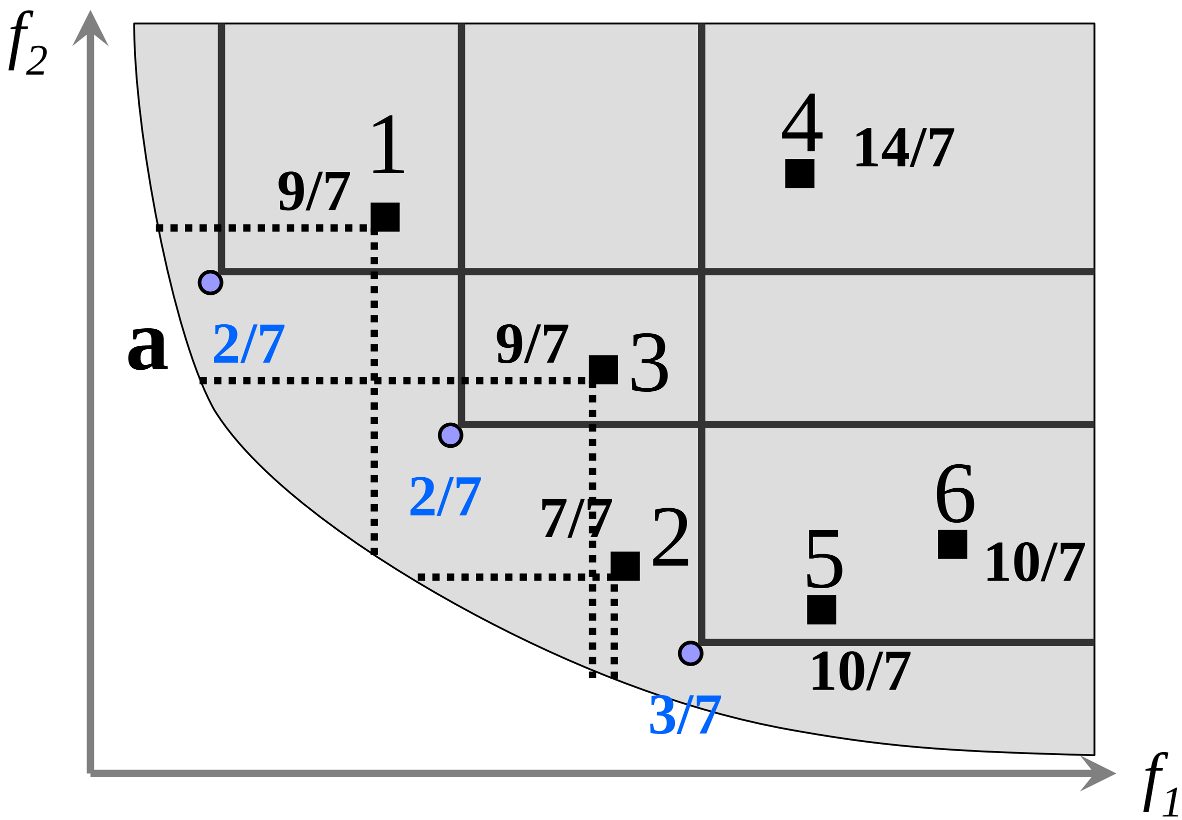
$$S_i = \frac{\#\text{ of population members dominated by elite } i}{\text{N}+1}$$

  - For current population: assign fitness to a current population member *j*

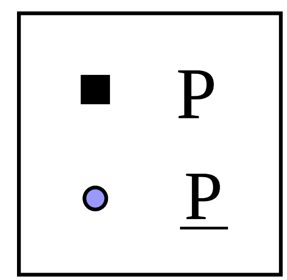
$$F_j = 1 + \sum_{i \in D_j} S_i$$

    where D_j is the set of all external population members dominating *j*
  - Note: a solution with smaller fitness is better

SPEA Algorithm Fitness Eval.



N= 6, $\underline{N}$=3



$$F_1 = 1 + \sum_{i \in D_j} S_i$$

$$= 1 + \frac{2}{7} = \frac{9}{7}$$

$$S_a = \frac{\#\text{ of current population members dominated by } \mathbf{a}}{N+1} = \frac{2}{7}$$

SPEA Algorithm

- Step 6
  - Apply a binary tournament selection (in a minimization sense), a crossover and a mutation operator to $P_t \cup \underline{P}_t$ and generate the next population P_{t+1}

Advantages of SPEA

- Once a solution Pareto-optimal front is found, it is stored in the external population
- Clustering ensures a spread among the non-dominated solutions
- The proposed clustering algorithm is parameterless

Disadvantages of SPEA

- A balance between the regular population size N and the external population size <u>N</u> is important
  - If <u>N</u> is too large, selection pressure for the elites is too large and the algorithm may not converge to the Pareto-optimal front
  - If <u>N</u> is too small, the effect of elitism will be lost