Multi-Objective Optimization

2019.10.27.

Jaesung Lee

Multi-Objective Optimization

- Most real-world problems involve simultaneous optimization of **several objective functions**.
- Generally, these objective functions are **measured** in different units and are often competing and conflicting.
- **Multi-objective optimization** having such conflicting objective functions gives rise to **a set of optimal solutions**, instead of one optimal solution because no solution can be considered to be better than any other with respect to all objectives.
- These optimal solutions are known as **Pareto-optimal** solutions.

Multi-Objective Optimization

- Multi-objective optimization problems with a number of objectives and a number of equality and inequality constraints can be **formulated** as:

$$
\begin{array}{ll}
\text{Minimize / Maximize} & f_i(x),\ i = 1, \ldots, N_{obj} \\
\text{Subject to:} & g_k(x) = 0,\ k = 1, \ldots, K \\
 & h_l(x) \leq 0,\ l = 1, \ldots, L
\end{array}
$$

- Where,
  - $f_i(x)$ is the objective function,
  - x is a decision vector that represents a solution
  - N_{obj} is the number of objectives, and
  - K and L are number of equality and inequality constraints respectively.

Multi-Objective Optimization

- Optimization problems with a number of **objective functions** to be satisfied.
- The objective functions may be **conflicting** with one another.
- In order to simplify the solution process, additional objective functions are usually handled as **constraints**.
- **Multiple objective functions** are handled at the same time.

Pareto-Optimality

Pareto Improvement

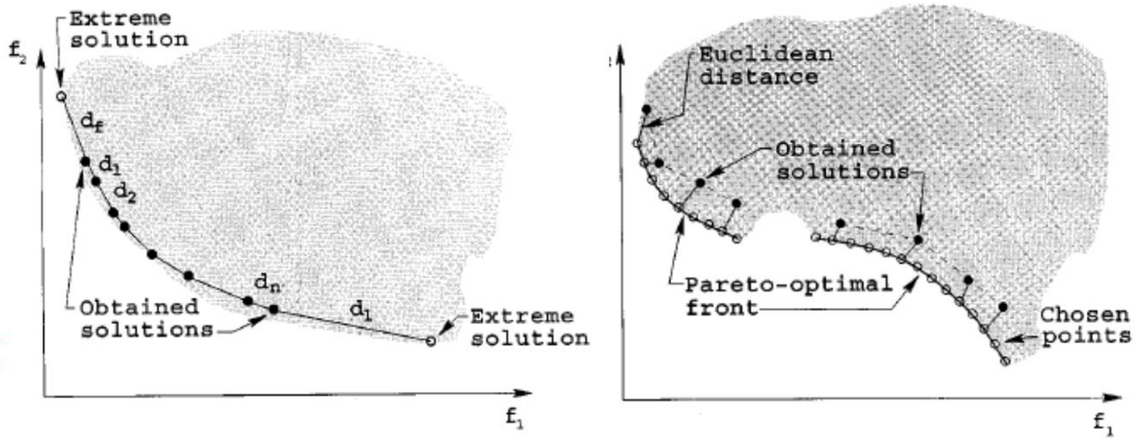
- With a set of **feasible solutions** and different **objective functions** to pursue, Pareto Improvement is a movement from **one feasible solution** to **another** that can make
  - At least **one objective function** to return a **better value**
  - With **no other** objective function becoming **worse off**

Pareto efficient or Pareto optimal

- A set of **feasible solutions** are Pareto efficient or Pareto optimal when **no further** Pareto improvements can be made.

Pareto-Optimality

- Solutions along the line are all **non-dominated** solutions.
- **Dominated** solutions are **inside** the line as there is another solution on the line with **at least** one objective that is better.
- The line is the **Pareto-optimal front** and the solutions on it are called **Pareto-optimal**.
- All Pareto-optimal solutions are **non-dominated**.



Pareto-Optimal Front

Source: "A Fast and Efficient Multiobjective Genetic Algorthm: NSGA-II" - Deb et al. IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, APRIL 2002

It is important to find solutions as close as possible to the Pareto front and as far along it as possible.

Example of Pareto-Optimal Solution

Ticket Type	Time (Hours)	Cost (1000 Won)
A	**2**	**7.5**
B	**3**	**6**
C	**3**	**7.5**
D	4	5
E	4	6.5
F	5	4.5
G	5	6
H	5	7
I	6	6.5

- There are 9 types of air tickets with different **time** and **cost** details.
- We need to choose from them with objective functions of **minimum cost** and **minimum time**.
- If we compare air tickets A and B - we see that while **A is better** from **Time point of view**, **B is better** from the **Cost** angle. **A and B** thus forms a **non-dominated set**.
- However, if we compare B and C, we see that B is equal to C in time, but better than C in Cost. Hence, we can say that **B "dominates" C**.
- So, as long as B is a feasible option, there is no reason to choose C.

Example of Pareto-Optimal Solution

- 9 air tickets to choose from them with objective functions of **minimum cost** and **minimum time**.

A "dominates" C
B "dominates" C, E, G, H, I
D "dominates" E, G, H, I
F "dominates" G, H, and I

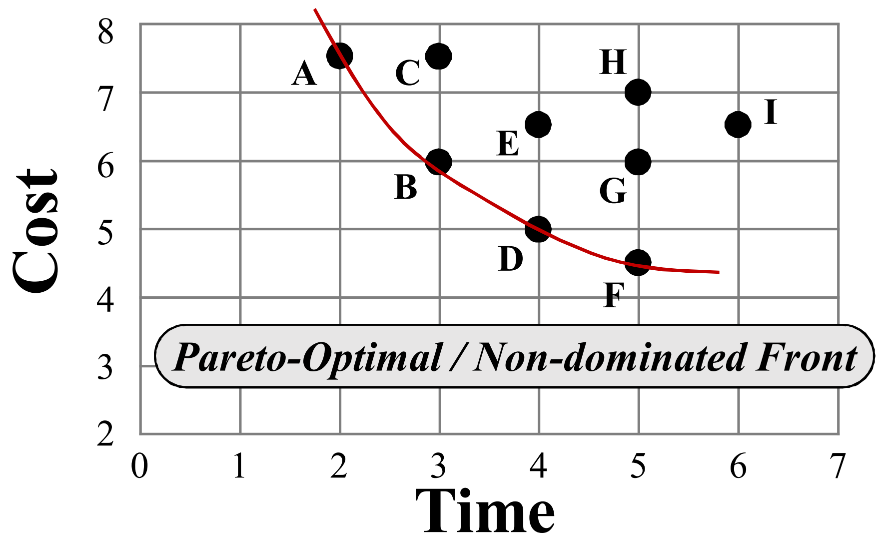
Considering any two air tickets between A, B, D, or F, we find that they do not dominate over each other!

Ticket Type	Time (Hours)	Cost (1000 Won)
A	**2**	**7.5**
B	**3**	**6**
C	3	7.5
D	**4**	**5**
E	4	6.5
F	**5**	**4.5**
G	5	6
H	5	7
I	6	6.5

- Thus, A, B, D, and F form a **"Non-dominated set"**.
- Hence, A, B, D, and F will form the **Pareto-Optimal Front** or **Non-Dominated Front** and they are the **Pareto-Optimal solutions**.

Example of Pareto-Optimal Solution

- 9 air tickets to choose from them with objective functions of **minimum cost** and **minimum time**.



Ticket Type	Time (Hours)	Cost (1000 Won)
A	**2**	**7.5**
B	**3**	**6**
C	3	7.5
D	**4**	**5**
E	4	6.5
F	**5**	**4.5**
G	5	6
H	5	7
I	6	6.5

A, B, D and F form a "Non-dominated set" and are "Pareto-Optimal solutions."

Evolutionary Optimization

- **Evolutionary algorithms** are particularly suitable to solve multi-objective optimization problems as they deal with a set of possible solutions **simultaneously**.
- Evolutionary algorithms are capable of finding several members of the **Pareto-Optimal set** in a **single run** of the algorithm.
- Evolutionary algorithms are less susceptible to the **shape** or continuity of the Pareto Front.
- Evolutionary algorithm can deal with **discontinuous** or **concave** Pareto Fronts.
- There are both **Non-Pareto** and **Pareto** techniques available for Multi-Objective Optimization using Evolutionary Optimization techniques.

Pareto vs. Non-Pareto Evolutionary Techniques

Non-Pareto Techniques

- Approaches that do not incorporate **directly** the concept of Pareto optimum.
- Incapable of producing **certain portions** of the Pareto Front.
- Efficient and easy to implement, but appropriate to handle only **a few** objectives.

Pareto Techniques

- Suggested originally by **Goldberg** (1989) to solve multi-objective problems.
- Use of **non-dominated ranking** and selection to move the population towards the **Pareto Front**.
- Require a **ranking procedure** and the technique to **maintain diversity** in the population.

Pareto vs. Non-Pareto Evolutionary Techniques

Non-Pareto Techniques

- Aggregating approaches
- Vector evaluated genetic algorithm (VEGA)
- Lexicographic ordering
- The ϵ-constraint method
- Target-vector approaches

Pareto Techniques

- Multi-objective genetic algorithm (MOGA)
- **Non-dominated sorting genetic algorithm-II (NSGA-II)**
- Multi-objective particle swarm optimization (MOPSO)
- Pareto evolution archive strategy (PAES)
- Strength Pareto evolutionary algorithm (SPEA-II)

Non-dominated Sorting Genetic Algorithm-II

Non-dominated Sorting Genetic Algorithm-II is known as NSGA-II

- Proposed by Deb et al. in 2000

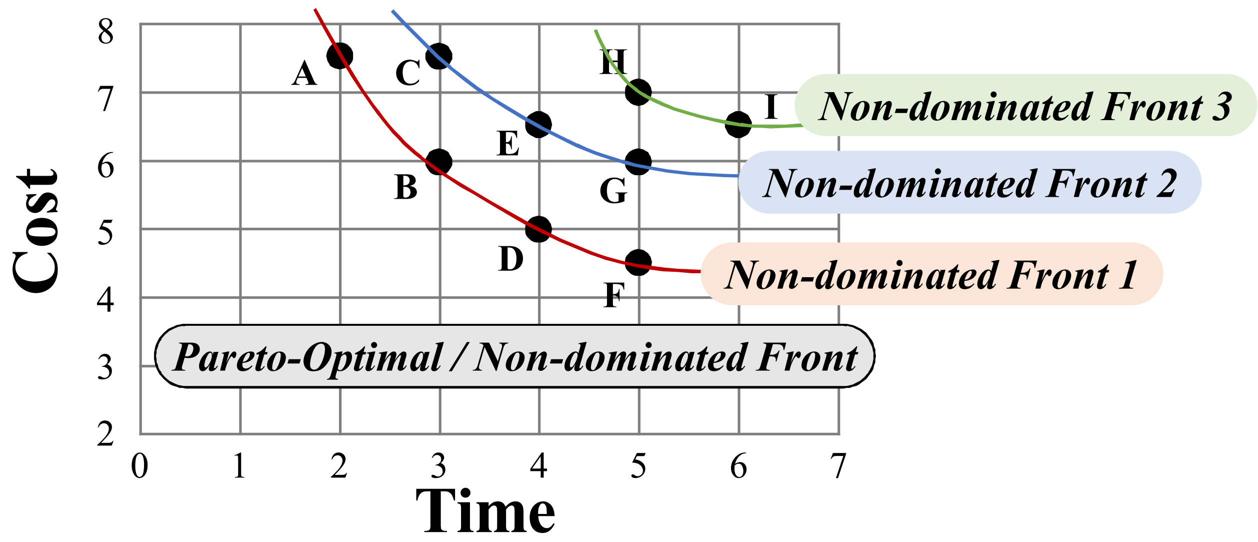
Key features:

- Emphasizes **non-dominated sorting**
- Use **diversity preserving** mechanism
- Does **crowding comparison**
- Uses **elitist principle**: Some of the parents go directly to the next generation based on the above-mentioned conditions.

Non-dominated Sorting

- 9 air tickets to choose from them with objective functions of **minimum cost** and **minimum time**.

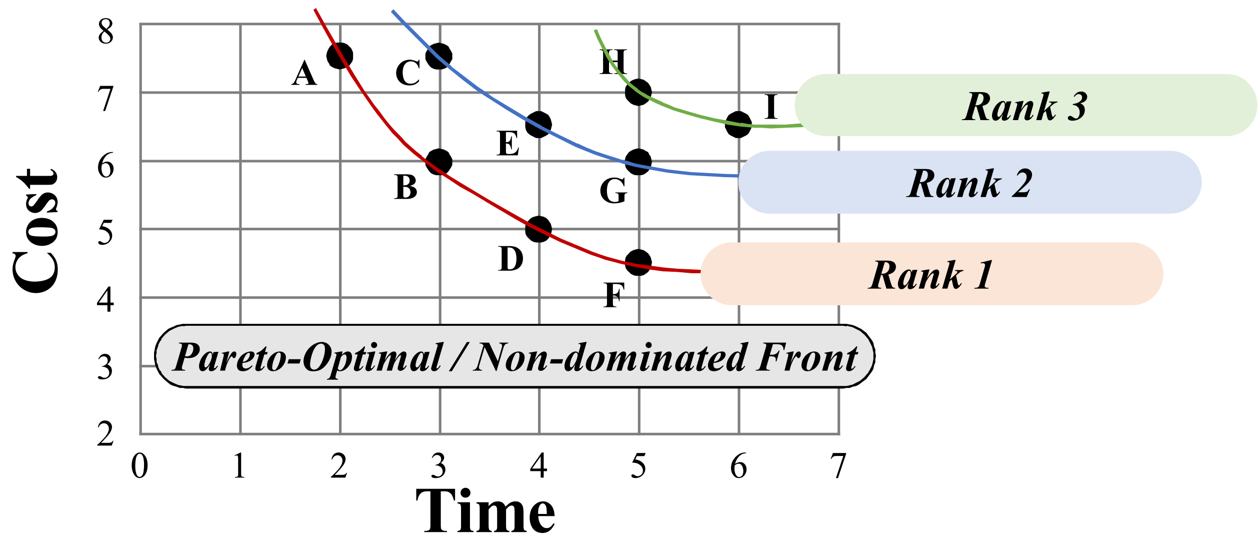
Ticket Type	Time (Hours)	Cost (1000 Won)
A	2	7.5
B	3	6
C	3	7.5
D	4	5
E	4	6.5
F	5	4.5
G	5	6
H	5	7
I	6	6.5



A, B, D and F form a "Non-dominated set" and are "Pareto-Optimal solutions."

Non-dominated Sorting

- 9 air tickets to choose from them with objective functions of **minimum cost** and **minimum time**.



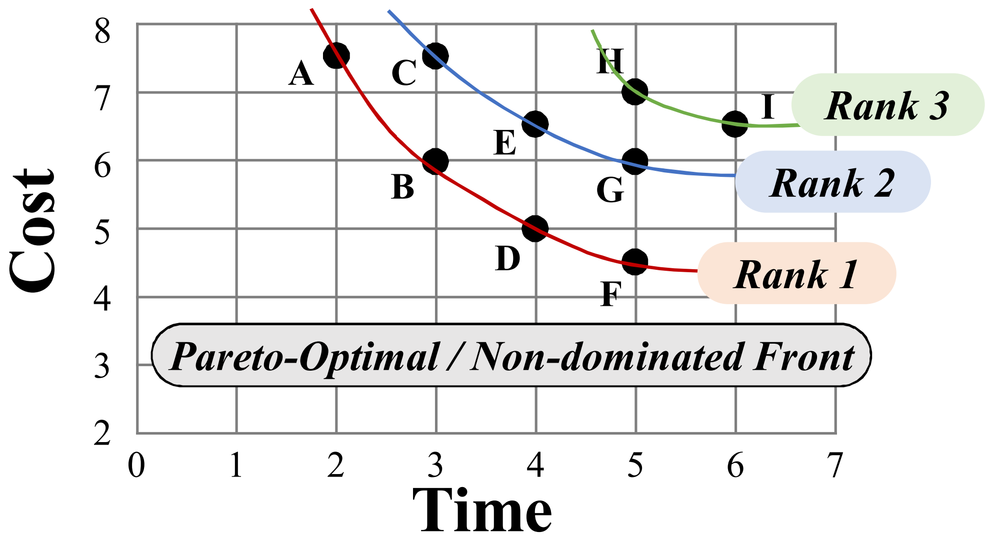
Ticket Type	Time (Hours)	Cost (1000 Won)
A	2	7.5
B	3	6
C	3	7.5
D	4	5
E	4	6.5
F	5	4.5
G	5	6
H	5	7
I	6	6.5

A, B, D and F form a "Non-dominated set" and are "Pareto-Optimal solutions."

Non-dominated Rank Comparison

- Given two solutions i and j, **i is preferred to j if**:

$$Rank(i) < Rank(j)$$



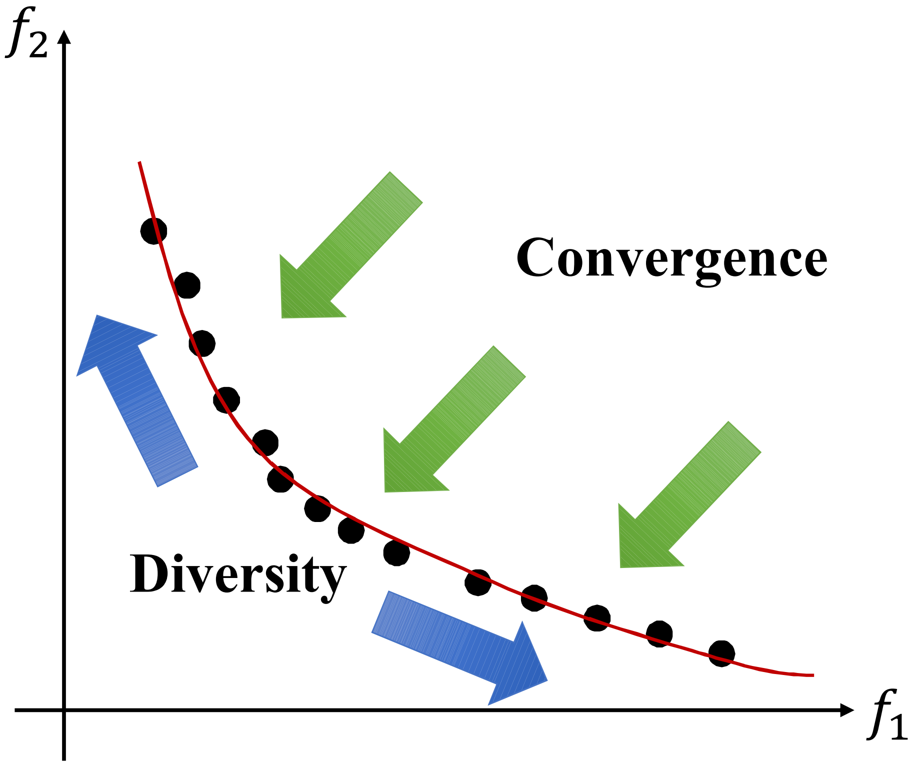
Fitness assignment:

- **Non-dominated Front 1** are **ranked 1** and have the **highest** fitness.
- Solutions in the **same front** have the **same rank** and **same fitness**.
- A solution with lower **non-dominated rank** is preferred over others.

When two solutions have the same non-dominated rank (belong to the same front), the one located in a less crowded region of the front is preferred.

NSGA-II Key Features

Convergence vs. Diversity



- Convergence to Pareto Optimal Frontier
- Diversity (representation of the entire Pareto Optimal Frontier)

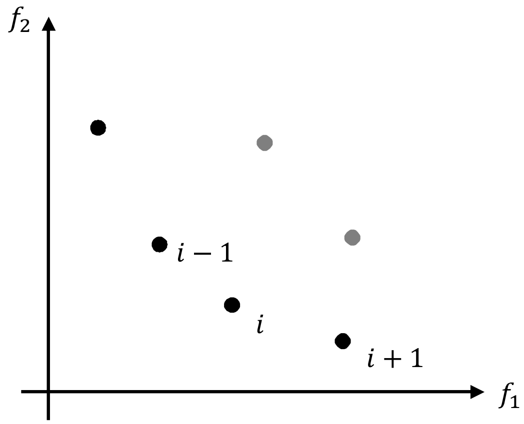
More the Crowding Distance, more the solution is likely to be retained for diversity preservation

NSGA-II Key Features

Diversity Preservation / Crowding Distance (CD)

1. **Sort all 'l' solution in a Pareto Front in ascending order of f_m and compute:**

$$CD_{im} = \frac{f_m(x_{i+1}) - f_m(x_{i-1})}{f_m(x_{max}) - f_m(x_{min})}, i = 2, \dots, (l-1)$$



2. **Repeat step 1 for each objective and find the crowding distance of solution *i* as:**

$$CD_i = \sum_{m=1}^{M} CD_{im}$$

More the Crowding Distance, more the solution is likely to be retained for diversity preservation.

3. **Given two solutions *i* and *j*, solution *i* is preferred to solution *j* if:**

$$Rank(i) < Rank(j) \text{ or } (Rank(i) = Rank(j) \text{ and } CD_i > CD_j)$$

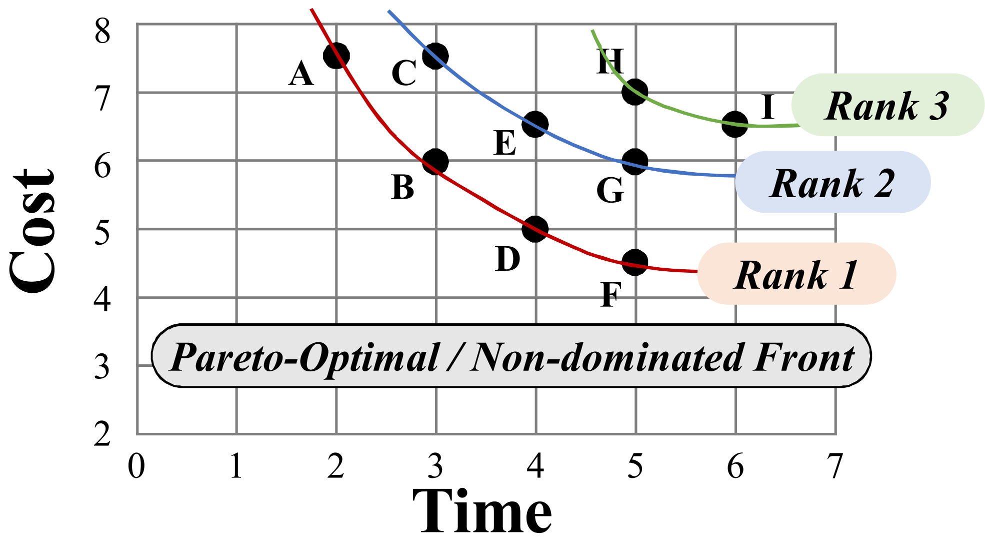
Computing Crowding Distances

$$CD_{im} = \frac{f_m(x_{i+1}) - f_m(x_{i-1})}{f_m(x_{max}) - f_m(x_{min})}$$

Ticket	Time	Cost
A	2	7.5
B	3	6
C	3	7.5
D	4	5
E	4	6.5
F	5	4.5
G	5	6
H	5	7
I	6	6.5

$f_{time}(x_{max}) = 6$
$f_{time}(x_{min}) = 2$
$f_{cost}(x_{max}) = 7.5$
$f_{cost}(x_{min}) = 4.5$

- What are the crowding distances of solutions A, B, D, and F in the Non-dominated Front 1?



Crowding Distance of A, B, D, F

- $CD_A = \infty$
- $CD_B = \frac{4-2}{6-2} + \frac{7.5-5}{7.5-4.5} = 0.50 + 0.83 = 1.33$
- $CD_D = \frac{5-3}{6-2} + \frac{6-4.5}{7.5-4.5} = 0.50 + 0.50 = 1.00$
- $CD_F = \infty$

QnA