

# Multi objective optimization

## MOP

## MOP general formulation

$$\left. \begin{array}{l}
 \text{Minimize/Maximize } f_m(\mathbf{x}), \quad m = 1, 2, \dots, M; \\
 \text{subject to } g_j(\mathbf{x}) \geq 0, \quad j = 1, 2, \dots, J; \\
 h_k(\mathbf{x}) = 0, \quad k = 1, 2, \dots, K; \\
 x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \dots, n.
 \end{array} \right\} \quad (2.1)$$

Multi-objective optimization is sometimes referred to as *vector optimization*, because a vector of objectives, instead of a single objective, is optimized.

## فضای اهداف ، فضای تصمیم

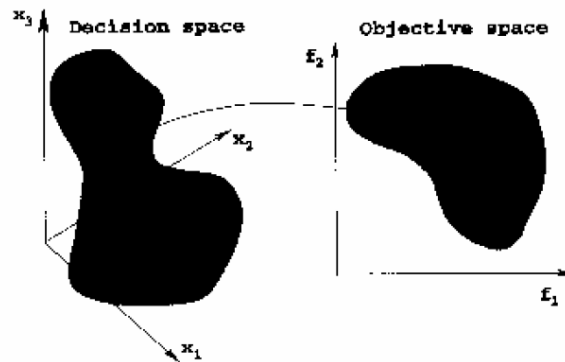


Figure 5 Representation of the decision variable space and the corresponding objective space.

**Definition 2.1.** A function  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  is a convex function if for any two pair of solutions  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in \mathbb{R}^n$ , the following condition is true:

$$f(\lambda \mathbf{x}^{(1)} + (1 - \lambda) \mathbf{x}^{(2)}) \leq \lambda f(\mathbf{x}^{(1)}) + (1 - \lambda) f(\mathbf{x}^{(2)}), \quad (2.2)$$

for all  $0 < \lambda \leq 1$ .

The above definition gives rise to the following properties of a convex function:

1. The linear approximation of  $f(\mathbf{x})$  at any point in the interval  $[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}]$  always underestimates the actual function value.
2. The Hessian matrix of  $f(\mathbf{x})$  is positive definite for all  $\mathbf{x}$ .
3. For a convex function, a local minimum is always a global minimum.<sup>1</sup>

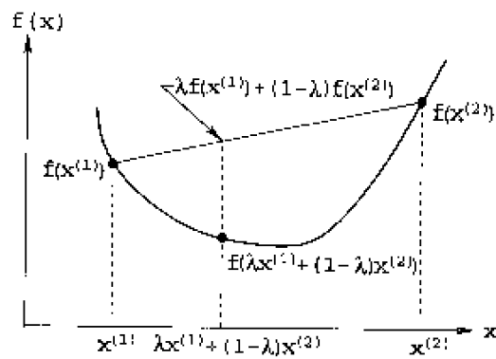


Figure 6 A convex function is illustrated. A line joining function values at two points  $x^{(1)}$  and  $x^{(2)}$  always estimates a large value of the true convex function.

## تست برای محدب بودن

test if a function is convex within an interval, the Hessian matrix  $\nabla^2 f$  is calculated and checked for its positive-definiteness at all points in the interval. One of the ways to check the positive-definiteness of a matrix is to compute the eigenvalues of the matrix and check to see if all eigenvalues are positive. To test if a function  $f$  is nonconvex in an interval, the Hessian matrix  $-\nabla^2 f$  is checked for its positive-definiteness. If it is positive-definite, the function  $f$  is nonconvex.

## Mathematical Programming Problem

- Multiple objectives, constraints, and variables

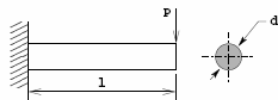
$$\begin{aligned} \text{Min/Max} \quad & (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})) \\ \text{Subject to} \quad & g_j(\mathbf{x}) \geq 0 \\ & h_k(\mathbf{x}) = 0 \\ & \mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)} \end{aligned}$$

Jamshid Nazemi

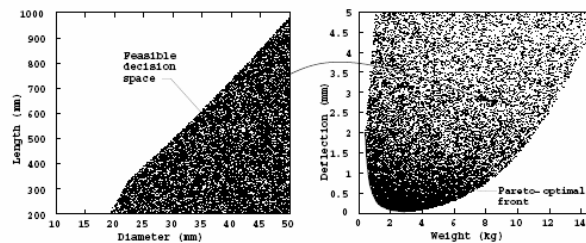
Multi Criteria Decision making

Slide no.: 7

## An Engineering Example



$$\begin{aligned} \text{Minimize} \quad & f_1(d, l) = \rho \frac{\pi d^2}{4} l \\ \text{Minimize} \quad & f_2(d, l) = \delta = \frac{64Pl^3}{3E\pi d^4} \\ \text{subject to} \quad & \sigma_{\max} \leq S_y \\ & \delta \leq \delta_{\max} \end{aligned}$$



Jamshid Nazemi

Multi Criteria Decision making

Slide no.: 8

## تصميم گيرى چند معياره :

تعيين طول و قطر يك تكيه گاه

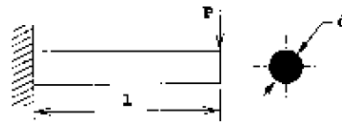


Figure 9 A schematic of a cantilever beam.

$$\begin{aligned} &\text{Minimize } f_1(d, l) = \rho \frac{\pi d^2}{4} l, && \left. \begin{array}{l} \leftarrow \text{وزن} \\ \leftarrow \text{انحراف} \end{array} \right\} \\ &\text{Minimize } f_2(d, l) = \delta = \frac{64Pl^3}{3E\pi d^4}, \\ &\text{subject to } \sigma_{\max} \leq S_y, \\ &\quad \delta \leq \delta_{\max}, \\ &\quad \sigma_{\max} = \frac{32Pl}{\pi d^3}. && \leftarrow \text{تنش} \end{aligned}$$

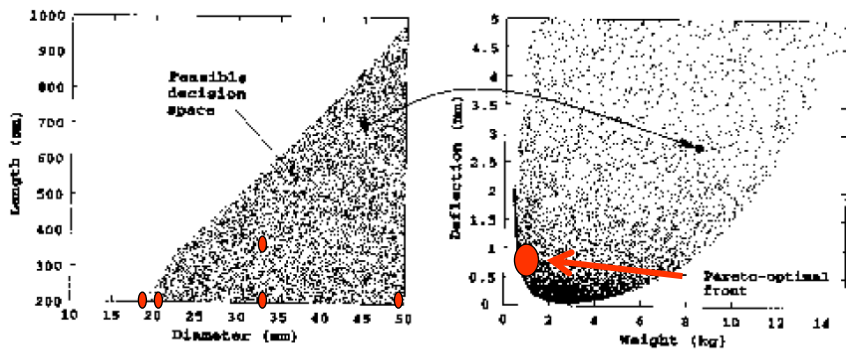


Figure 10 The feasible decision variable space (left) and the feasible objective space (right).

Table 1 Five solutions for the cantilever design problem.

Solution	d (mm)	l (mm)	Weight (kg)	Deflection (mm)
A	18.94	200.00	0.44	2.04
B	21.24	200.00	0.58	1.18
C	34.19	200.00	1.43	0.19
D	50.00	200.00	3.06	0.04
E	33.02	362.49	2.42	1.31

## مقایسه پاسخ ها

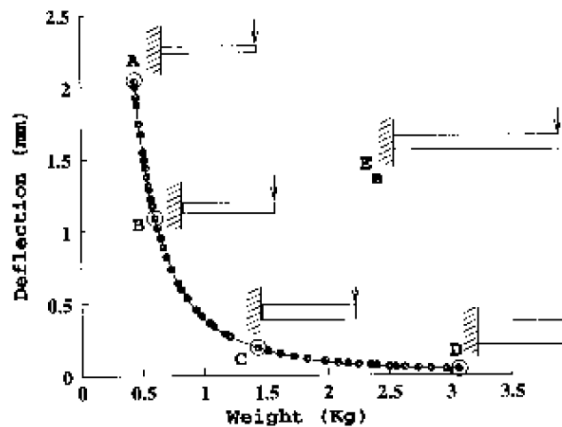


Figure 11 Four Pareto-optimal solutions and one non-optimal solution.

## بهینه یابی یک هدف

### 2.3.1 Two Goals Instead of One

In a single-objective optimization, there is one goal – the search for an optimum solution. Although the search space may have a number of local optimal solutions, the goal is always to find the global optimum solution. However, there is an exception. In the case of multi-modal optimization (see Section 4.6 later), the goal is to find a number of local and global optimal solutions, instead of finding one optimum solution. However, most single-objective optimization algorithms aim at finding one optimum solution, even when there exist a number of optimal solutions. In a single-objective optimization algorithm, as long as a new solution has a better objective function value than an old solution, the new solution can be accepted.

## جواب ایده آل

For each of the  $M$  conflicting objectives, there exists one different optimal solution. An objective vector constructed with these individual optimal objective values constitutes the ideal objective vector.

**Definition 2.3.** The  $m$ -th component of the ideal objective vector  $z^*$  is the constrained minimum solution of the following problem:

$$\left. \begin{array}{l} \text{Minimize } f_m(x) \\ \text{subject to } x \in S. \end{array} \right\} \quad (2.5)$$

Thus, if the minimum solution for the  $m$ -th objective function is the decision vector  $x^{*(m)}$  with function value  $f_m^*$ , the ideal vector is as follows:

$$z^* = f^* = (f_1^*, f_2^*, \dots, f_M^*)^T.$$

In general, the ideal objective vector corresponds to a non-existent solution. This is

## کاربرد جواب ایده آل

It is interesting to ponder the question: 'If the ideal objective vector is non-existent, what is its use?' In most algorithms which are seeking to find Pareto-optimal solutions, the ideal objective vector is used as a reference solution (we are using the word 'solution' corresponding to the ideal objective vector loosely here, realizing that an ideal vector represents a non-existent solution). It is also clear from Figure 13 that solutions closer to the ideal objective vector are better. Moreover, many algorithms require the knowledge of the lower bound on each objective function to normalize objective values in a common range, a matter which we shall discuss later



Figure 13 The ideal, utopian, and nadir objective vectors.

## محدوده پاسخ (تخیلی خوب و بد)

**Definition 2.4.** A utopian objective vector  $z^{**}$  has each of its components marginally smaller than that of the ideal objective vector, or  $z_i^{**} = z_i^* - \epsilon_i$  with  $\epsilon_i > 0$  for all  $i = 1, 2, \dots, M$ .

Figure 13 shows a utopian objective vector. Like the ideal objective vector, the utopian objective vector also represents a non-existent solution.

Although the ideal objective vector is easy to compute (except in complex multimodal objective problems), the nadir objective vector is difficult to compute in practice. However, for well-behaved problems (including linear MOOPs), the nadir objective vector can be derived from the ideal objective vector by using the *payoff table* method described in Miettinen (1999). For two objectives (Figure 13), if  $z^{*(1)} = (f_1(x^{*(1)}), f_2(x^{*(1)}))^T$  and  $z^{*(2)} = (f_1(x^{*(2)}), f_2(x^{*(2)}))^T$  are coordinates of the minimum solutions of  $f_1$  and  $f_2$ , respectively, in the objective space, then the nadir objective vector can be estimated as  $z^{nad} = (f_1(x^{*(2)}), f_2(x^{*(1)}))^T$ .

## پاسخ مسلط

**Definition 2.5.** A solution  $x^{(1)}$  is said to dominate the other solution  $x^{(2)}$ , if both conditions 1 and 2 are true:

1. The solution  $x^{(1)}$  is no worse than  $x^{(2)}$  in all objectives, or  $f_j(x^{(1)}) \leq f_j(x^{(2)})$  for all  $j = 1, 2, \dots, M$ .
2. The solution  $x^{(1)}$  is strictly better than  $x^{(2)}$  in at least one objective, or  $f_j(x^{(1)}) < f_j(x^{(2)})$  for at least one  $j \in \{1, 2, \dots, M\}$ .

If any of the above condition is violated, the solution  $x^{(1)}$  does not dominate the solution  $x^{(2)}$ . If  $x^{(1)}$  dominates the solution  $x^{(2)}$  (or mathematically  $x^{(1)} \preceq x^{(2)}$ ), it is also customary to write any of the following:

- $x^{(2)}$  is dominated by  $x^{(1)}$ ;
- $x^{(1)}$  is non-dominated by  $x^{(2)}$ , or;
- $x^{(1)}$  is non-inferior to  $x^{(2)}$ .

## مثال

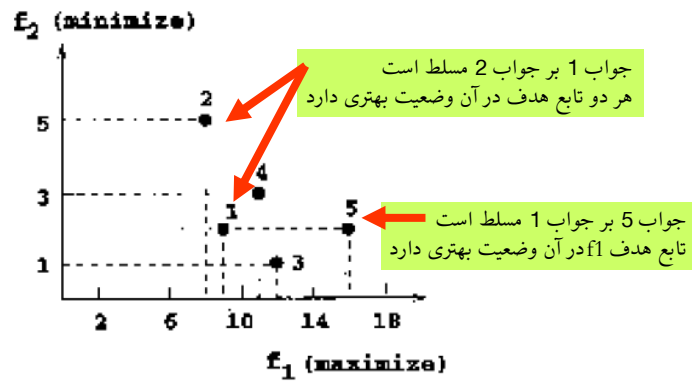
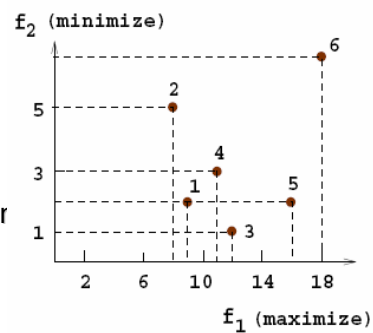


Figure 14 A population of five solutions.

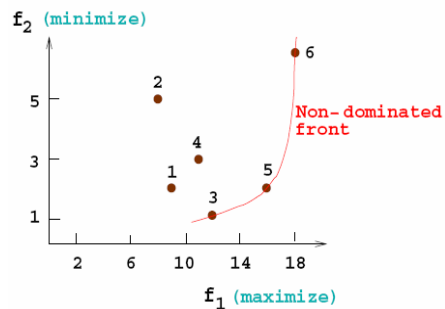
## Which Solutions are Optimal?

- ▶ Relates to the concept of domination
- ▶  $x^{(1)}$  dominates  $x^{(2)}$ , if
  - ▶  $x^{(1)}$  is no worse than  $x^{(2)}$  in all objectives
  - ▶  $x^{(1)}$  is strictly better than  $x^{(2)}$  in at least one objective
- ▶ Examples:
  - ▶ 3 dominates 2
  - ▶ 3 does not dominate 5



## Pareto-Optimal Solutions

- ▶  $P^* = \text{Non-dominated}(P)$   
Solutions which are not dominated by any member of the set  $P$
- ▶  $O(N \log N)$  algorithms exist
- ▶ Pareto-Optimal set = Non-dominated(S)
- ▶ A number of solutions are optimal



## انواع ترکیب جواب

There are three possibilities that can be the outcome of the dominance check between two solutions 1 and 2. i.e. (i) solution 1 dominates solution 2, (ii) solution 1 gets dominated by solution 2, or (iii) solutions 1 and 2 do not dominate each other. Let us now discuss the different binary relation properties (Cormen et al., 1990) of the dominance operator.

**Reflexive:** The dominance relation is *not reflexive*, since any solution  $p$  does not dominate itself according to Definition 2.5. The second condition of dominance relation in Definition 2.5 does not allow this property to be satisfied.

**Symmetric:** The dominance relation is also *not symmetric*, because  $p \prec q$  does not imply  $q \preceq p$ . In fact, the opposite is true. That is, if  $p$  dominates  $q$ , then  $q$  does not dominate  $p$ . Thus, the dominance relation is *asymmetric*.

**Antisymmetric:** Since the dominance relation is *not symmetric*, it cannot be antisymmetric as well.

**Transitive:** The dominance relation is *transitive*. This is because if  $p \prec q$  and  $q \preceq r$ , then  $p \preceq r$ .

## مجموعه جوابهای غیر مسلط

---

respect to each other. At the end, we expect to have a set of solutions, any two of which do not dominate each other. This set also has another property. For any solution outside of this set, we can always find a solution in this set which will dominate the former. Thus, this particular set has a property of dominating all other solutions which do not belong to this set. In simple term, this means that the solutions of this set are better compared to the rest of solutions. This set is given a special name. It is called the *non-dominated set* for the given set of solutions. In the example problem, solutions 3 and 5 constitutes the non-dominated set of the given set of five solutions. Thus, we define a set of non-dominated solutions as follows.

**Definition 2.6 (Non-dominated set).** *Among a set of solutions  $P$ , the non-dominated set of solutions  $P'$  are those that are not dominated by any member of the set  $P$ .*

---

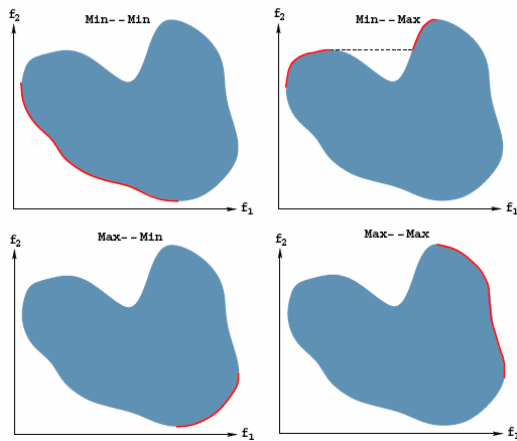
**Definition 2.7 (Globally Pareto-optimal set).** *The non-dominated set of the entire feasible search space  $S$  is the globally Pareto-optimal set.*

On many occasions, the globally Pareto-optimal set is simply referred to as the Pareto-optimal set. Since solutions of this set are not dominated by any feasible member of the search space, they are optimal solutions of the MOOP. We define a locally Pareto-optimal set as follows (Deb, 1999c; Miettinen, 1999).

**Definition 2.8.** *If for every member  $x$  in a set  $P$  there exists no solution  $y$  (in the neighborhood of  $x$  such that  $\|y - x\|_{\infty} \leq \epsilon$ , where  $\epsilon$  is a small positive number) dominating any member of the set  $P$ , then solutions belonging to the set  $P$  constitute a locally Pareto-optimal set.*

# Pareto-Optimal Fronts

- Depends on the type of objectives
- Always on the boundary of feasible region



Jamshid Nazemi

Multi Criteria Decision making

Slide no.: 23

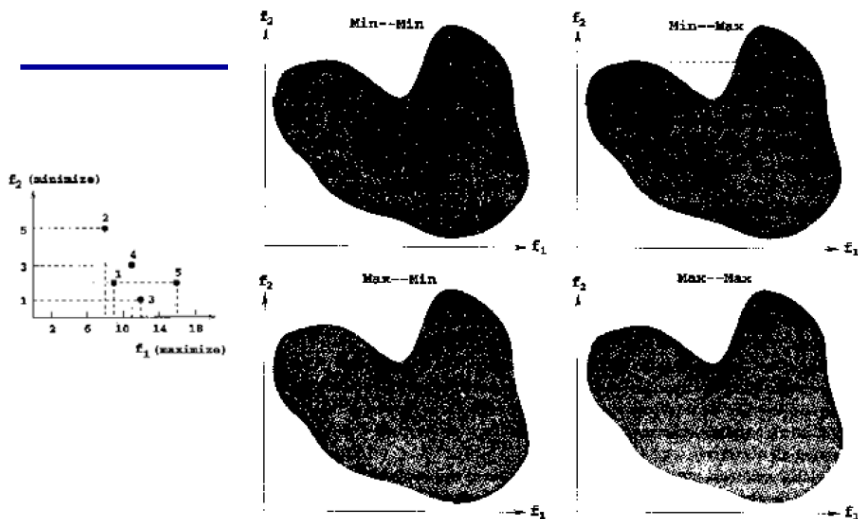


Figure 15 Pareto-optimal solutions are marked with continuous curves for four combinations of two types of objectives.

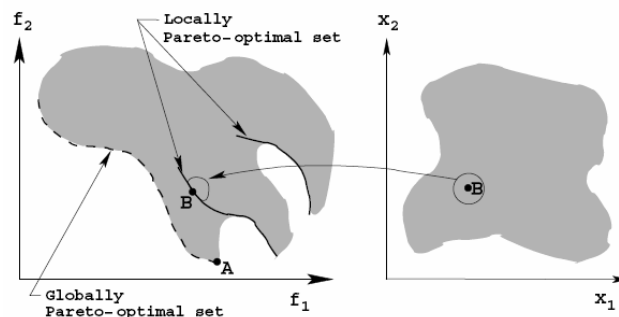
Jamshid Nazemi

Multi Criteria Decision making

Slide no.: 24

## Local Versus Global Pareto-Optimal Fronts

**Local Pareto-optimal Front:** Domination check is restricted within a neighborhood (in decision space) of  $P$



**Definition 2.9.** A solution  $x^{(1)}$  strongly dominates a solution  $x^{(2)}$  (or  $x^{(1)} \prec x^{(2)}$ ), if solution  $x^{(1)}$  is strictly better than solution  $x^{(2)}$  in all  $M$  objectives.

**Definition 2.10 (Weakly non-dominated set).** Among a set of solutions  $P$ , the weakly non-dominated set of solutions  $P'$  are those that are not strongly dominated by any other member of the set  $P$ .

The above definition suggests that a weakly non-dominated set found from a set of  $P$  solutions contains all members of the non-dominated set obtained by using Definition 2.6 from the same set  $P$ . In other words, for a given population of solutions, the cardinality of the weakly non-dominated set is greater than or equal to the cardinality of the non-dominated set obtained by using the usual Definition 2.6. The definition of a globally or locally weakly Pareto-optimal set can also be defined similarly by using the definition of the weakly non-dominated set.

## روش شناسایی مجموعه غیر مسلط

---

Identifying the Non-Dominated Set: Approach 1

- Step 1** Set solution counter  $i = 1$  and create an empty non-dominated set  $P'$ .
- Step 2** For a solution  $j \in P$  (but  $j \neq i$ ), check if solution  $j$  dominates solution  $i$ . If yes, go to Step 4.
- Step 3** If more solutions are left in  $P$ , increment  $j$  by one and go to Step 2; otherwise, set  $P' = P' \cup \{i\}$ .
- Step 4** Increment  $i$  by one. If  $i \leq N$ , go to Step 2; otherwise stop and declare  $P'$  as the non-dominated set.

## مثال - روش اول

---

- Step 1** We set  $i = 1$  and  $P' = \emptyset$ .
- Step 2** We compare solution 1 with all other solutions for domination, starting from solution 2. We observe that solution 2 does not dominate solution 1.
- Step 3** However, solution 3 dominates solution 1. Thus, we move to Step 4.
- Step 4** Solution 1 does not belong to the non-dominated set and we increment  $i$  to 2 and move to Step 2 to check the fate of solution 2.
- Step 2** We observe that solution 1 dominates solution 2. We therefore move to Step 4.
- Step 4** Thus, solution 2 does not belong to the non-dominated set. Next, we check solution 3.
- Steps 2 and 3** Starting from solution 1, we observe that neither solution 1 nor 2 dominates solution 3. In fact, solutions 4 and 5 also do not dominate solution 3. Thus, we include solution 3 in the non-dominated set,  $P' = \{3\}$ .
- Step 4** We now check solution 4.

## ادامه مثال - روش اول

**Step 2** Solution 5 dominates solution 4. Thus, it cannot be a member of  $P'$ .

**Step 4** Now we check the final solution (solution 5).

**Step 2** We observe that none of the solutions (1 to 4) dominates solution 5.

**Step 3** So, solution 5 also belongs to the non-dominated set. Thus, we update  $P' = \{3, 5\}$ .

**Step 4** We have now considered all five solutions and found the non-dominated set  $P' = \{3, 5\}$ .

## روش شناسایی مجموعه غیر مسلط

Identifying the Non-Dominated Set: Approach 2

**Step 1** Initialize  $P' = \{1\}$ . Set solution counter  $i = 2$ .

**Step 2** Set  $j = 1$ .

**Step 3** Compare solution  $i$  with  $j$  from  $P'$  for domination.

**Step 4** If  $i$  dominates  $j$ , delete the  $j$ -th member from  $P'$  or update  $P' = P' \setminus \{P'^{(j)}\}$ . If  $j < |P'|$ , increment  $j$  by one and then go to Step 3. Otherwise, go to Step 5. Alternatively, if the  $j$ -th member of  $P'$  dominates  $i$ , increment  $i$  by one and then go to Step 2.

**Step 5** Insert  $i$  in  $P'$  or update  $P' = P' \cup \{i\}$ . If  $i < N$ , increment  $i$  by one and go to Step 2. Otherwise, stop and declare  $P'$  as the non-dominated set.

## مثال - روش دوم

**Step 1**  $P' = \{1\}$ ; and we set  $i = 2$ .

**Step 2** We set the solution counter of  $P'$  as  $j = 1$  (which refers to solution 1).

**Step 3** We now compare solution 2 ( $i = 2$ ) with the lone member of  $P'$  (solution 1) for domination. We observe that solution 1 dominates solution 2. Since the  $j$ -th member of  $P'$  dominates solution  $i$ , we increment  $i$  to 3 and go to Step 2. This means that solution 2 does not belong to the non-dominated set.

**Step 2** The set  $P'$  still has solution 1 ( $j = 1$ ) only.

**Step 3** Now we compare solution 3 with solution 1. We observe that solution 3 ( $i = 3$ ) dominates solution 1. Thus, we delete the  $j$ -th (or the first) member from  $P'$  and update  $P' = \emptyset$ . Thus,  $P' = \emptyset$ . This depicts that a non-member of the non-dominated set gets deleted from  $P'$ . We now move to Step 5.

**Step 5** We insert  $i = 3$  in  $P'$  or update  $P' = \{3\}$ . Since  $i < 5$  here, we increment  $i$  to 4 and move to Step 2.

## ادامه مثال - روش دوم

**Step 2** We set  $j = 1$ , which refers to the lone element (solution 3) of  $P'$ .

**Step 3** By comparing solution 4 with solution 3, we observe that solution 3 dominates solution 4. Thus, we increment  $i$  to 5 and move to Step 2.

**Step 2** We still have solution 3 in  $P'$ .

**Step 3** Now, we compare solution 5 with solution 3. We observe that neither of them dominates the other. Thus, we move to Step 5.

**Step 5** We insert solution 5 in  $P'$  and update  $P' = \{3, 5\}$ . Since  $i = 5$ , we stop and declare  $P' = \{3, 5\}$  as the non-dominated set.

## روش شناسایی مجموعه غیر مسلط

Identifying the Non-Dominated Set: Approach 3

**Step 1** Sort the population according to the descending order of importance in the first objective function and rename the population as  $P$  of size  $N$ .

**Step 2, Front( $P$ )** If  $|P| = 1$ , return  $P$  as the output of **Front( $P$ )**. Otherwise,  $T = \text{Front}(P^{(1)} - P^{(|P|/2)})$  and  $B = \text{Front}(P^{(|P|/2+1)} - P^{(|P|)})$ . If the  $i$ -th non-dominated solution of  $B$  is not dominated by any non-dominated solution of  $T$ , create a merged set  $M = T \cup \{i\}$ . Return  $M$  as the output of **Front( $P$ )**.

**Step 1** Since the first objective is to be maximized in this problem, we sort the population according to the decreasing value of the first objective function. We obtain the following sequence:

$$P = \{5, 3, 4, 1, 2\}$$

**Step 2** When the entire set  $P$  enters **Front()**, the size of  $P$  is not one. Thus, we set  $T = \text{Front}(\{5, 3\})$  and  $B = \text{Front}(\{4, 1, 2\})$ . Recursively, the **Front**({5, 3}) would divide the set {5, 3} into two halves and make **Front**({5}) and **Front**({3}) as the next inner  $T$  and  $B$  sets. Since the sizes of these two inner sets are one each, these two solutions are sent as output to the above **Front()** calls. Moving up a step to **Front**({5, 3}) with  $T = \{5\}$  and  $B = \{3\}$ , the domination check and merging operation are then executed. We observe that solution 3 is not dominated by solution 5. Therefore, the merged set contains both solutions, or  $M = \{5, 3\}$ . Thus, the outcome of the  $T = \text{Front}(\{5, 3\})$  is the set {5, 3}.

Similarly, the **Front**({4, 1, 2}) operation makes two calls, i.e. **Front**({4}) and **Front**({1, 2}). The output from the former is {4}. The latter makes two more calls, i.e. **Front**({1}) and **Front**({2}), with outputs as {1} and {2}, respectively. Now, to complete the **Front**({1, 2}) operation, we need to check whether solution 2 is indeed dominated by solution 1. As we already know that solution 2 is dominated by solution 1, we have  $M = \{1\}$  as the output of **Front**({1, 2}). Now, to complete the **Front**({4, 1, 2}) operation, we need to check the domination of solutions 4 ( $T$ ) and 1 ( $B$ ). Since solution 1 is not dominated by solution 4, the merged set is  $M = \{4, 1\}$ , which is also the output of the **Front**({4, 1, 2}) call.

Now, for the overall call to **Front**( $P$ ), the sets are {5, 3} and {4, 1}, respectively. Checking the domination of the non-dominated solution (solution 1 in the second objective) in  $B$  with the non-dominated solution (solution 3 in the second objective) of  $T$ , we observe that both solution 1 gets dominated by solution 3 in the second objective. Thus, the output of the **Front**( $P$ ) call is the set {5, 3}, and this is the non-dominated set.

---

algorithms require us to find only the best non-dominated front in a population. These algorithms classify the population into two sets -- the non-dominated set and the remaining dominated set. However, there exist some algorithms which require the entire population to be classified into various non-domination levels. In such algorithms, the population needs to be sorted according to an ascending level of non-domination. The best non-dominated solutions are called non-dominated solutions of level 1. In order to find solutions for the next level of non-domination, there is a simple procedure which is usually followed. Once the best non-dominated set is identified, they are temporarily disregarded from the population.

---

#### Non-Dominated Sorting

- Step 1** Set all non-dominated sets  $P_j$ , ( $j = 1, 2, \dots$ ) as empty sets. Set non-domination level counter  $j = 1$ .
- Step 2** Use any one of the Approaches 1 to 3 to find the non-dominated set  $P^j$  of population  $P$ .
- Step 3** Update  $P_j = P^j$  and  $P = P \setminus P^j$ .
- Step 4** If  $P \neq \emptyset$ , increment  $j$  by one and go to Step 2. Otherwise, stop and declare all non-dominated sets  $P_i$ , for  $i = 1, 2, \dots, j$ .

---

**Step 1** We first set  $j = 1$  to identify the non-dominated solutions of the first level.

**Step 2** As shown earlier, the first non-dominated set is  $P^1 = \{3, 5\}$ .

**Step 3** Update  $P = P^1 - \{3, 5\}$  and modify  $P$  by deleting solutions 3 and 5 from it, or  $P = \{1, 2, 4\}$ .

**Step 4** Since  $P$  is a non-empty set, we move to Step 2 in search of solutions of the second non-domination level ( $j = 2$ ).

---

**Step 2** We now move to Approach 1, outlined above on page 34.

**Step 1** Set  $i = 1$  and  $P^i = \emptyset$ .

**Step 2** Solution 2 does not dominate solution 1.

**Step 3** We now check solution 4 with solution 1.

**Step 2** Solution 4 does not dominate solution 1 either.

**Step 3** We set  $P^i = \{1\}$ .

**Step 4** We increment  $i$  to 2 (check for solution 2) and move to Step 2.

**Step 2** Solution 1 dominates solution 2.

**Step 3** Thus, we move to Step 4.

**Step 4** We now check solution 4 for its inclusion in the second non-dominated set.

---

**Step 2** Solution 1 does not dominate solution 4.

**Step 3** Check with solution 2.

**Step 2** Solution 2 does not dominate solution 4 either.

**Step 3** We include solution 4 in the set or  $P' = \{1,4\}$ .

**Step 4** All three solutions are checked for non-domination. Thus, we declare the non-dominated set as  $P' = \{1,4\}$ .

**Step 3** Update  $P_2 = P' = \{1,4\}$  and modify  $P$  by deleting  $P'$  from it, or  $P = \{2\}$ .

**Step 4** We increment  $j$  to 3 and move to Step 2 in search of solutions in the third non-dominated set.

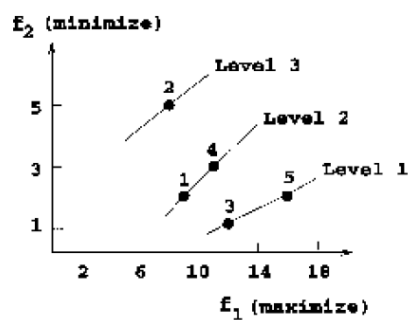


Figure 18 Solutions classified into various non-domination classes.

---

An  $O(MN^2)$  Non-Dominated Sorting Algorithm

- Step 1** For each  $i \in P$ ,  $n_i = 0$  and initialize  $S_i = \emptyset$ . For all  $j \neq i$  and  $j \in P$ , perform Step 2 and then proceed to Step 3.
- Step 2** If  $i \preceq j$ , update  $S_j = S_j \cup \{i\}$ . Otherwise, if  $j \preceq i$ , set  $n_i = n_i + 1$ .
- Step 3** If  $n_i = 0$ , keep  $i$  in the first non-dominated front  $P_1$  (we called this set  $P'$  in the above paragraph). Set a front counter  $k = 1$ .
- Step 4** While  $P_k \neq \emptyset$ , perform the following steps.
- Step 5** Initialize  $Q = \emptyset$  for storing next non-dominated solutions. For each  $i \in P_k$  and for each  $j \in S_i$ ,
- Step 5a** Update  $n_j = n_j + 1$ .
- Step 5b** If  $n_j = 0$ , keep  $j$  in  $Q$ , or perform  $Q = Q \cup \{j\}$ .
- Step 6** Set  $k = k + 1$  and  $P_k = Q$ . Go to Step 4.

---

**Theorem 2.5.1.** (Fritz-John necessary condition). A necessary condition for  $x^*$  to be Pareto-optimal is that there exist vectors  $\lambda \geq 0$  and  $u \geq 0$  (where  $\lambda \in \mathbb{R}^M$ ,  $u \in \mathbb{R}^l$  and  $\lambda, u \neq 0$ ) such that the following conditions are true:

1.  $\sum_{m=1}^M \lambda_m \nabla f_m(x^*) - \sum_{j=1}^l u_j \nabla g_j(x^*) = 0$ , and
2.  $u_j g_j(x^*) = 0$  for all  $j = 1, 2, \dots, l$ .

For a proof, readers may refer to Cunha and Polak (1967). Miettinen (1999) argues that the above theorem is also valid as the necessary condition for a solution to be weakly Pareto-optimal. Those readers familiar with the Kuhn-Tucker necessary conditions for single-objective optimization will immediately recognize the similarity between the above conditions and that of the single-objective optimization. The difference is in the inclusion of a  $\lambda$ -vector with the gradient vector of the objectives.

---

For an unconstrained MOOP, the above theorem requires the following condition:

$$\sum_{m=1}^M \lambda_m \nabla f_m(x^*) = 0$$

$n$ -variable unconstrained MOOP:

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_M}{\partial x_1} & \frac{\partial f_M}{\partial x_2} & \dots & \frac{\partial f_M}{\partial x_n} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_M \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (2.7)$$

---

For problems with  $n = M$  (identical number of decision variables and objectives), the Pareto-optimal solutions must satisfy the following:

$$\begin{vmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_M}{\partial x_1} & \frac{\partial f_M}{\partial x_2} & \dots & \frac{\partial f_M}{\partial x_n} \end{vmatrix} = 0. \quad (2.8)$$

The determinant of the partial derivative matrix must be zero for Pareto-optimal solutions. For a two-variable, two-objective MOOP, the above condition reduces to the following:

$$\frac{\partial f_1}{\partial x_1} \frac{\partial f_2}{\partial x_2} - \frac{\partial f_1}{\partial x_2} \frac{\partial f_2}{\partial x_1} = 0 \quad (2.9)$$

The following theorem offers sufficient conditions for a solution to be Pareto-optimal for convex functions.

**Theorem 2.5.2.** (Karush–Kuhn–Tucker sufficient condition for Pareto-optimality).  
 Let the objective functions be convex and the constraint functions of the problem shown in equation (2.1) be nonconvex. Let the objective and constraint functions be continuously differentiable at a feasible solution  $x^*$ . A sufficient condition for  $x^*$  to be Pareto-optimal is that there exist vectors  $\lambda > 0$  and  $u \geq 0$  (where  $\lambda \in \mathbb{R}^M$  and  $u \in \mathbb{R}^J$ ) such that the following equations are true:

1.  $\sum_{m=1}^M \lambda_m \nabla f_m(x^*) - \sum_{j=1}^J u_j \nabla g_j(x^*) = 0$ , and
2.  $u_j g_j(x^*) = 0$  for all  $j = 1, 2, \dots, J$ .

## Some Terminologies

► **Ideal point ( $z^*$ )**

- nonexistent, lower bound on Pareto-optimal set

► **Utopian point ( $z^{**}$ )**

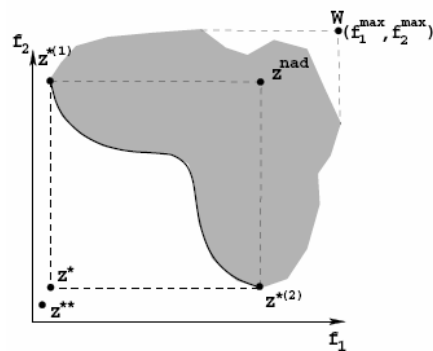
- nonexistent

► **Nadir point ( $z^{nad}$ )**

- Upper bound on Pareto-optimal set

► **Normalization:**

$$f_i^{norm} = \frac{f_i - z_i^*}{z_i^{nad} - z_i^*}$$



## Differences with Single-Objective Optimization

- ▶ One optimum versus multiple optima
- ▶ Requires search and decision-making
- ▶ Two spaces of interest, instead of one

