MULTIOBJECTIVE OPTIMIZATION CONCEPTS
Multiobjective optimization problems (MOP)

- There are many conflicting objectives to handle
- Pareto optimal solution
  - It is not possible to improve a given objective without deteriorating at least another objective
- Mostly focused on P-metaheuristics
MOP

\[
\text{MOP} = \begin{cases} 
\min F(x) = (f_1(x), f_2(x), \ldots, f_n(x)) \\
\text{s.c. } x \in S
\end{cases}
\]

- \( x = (x_1, \ldots, x_k) \): vector representing the decision variables
- \( S \): the set of feasible solutions
- \( F(x) \): the vector of objectives to be optimized
Decision space and objective space

FIGURE 4.1 Decision space and objective space in a MOP.
Definition 4.2  **Pareto dominance.** An objective vector \( u = (u_1, \ldots, u_n) \) is said to dominate \( v = (v_1, \ldots, v_n) \) (denoted by \( u < v \)) if and only if no component of \( v \) is smaller than the corresponding component of \( u \) and at least one component of \( u \) is strictly smaller, that is,

\[
\forall i \in \{1, \ldots, n\} : u_i \leq v_i \land \exists i \in \{1, \ldots, n\} : u_i < v_i
\]
Definitions

**Definition 4.3  Pareto optimality.** A solution $x^* \in S$ is Pareto optimal if for every $x \in S$, $F(x)$ does not dominate $F(x^*)$, that is, $F(x) \not\prec F(x^*)$.

**Definition 4.4  Pareto optimal set.** For a given MOP $(F, S)$, the Pareto optimal set is defined as $\mathcal{P}^* = \{x \in S/\forall x' \in S, F(x') \prec F(x)\}$. 
Pareto front

**Definition 4.5  Pareto front.** For a given MOP \((F, S)\) and its Pareto optimal set \(\mathcal{P}^*\), the Pareto front is defined as \(\mathcal{P}F^* = \{F(x), x \in \mathcal{P}^*\}\).

- The image of the Pareto optimal set in the objective space
- The main goal of MOP
Desired Pareto front

FIGURE 4.3 Examples of Pareto fronts: bad convergence and good diversity (left), good convergence and bad diversity (center), good convergence and diversity (right).
The ideal and nadir points give some information on the ranges of the Pareto optimal front.
Definitions

**Definition 4.9  Utility function.** A utility (or value) function \( v \), which represents the preferences of the decision maker, maps the objective vector to a scalar-valued function: \( v : \mathbb{R}^n \rightarrow \mathbb{R} \).

**Definition 4.10  Locally Pareto optimal solution.** A solution \( x \) is locally Pareto optimal if and only if \( \forall w \in N(x) \), \( F(w) \) does not dominate \( F(x) \), and \( N(x) \) represents the neighborhood of the solution \( x \).

\[
\begin{align*}
\text{(MOP}_\lambda\text{)} & \quad \min F(x) = \sum_{i=1}^{n} \lambda_i f_i(x) \\
\text{s.c. } x & \in S
\end{align*}
\]
Definitions

**Definition 4.11  Weak dominance.** An objective vector \( u = (u_1, \ldots, u_n) \) is said to weakly dominate \( v = (v_1, \ldots, v_n) \) (denoted by \( u \preceq v \)) if all components of \( u \) are smaller than or equal to the corresponding components of \( v \), that is, \( \forall i \in \{1, \ldots, n\}, \ u_i \leq v_i \) (Fig. 4.5).

**Definition 4.12  Strict dominance.** An objective vector \( u = (u_1, \ldots, u_n) \) is said to strictly dominate \( v = (v_1, \ldots, v_n) \) (denoted by \( u < v \)) if all components of \( u \) are smaller than the corresponding components of \( v \), that is, \( \forall i \in \{1, \ldots, n\}, \ u_i < v_i \).

**Definition 4.13  \( \epsilon \)-Dominance.** An objective vector \( u = (u_1, \ldots, u_n) \) is said to \( \epsilon \)-dominate \( v = (v_1, \ldots, v_n) \) (denoted by \( u \prec_{\epsilon} v \)) if and only if no component of \( v \) is smaller than the corresponding component of \( u \) and at least one component of \( u - \epsilon \) is strictly better, that is, \( \forall i \in \{1, \ldots, n\} : u_i - \epsilon_i \leq v_i \land \exists i \in \{1, \ldots, n\} : u_i - \epsilon_i < v_i \) (Fig. 4.6).
Weak dominance and strict dominance concepts. Solution $u$ weakly dominate solution $v$; solution $u'$ weakly dominates solution $v'$; solution $u$ strictly dominates solutions $v'$ and $v''$. 
MULTIOBJECTIVE OPTIMIZATION PROBLEM
Categories and applications

- Two categories:
  - continuous
  - discrete

- Academic applications
- Real-Life applications
Academic applications

Multiobjective Continuous Problems

\[
\min F(x) = (f_1(x), f_2(x), \ldots, f_n(x)) \\
\text{subject to} \\
g_i(x) \leq 0, \quad i = 1, \ldots, m \\
h_i(x) = 0, \quad i = 1, \ldots, p
\]

- Well-known test problems (standard benchmarks):
  - ZDT
  - DTLZ
Academic applications

- Multiobjective Combinatorial Problems
  - lack of “standard” benchmarks
  - Two complexity categories:
    - **Polynomial problems**: shortest path, spanning tree problems, assignment problems
    - **NP-hard problems**: scheduling problems, routing problems
Multiobjective scheduling problems

<table>
<thead>
<tr>
<th></th>
<th>$J_2$</th>
<th>$J_4$</th>
<th>$J_5$</th>
<th>$J_1$</th>
<th>$J_6$</th>
<th>$J_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_3$</td>
<td>$J_2$</td>
<td>$J_4$</td>
<td>$J_5$</td>
<td>$J_1$</td>
<td>$J_6$</td>
<td>$J_3$</td>
</tr>
</tbody>
</table>

- $C_{max}$: Makespan (total completion time): $\max\{C_i|i \in [1 \ldots n]\}$
- $\bar{C}$: Mean value of jobs completion time
- $T_{max}$: Maximum tardiness: $\max\{[\max(0, C_i - d_i)]|i \in [1 \ldots n]\}$
- $T$: Total tardiness: $\sum_{i=1}^{n}[\max(0, C_i - d_i)]$
- $U$: Number of jobs delayed with regard to their due date $d_i$
- $F_{max}$: Maximum job flow-time: $\max\{C_i - r_i|i \in [1 \ldots n]\}$
- $\bar{F}$: Mean job flow-time
Real-life applications

- Engineering design
- Environment and energetics
- Telecommunications
- Control
- Computational biology and bioinformatics
- Transportation and logistics
Multicriteria decision making

- There is a need for interaction between the decision maker and the problem solver:
  - A priori
  - A posteriori
  - Interactive