Main steps of the ABC algorithm
(1) cycle = 1
(2) Initialize the food source positions (solutions) x_i, $i = 1, \ldots, SN$
(3) Evaluate the nectar amount (fitness function fit_i) of food sources
(4) **repeat**
(5) Employed Bees' Phase
    For each employed bee
        Produce new food source positions v_i
        Calculate the value fit_i
        If new position better than previous position
        Then memorizes the new position and forgets the old one.
    End For.
(6) Calculate the probability values p_i for the solution.
(7) Onlooker Bees' Phase
    For each onlooker bee
        Chooses a food source depending on p_i
        Produce new food source positions v_i
        Calculate the value fit_i
        If new position better than previous position
        Then memorizes the new position and forgets the old one.
    End For
(8) Scout Bee Phase
    If there is an employed bee becomes scout
    Then replace it with a new random source positions
(9) Memorize the best solution achieved so far
(10) cycle = cycle + 1.
(11) **until** cycle = Maximum Cycle Number

Algorithm 1: Pseudocode for ABC algorithm.

Main steps of the MOABC algorithm
(1) cycle = 1
(2) Initialize the food source positions (solutions) $\mathbf{x}_i$, $i = 1, \ldots, SN$
(3) Evaluate the nectar amount (fitness fit_i) of food sources
(4) The initialized solutions are sorted based on nondomination
(5) Store nondominated solutions in the external archive EA
(6) **repeat**
(7) Onlooker Bees' Phase
    For each onlooker bee
        Randomly chooses a solution from EA
        Produce new solution v_i by using expression (4.1)
        Calculate the value fit_i
        Apply greedy selection mechanism in Algorithm 3 to decide which solution enters EA
    EndFor
(8) The solutions in the EA are sorted based on nondomination
(9) Keep the nondomination solutions of them staying in the EA
(10) If the number of nondominated solutions exceeds the allocated the size of EA
    Use crowding distance to remove the crowded members
(11) cycle = cycle + 1.
(12) **until** cycle = Maximum Cycle Number

Algorithm 2: Pseudocode for MOABC algorithm.

```
Greedy selection mechanism
If v_i dominates x_i
    Put v_i into EA
Else if x_i dominates v_i
    Do nothing
Else if x_i and v_i are not dominated by each other
    Put v_i into EA
    Produce a random number r drawn from a uniform distribution on the unit interval
    If r < 0.5
      The the original solution is replaced by the new solution as new food source position.
      That means x_i is replaced by v_i.
    Else
        Do nothing
    End If
End If
```

Algorithm 3: Greedy selection mechanism.

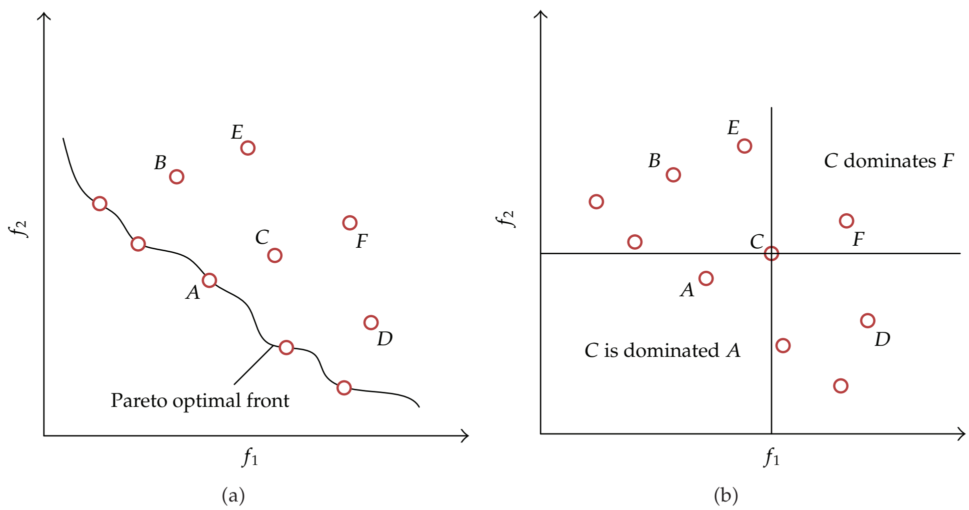

Figure 1: Illustrative example of Pareto optimality in objective space (a) and the possible relations of solutions in objective space (b).

Definition 2 (Pareto optimal). For (2.1), let $\mathbf{a} = (a_1, \ldots, a_m)$, $\mathbf{b} = (b_1, \ldots, b_m) \in R^m$ be two vectors, $\mathbf{a}$ is said to dominate $\mathbf{b}$ if $a_i \leq b_i$ for all $i = 1, \ldots, m$, and $\mathbf{a} \neq \mathbf{b}$. A point $\mathbf{x}^* \in X$ is called (globally) Pareto optimal if there is no $\mathbf{x} \in X$ such that $\mathbf{F}(\mathbf{x})$ dominates $\mathbf{F}(\mathbf{x}^*)$. Pareto optimal solutions are also called efficient, nondominated, and noninferior solutions. The set of all the Pareto optimal solutions, denoted by PS, is called the Pareto set. The set of all the Pareto objectives vectors, $\mathrm{PF} = \{\mathbf{F}(\mathbf{x}) \in R^m \mid \mathbf{x} \in \mathrm{PS}\}$, is called the Pareto front [11, 12]. Illustrative example can be seen in Figure 1.