Evolutionary Computation and Learning

Evolutionary Multi-Objective Optimisation: part 1

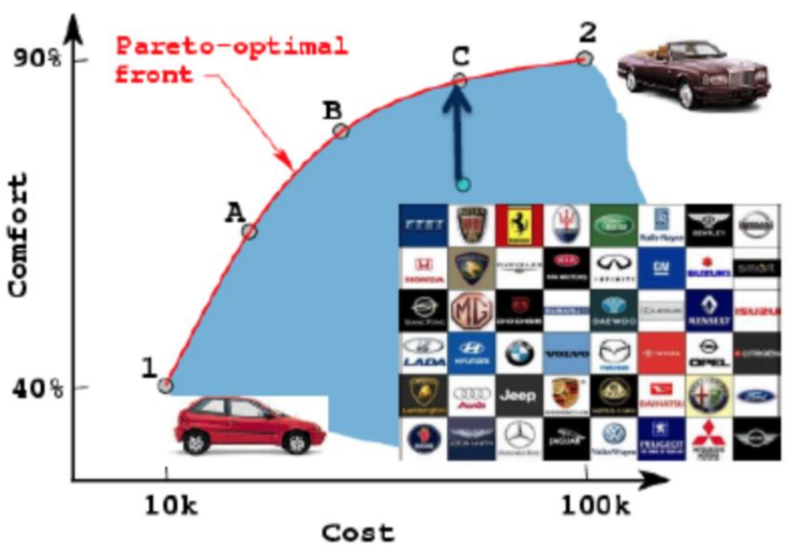
By: Dr. Vahid Ghasemi

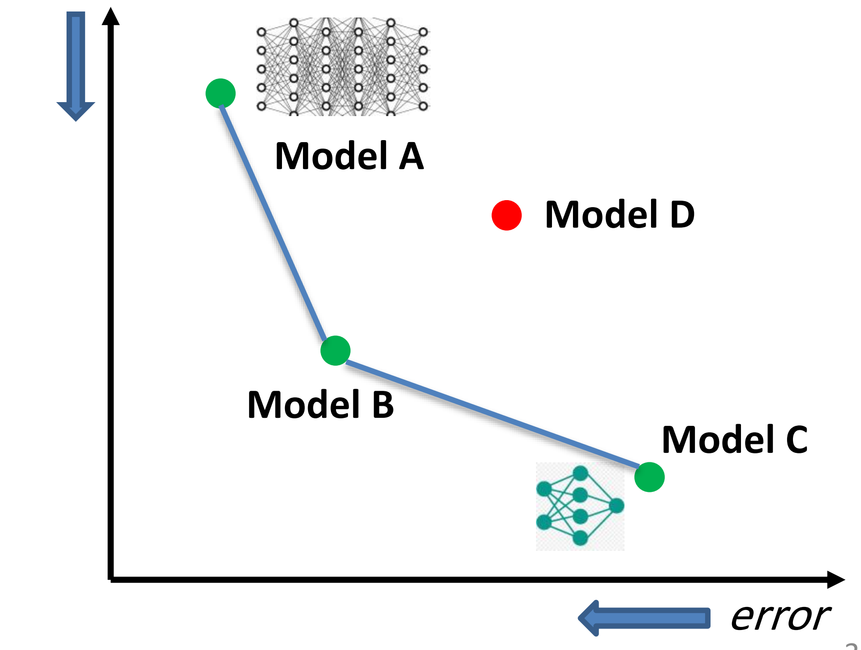
Outline

- Why and What is Multi-objective optimisation
- Dominance relation and Pareto optimality
- Evolutionary Multi-objective Optimisation
- Performance indicators
- Design issues

Multi-objective Optimisation

- Many real-world problems have multiple conflicting objectives
  - Buy a car: cost vs comfort
  - Learn a neural network: error vs complexity
  - Feature selection: accuracy vs #features
  - …
- NO single optimum





Multi-objective Optimisation

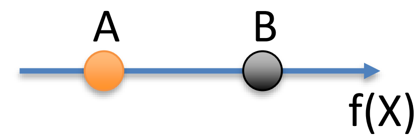
- Mininise a number of **conflicting** objectives
  - No single solution that can achieve optimal value for ALL objectives at the same time
- $\min \boldsymbol{F}(X) = \left(f_1(X), f_2(X), \ldots, f_N(X)\right)$

- A simple method is to aggregate multiple objectives into a single objective
- $\min f^{agg}(X) = w_1 f_1(X) + w_2 f_2(X) + \cdots + w_N f_N(X)$

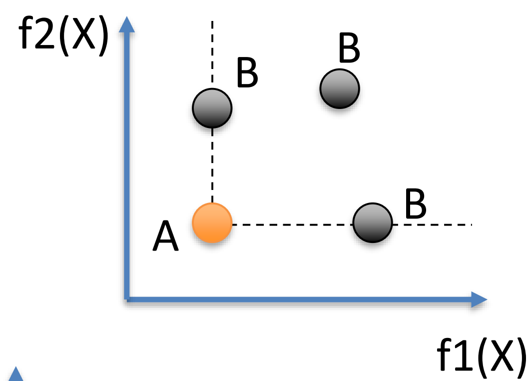
- But it is usually hard to determine proper weights
- Different weights will lead to different solutions
- Which solution is the best/optimal?

Dominance Relation

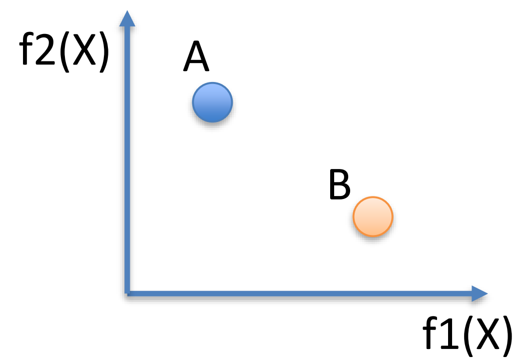
- How to compare two solutions? (minimise)
- **1 objective**:
  - A is better than B if its objective is smaller



- **2 objectives**:
  - A is better than B if its both objective is smaller
  - A is better than B if one objective is better, and the other is equal
- Any objectives:
  - **[Dominance]** A dominates B if
    - A is no worse than B in ALL objectives, and
    - A is better than B in at least one objective

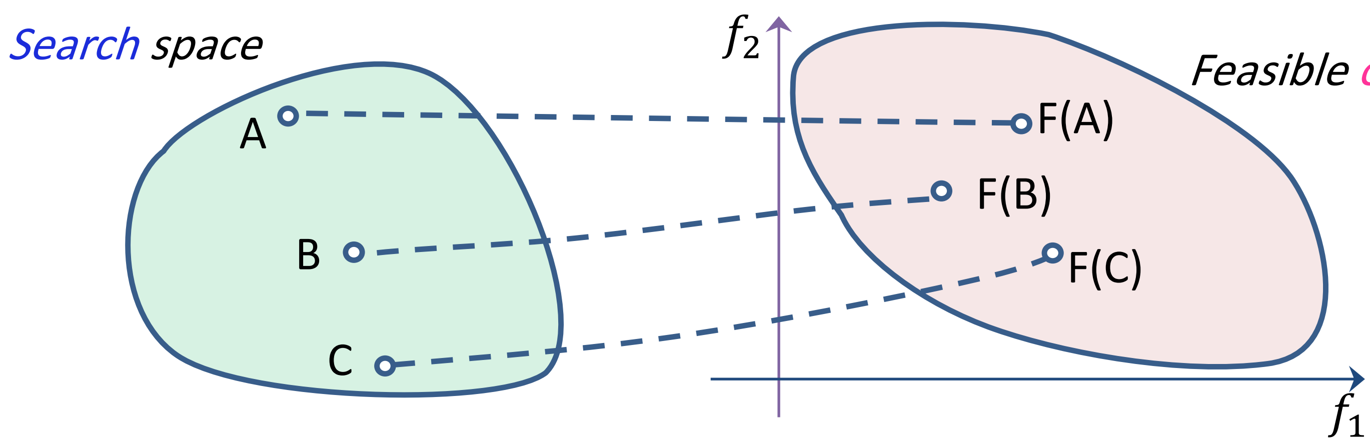


- "A dominates B" = "B is dominated by A"
- A and B are non-dominated
  - A does not dominate B
  - B does not dominate A



Dominance Relation

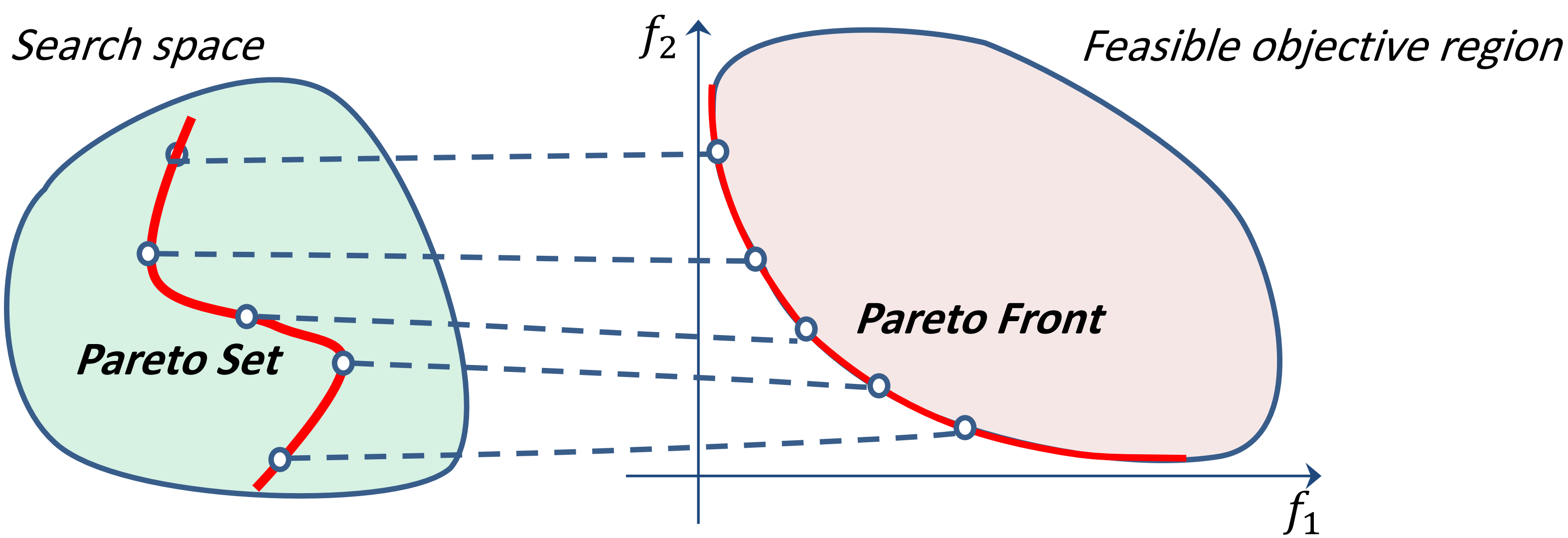
- Search/Solution space -> Objective space
  - Each possible solution X -> corresponding objectives $\boldsymbol{F}(X)$
- Dominance is on the objective space
  - B dominates A
  - B and C are non-dominated
  - A and C are non-dominated



Pareto Optimality

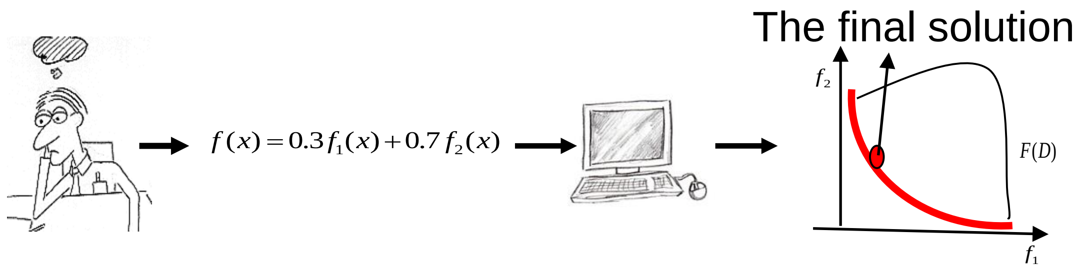
- Solution X is a Pareto optimal solution, if NO other solution in the objective space dominates it
- Pareto (optimal) set: All the Pareto optimal solutions
- Pareto front: The objectives of the Pareto optimal solutions
- In practice, usually Pareto front = Pareto set

The goal is to find the entire or part of the Pareto front

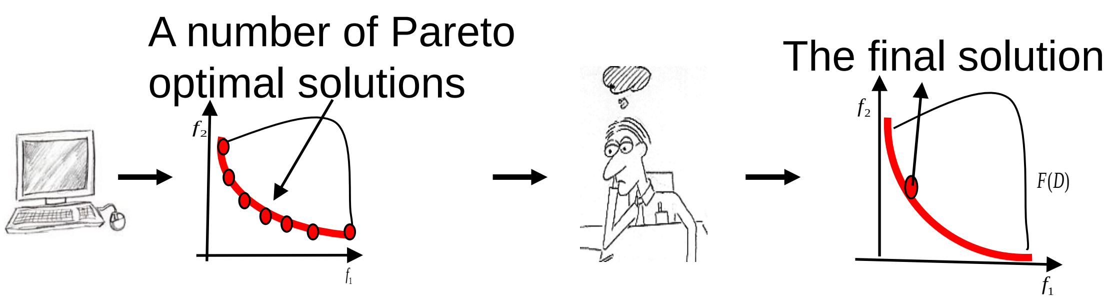


Multi-Objective Decision Making

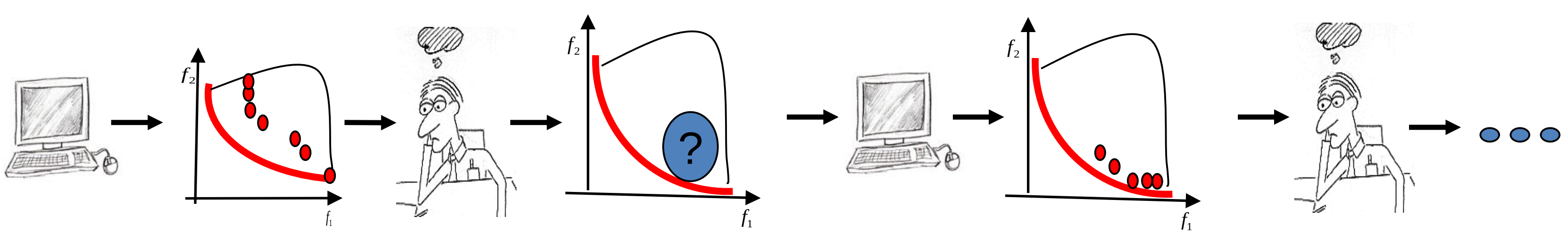
- A-priori



- A-posteriori



- Interactive

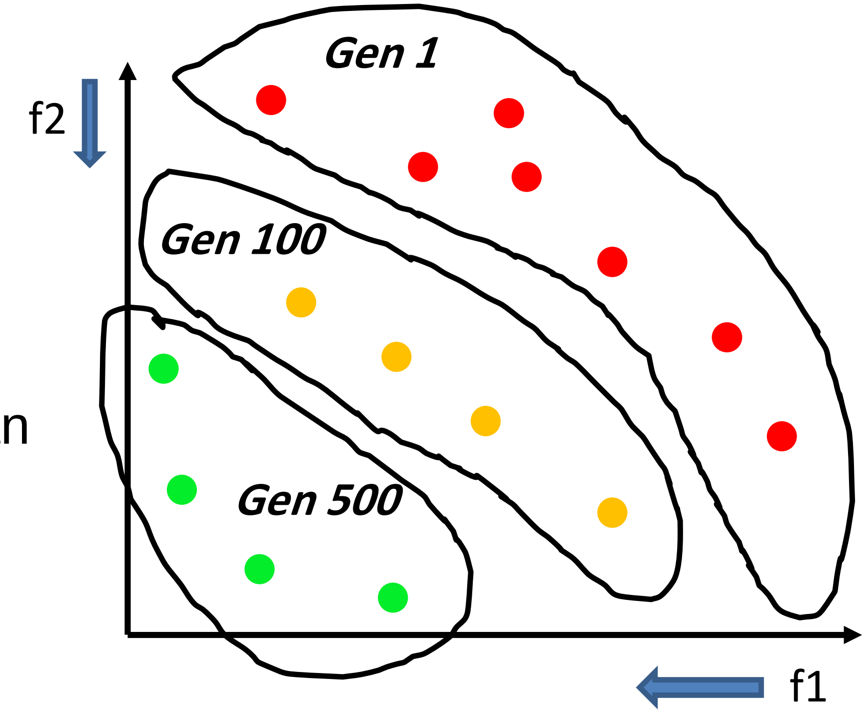


Evolutionary Multi-objective Optimisation

- Use evolutionary computation to solve multi-objective optimisation
- EC uses a population to search, easy to search towards different parts of the Pareto front, and maintain multiple Pareto optimal solutions

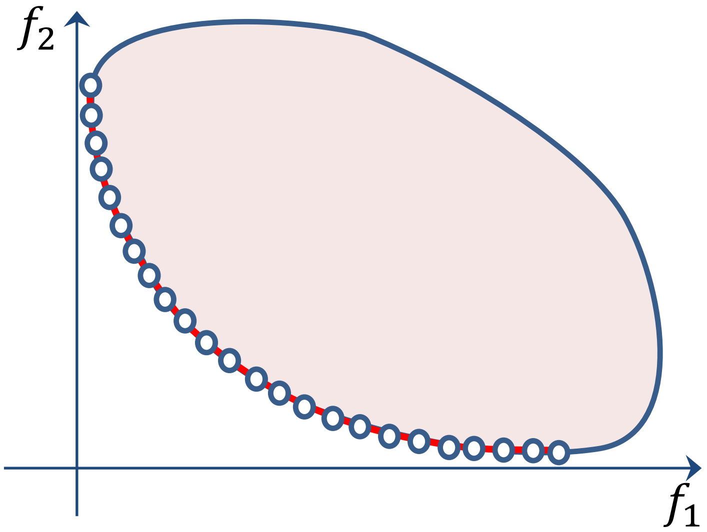
Advantages:

- Can get a set of Pareto solutions in a single run
- Less sensitive to the Pareto front shape and continuity
- Individuals in the population can help each other – search more effectively



Performance Indicators

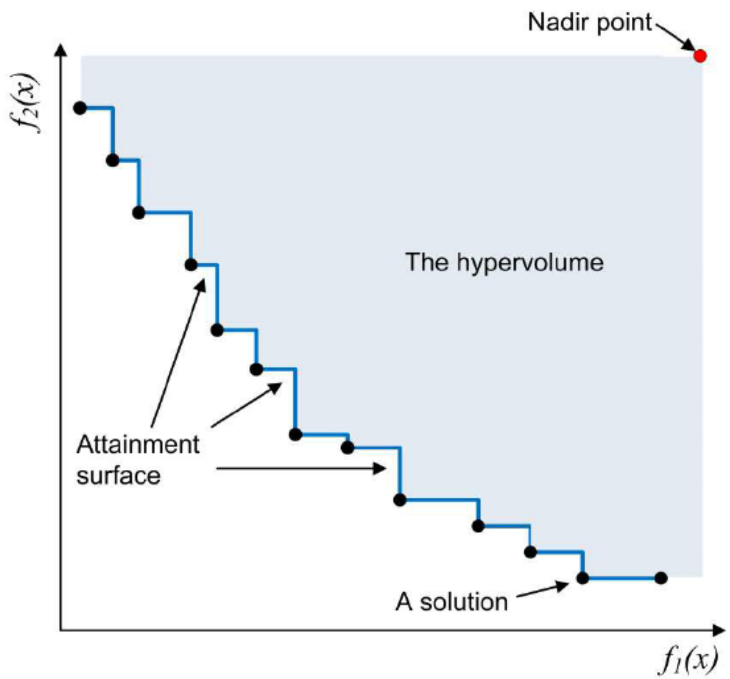
- How well a solution set approximates the Pareto front? (We normally can not get the true Pareto front in practice)
  - Convergence: How close the solutions are to the Pareto front
  - Diversity: How uniform the solutions are distributed
  - Spread/Coverage: How well the solutions cover the Pareto front
- Ideally, we aim to get a solution set that locates on the Pareto front, and uniformly covers the entire Pareto front
  - No possible in practice, need MANY individuals



Performance Indicators

- **Hypervolume**: the area dominated by the solution set. The larger the better. A very widely used indicator

- **Step 1**: Get the boundaries of the true (estimated) Pareto front
- **Step 2**: Normalise the objective values of the solutions into [0, 1]
  - $f_1'(X) = \frac{f_1(X) - f_{1,min}}{f_{1,max} - f_{1,min}}$
  - $f_2'(X) = \frac{f_2(X) - f_{2,min}}{f_{2,max} - f_{2,min}}$
- **Step 3**: Set the nadir/reference point
  - Slightly outside the boundary, e.g., [1.1, 1.1]
- **Step 4**: Calculate the area covered
  - A bit geometry, but packages can do for you



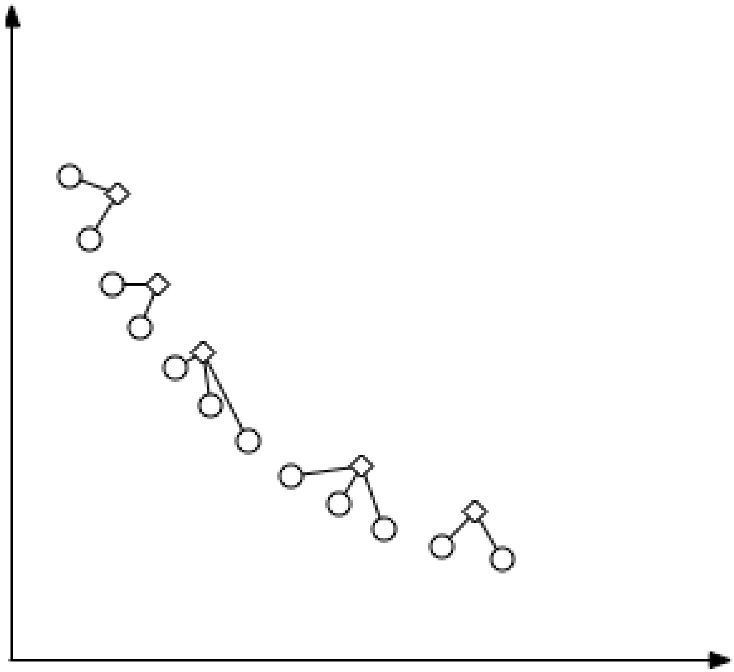
Performance Indicators

- Advantages of hypervolume
  - Consistent to Pareto dominance
    - If set A dominates set B, then A must have a larger hypervolume than B
    - The True Pareto front always has the largest hypervolume
  - NO need to know the true Pareto front
  - Reflects convergence, diversity and spread simultaneously
- Disadvantages of hypervolume
  - Inefficient, time complexity grows exponentially
  - Becomes very time consuming if the number of objectives is large (e.g., > 3)
  - Can be sensitive to the reference point
- Pareto front approximation:
  - Collect ALL the available data (all the compared algorithms, all the runs, …), merge them to get the final non-dominated set

Performance Indicators

- **Inverted Generational Distance (IGD)**: the average distance from the true Pareto front to the solution set
  - The smaller the better. Optimal value = 0

- **Step 0**: Calculate the boundary and normalise the objective values
- **Step 1**: Select a set of uniformly distributed reference points on the true Pareto front
- **Step 2**: For each reference point, find out the closest point in the solution set, and calculate the distance between them
- **Step 3**: Average the distances for all the reference points



Performance Indicators

- Advantages of IGD:
  - Can reflect convergence, uniformity and spread simultaneously
  - Computationally affordable: $O(NM)$, N is the number of reference points, M is the number of non-dominated solutions

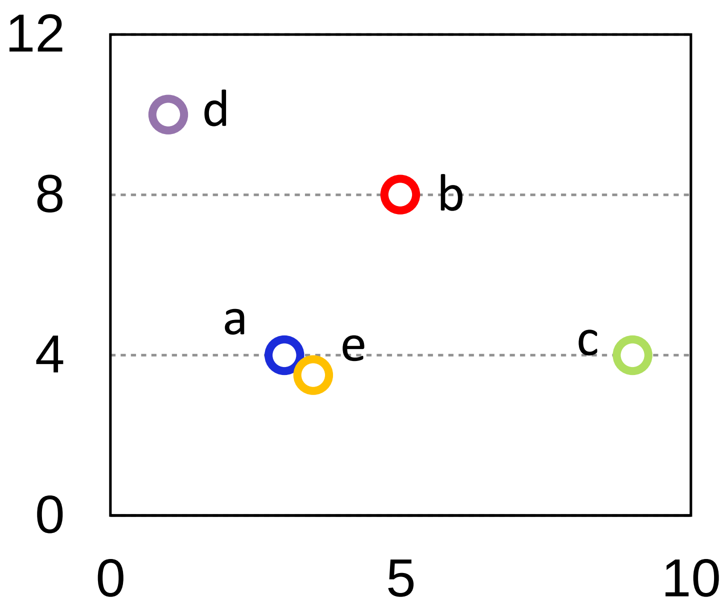
- Disadvantages of IGD:
  - Not consistent with dominance relation (in fact, hypervolume is the ONLY indicator consistent with dominance relation)
  - Need to know the true Pareto front (not practical), use Pareto front approximation

Design Issues in EMO

- **Fitness Assignment**
  - Define a scalar fitness based on multiple objective values
  - Dominance relation
- **Diversity Preservation**
  - Ensure the coverage and uniform distribution of the Pareto front
- **Elitism**
  - Do not lose non-dominated solutions during the search

Design Issues in EMO

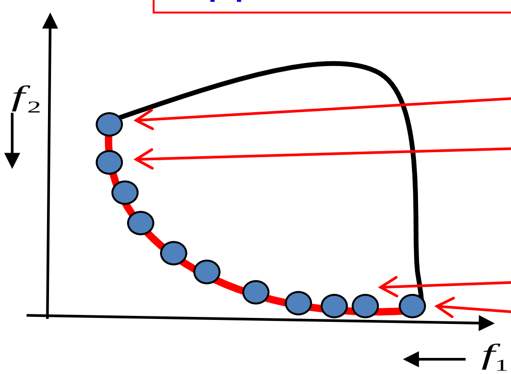
- **Fitness Assignment**
  - Not as straightforward as single-objective: objective value = fitness
  - Define a scalar fitness value based on multiple objective values
- Which one has better fitness? a, b, c, d, e?
  - d seems good
  - a and e are too close
  - c is dominated by e, but potential
  - b is bad
  - Which two will survive?



Design Issues in EMO

- Criteria-based methods
  - Divide the population into subpopulations, each optimise one objective separately
- Domination-based methods
  - If A dominates B, then A has a better fitness than B
- Decomposition-based methods
  - Decompose into sub-problems, each searching for a region/point of the Pareto front

Approximation of the PF $\Rightarrow$ N single obj opt suproblems



$$\min g^{ws}(x \mid \lambda^1) = 1 \times f_1(x) + 0 \times f_2(x) \qquad \lambda^1 = (1, 0)$$

$$\min g^{ws}(x \mid \lambda^2) = 0.9 \times f_1(x) + 0.1 \times f_2(x) \qquad \lambda^2 = (0.9, 0.1)$$

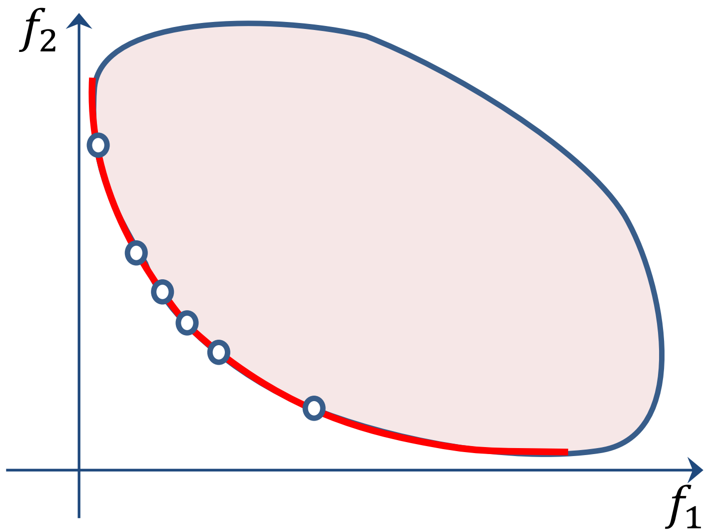
$$\vdots$$

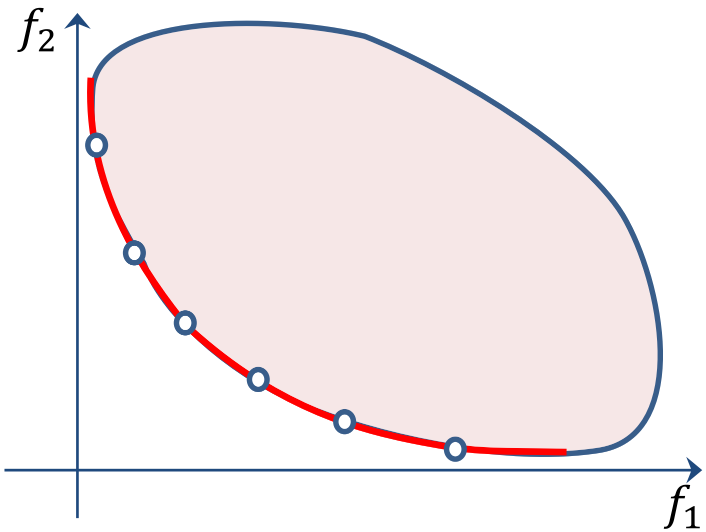
$$\min g^{ws}(x \mid \lambda^{10}) = 0.1 \times f_1(x) + 0.9 \times f_2(x) \qquad \lambda^{10} = (0.1, 0.9)$$

$$\min g^{ws}(x \mid \lambda^{11}) = 0 \times f_1(x) + 1 \times f_2(x) \qquad \lambda^{11} = (0, 1)$$

Design Issues in EMO

- **Diversity Preservation**
  - A single optimum is not enough, as we need to approximate the entire Pareto front
  - Use much fewer solutions to approximate: coverage and uniformity
- Which distribution is better?

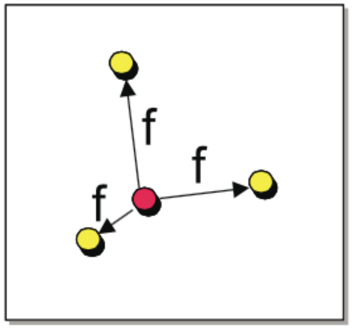




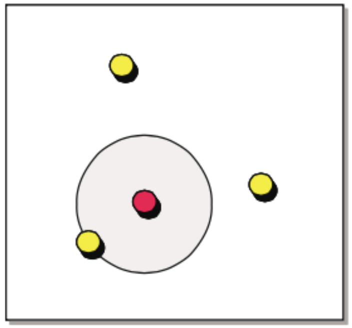
Design Issues in EMO

- **Diversity Preservation**
  - Penalise by neighbouring individuals
  - Larger penalty if more individuals in the population closer to it
  - Limit the number of individuals in a small region/grid
  - Crowding distance (NSGA-II)

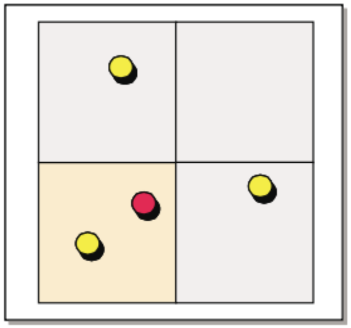
Kernel
MOGA



Nearest neighbor
SPEA2



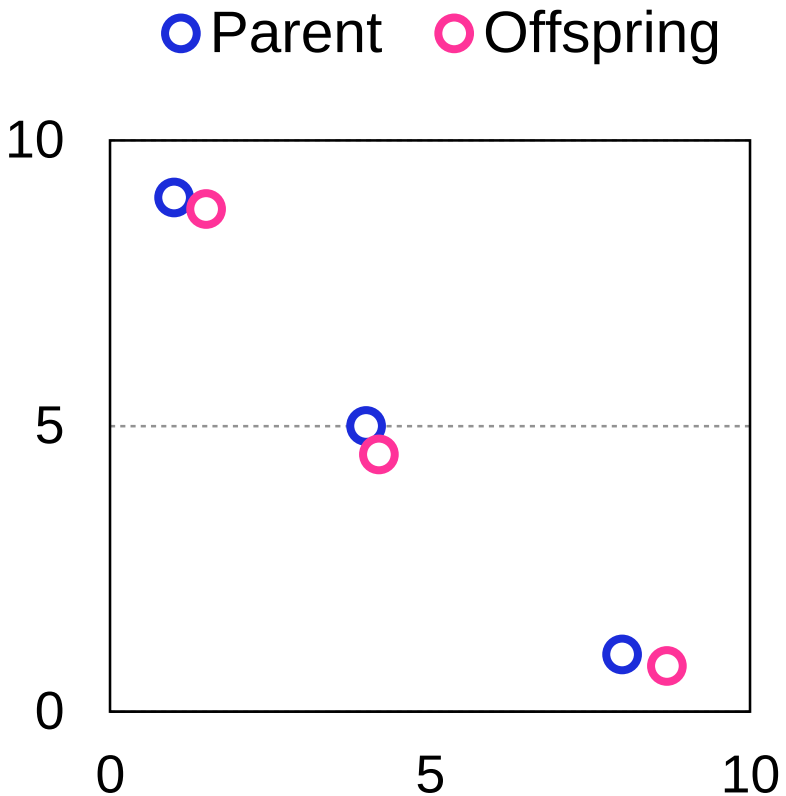
Histogram
PAES



Design Issues in EMO

- **Elitism**
  - Should the offspring replace the parents?

- Keep both parents and offspring
  - May need a large population size
- Maintain an external archive



Summary

- EC is a natural powerful technique for multi-objective optimisation
  - Population-based search, can get a set in a single run
  - Individuals can help each other
- MO Performance is a key issue (convergence, diversity, spread)
  - HV
  - IGD
- Design issues
  - Fitness assignment
  - Diversity preservation
  - Elitism

- Next lecture: well-known EMO algorithms