Lecture 9 - Multi-Objective Evolutionary Optimisation

Yuhui Shi

CSE, SUSTech

Review of the Last Lecture

- Strategy Games
- Co-evolutionary Learning of Game-playing Strategies
- Theoretical Framework of Generalisation in Co-evolutionary Learning Examples of Generalisation Framework
- Estimating Generalisation in Co-evolutionary Learning

Outline of This Lecture

- Multi-Objective Optimisation and Pareto Dominance
  - ✓ Multi-Objective Optimisation (MOO)
  - ✓ Pareto Dominance
- Multi-Objective Evolutionary Algorithms (MOEAs)
  - ✓ Introduction to MOEAs
  - ✓ Non-dominated Sorting GA (NSGA II)
- From Multi- to Many Objective Optimisation
  - ✓ Many Objective Optimisation
  - ✓ Two Arch2
- Reading List

Outline of This Lecture

- **Multi-Objective Optimisation and Pareto Dominance**
  - ✓**Multi-Objective Optimisation (MOO)**
  - ✓**Pareto Dominance**
- Multi-Objective Evolutionary Algorithms (MOEAs)
  - ✓Introduction to MOEAs
  - ✓Non-dominated Sorting GA (NSGA II)
- From Multi- to Many Objective Optimisation
  - ✓Many Objective Optimisation
  - ✓Two Arch2
- Reading List

Multi-Objective Optimisation (MOO)

Compared to “optimisation” that we have seen previously:

- ✓More than one objective to be optimised,
- ✓with or without constrains.

$$\min/\max F(x) = (f_1(x), f_2(x), \dots, f_m(x))$$
$$\text{s.t. } g_j(x) \geq 0,\ j = 1, 2, \dots, J$$
$$h_k(x) = 0,\ k = 1, 2, \dots, K$$
$$x^{(L)}_i \leq x_i \leq x^{(U)}_i,\ i = 1, 2, \dots, I$$

where

- ✓x is a vector of continuous, discrete or mixed variables.
- ✓“s.t.” stands for “subject to”.
- ✓m is the number of objectives.
- ✓$x^{(L)}_i$ and $x^{(U)}_i$ refer to the lower bound and upper bound of x_i, respectively.

Pareto (帕雷托) Dominance

- x_a dominates x_b if
  - ✓Solution x_a is no worse than x_b in all objectives.
  - ✓Solution x_a is strictly better than x_b in at least one objective.
  - ✓Denoted as $x_a \preceq x_b$ if minimisation.
- x_a dominates x_b $\Leftrightarrow$ x_b is dominated by x_a

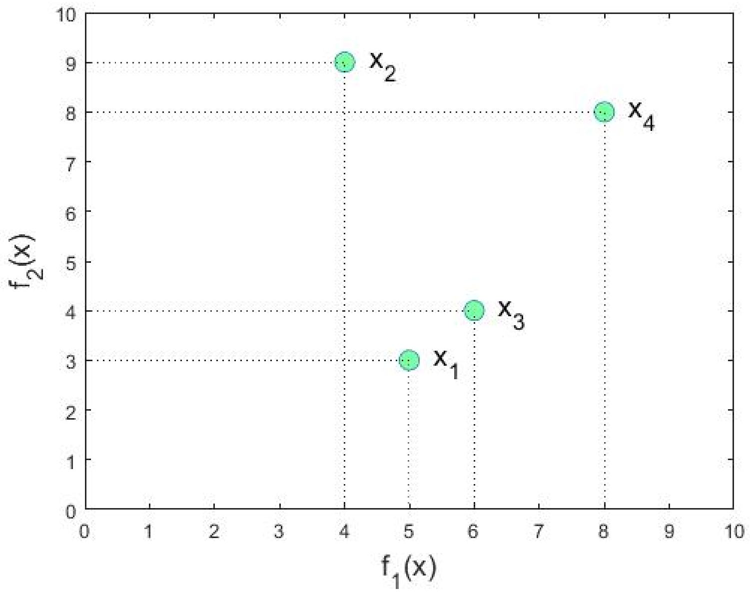


Figure 1: Example: minimise $F(x) = (f_1(x), f_2(x))$.

Pareto Front

- Among a set of solutions P, the non-dominated solution set is a set of solutions that are not dominated by any member of P.
- The non-dominated set of the entire feasible decision space is called the Pareto-optimal set.
- The boundary defined by the set of all points mapped from the Pareto optimal set to objective space is called the Pareto optimal front.

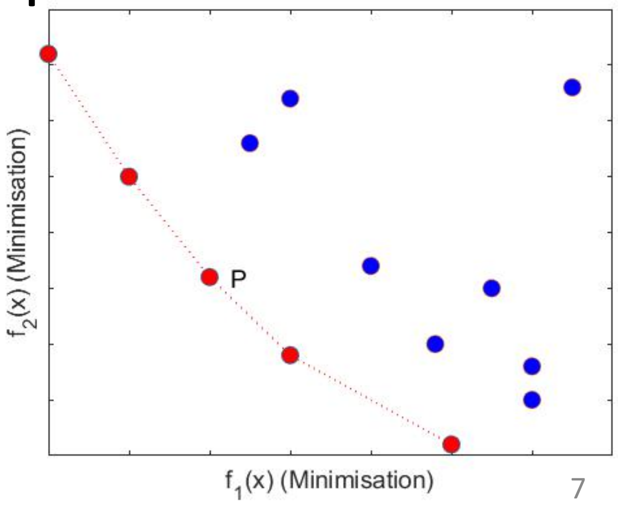


Figure 2: Pareto optimal: red points. Pareto optimal front: dashed red curve.

Pareto Optimal Solutions

- Pareto optimal set in the decision space (决策空间).
- Pareto optimal front in the objective space (目标空间).

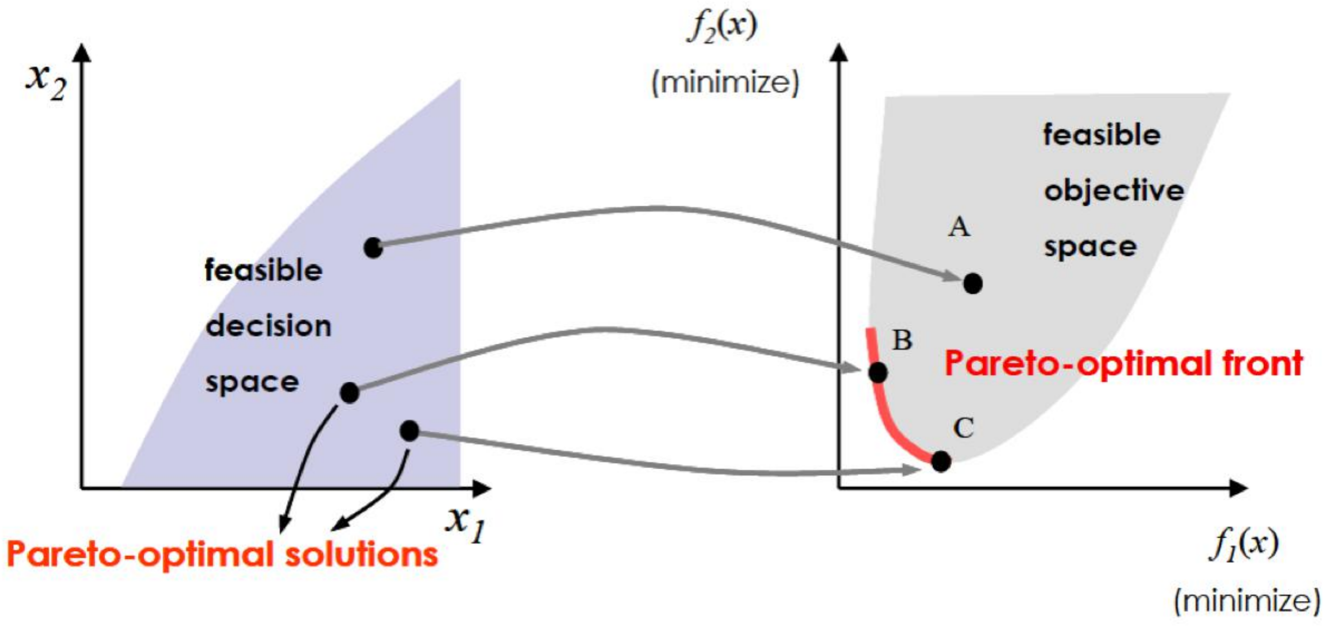


Figure 3: Image source: “Multi-Objective Optimization” by K. Deb.

Main Goals of MOO

- To find a set of solutions as close as possible to the Pareto optimal front.

  → Convergence (收敛性).
- To find a set of solutions as diverse as possible.

  → Diversity (多样性).

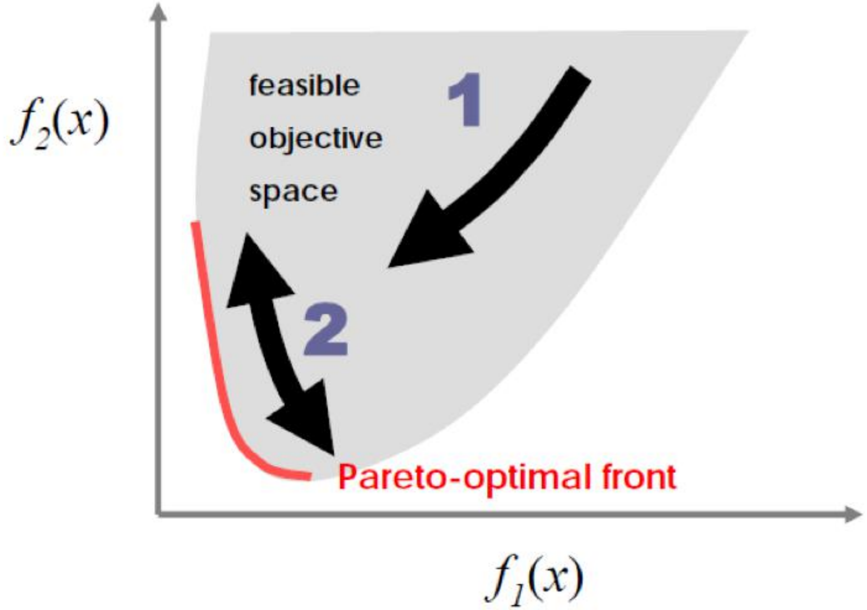


Figure 4: Image source: “Multi-Objective Optimization” by K. Deb.

Pareto Dominance Relation

- Reflexive?

  → No. Any solution x does not dominate itself.
- Symmetric?
- → No. $\mathbf{x}_a \preceq \mathbf{x}_b \not\Rightarrow \mathbf{x}_b \preceq \mathbf{x}_a$
- Antisymmetric?[1]

  → No.
- Transitive?

  → Yes. **If** $\mathrm{x}_a \preceq \mathrm{x}_b$ **and** $\mathrm{x}_b \preceq \mathrm{x}_c$ **then** $\mathrm{x}_a \preceq \mathrm{x}_c$**.**

1. $\mathbf{x}_a \not\preceq \mathbf{x}_b \not\Rightarrow \mathbf{x}_b \preceq \mathbf{x}_a$

How to Solve a MOO Problem?

- Provide one solution:
  - ✓Straightforward solution: Convert it to a single-objective problem. E.g., The weighted sum approach.
- Provide several solutions:
  - ✓"Approach" (逼近) the solutions to the Pareto front, then select a solution from the set.
    - ➢Non-trivial, depends on the decision maker's experience.
  - ✓A decision maker selects an area of solutions, then apply local search.
    - ➢Non-trivial, depends on the decision maker's experience.

1.Convert to A Single-objective Problem

- It's straightforward:
  - ✓Build a single objective using a weighted sum of objectives:

$$\textit{Combined objective} = \alpha f_1 + (1 - \alpha) f_2$$

  - ✓It seems to be a very simple method!

Questions

1. What the value α should be?
2. If you don't know the exact value, how to decide/compute the value of α?

1.Convert to A Single-objective Problem
-*Weaknesses*

- We don't know the exact weights in many cases. Though there are various methods for computing the weights, they also have weakness:
  - ✓Rely on the assumption of convexity/differentiability.
  - ✓Require knowledge of bounds of the objective values.
  - ✓The solution highly depends on the choice of weights.
  - ⇒ Search in the solution space involves search in the weight space.
- We get only one solution given a set of weights.
  - ✓Unable to provide different trade-off to the decision maker.
  - ✓We don't really know other possible trade-off among objectives.

2.Provide Several Solutions

- Multi-Objective Evolutionary Algorithms.
- Many-Objective Evolutionary Algorithms.

Outline of This Lecture

- Multi-Objective Optimisation and Pareto Dominance
  - ✓Multi-Objective Optimisation (MOO)
  - ✓Pareto Dominance
- **Multi-Objective Evolutionary Algorithms (MOEAs)**
  - ✓**Introduction to MOEAs**
  - ✓**Non-dominated Sorting GA (NSGA II)**
- From Multi- to Many Objective Optimisation
  - ✓Many Objective Optimisation
  - ✓Two Arch2
- Reading List

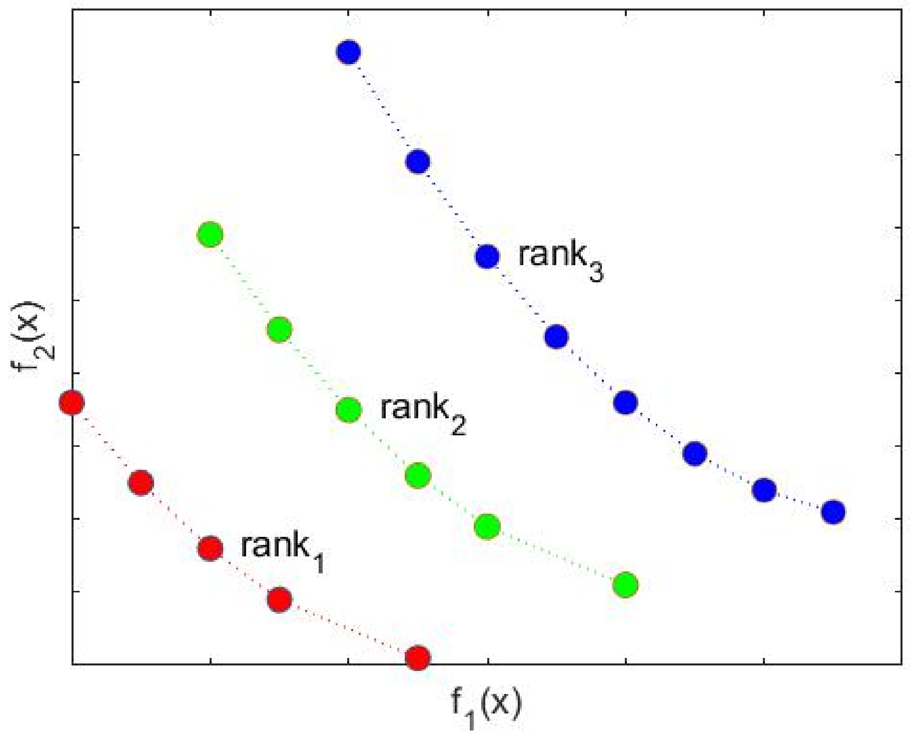
Advantages of MOEAs

- They can provide a set of non-dominated solutions in a single run without requiring the set of weights.
- They do not require the objective functions to be convex, smooth, or even continuous (fewer assumptions).
- They can handle nonlinear constraints.
- They can deal with uncertainty and dynamics better than others.

Key Ingredient of NSGA II: Non-dominated Sorting [2]

Classify the solutions into a number of mutually exclusive non-dominated sets.

$$F = \bigcup_{i=1}^{3} rank_i$$



[2] Kalyanmoy Deb et al. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: IEEE transactions on evolutionary computation 6.2 (2002), pp. 182–197

Crowding Distance

- Determine crowding distance.
- Denotes half of the perimeter of the enclosing cuboid with the nearest neighbouring solutions in the same front.
- Estimation of the largest cuboid enclosing a particular solution (density estimation).

normalized m^{th} side distance:

$$\mathrm{d}_i^m = \frac{d^m_{(i+1)(i-1)}}{f^m_{\max} - f^m_{\min}}$$

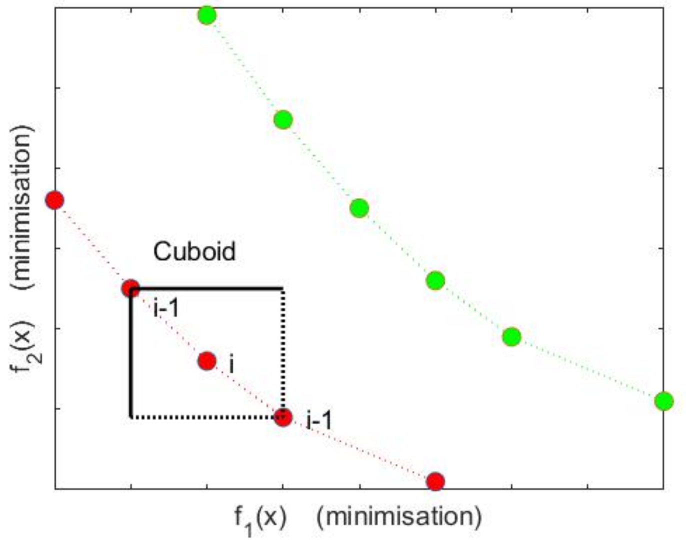
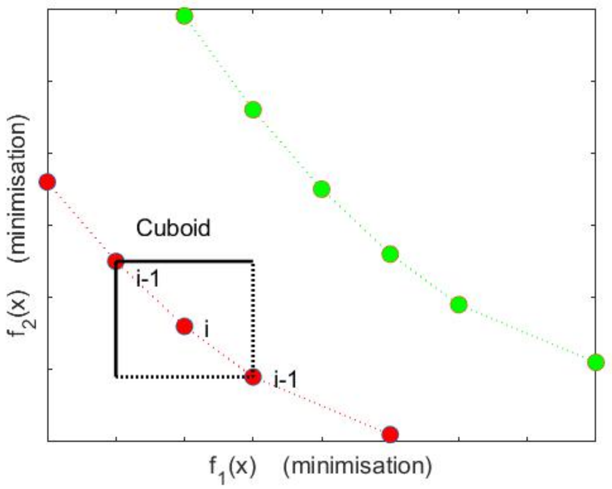


Figure 6: The crowding distance of the ith solution in its front (red) is the average side-length of the cuboid (box) (or half of the perimeter of the enclosing cuboid with the nearest neighbouring solutions in the same front).

Comparing Solutions

- Crowding tournament selection
  - ✓Assume that every solution has a non-domination rank and a local crowding distance.
  - ✓A solution x_a wins a tournament against another solution x_b:
    - ➢If the solution x_a has a better rank.
    - ➢If they have the same rank but solution x_a has a larger crowing distance than solution x_b.



Non-dominated Sorting GA

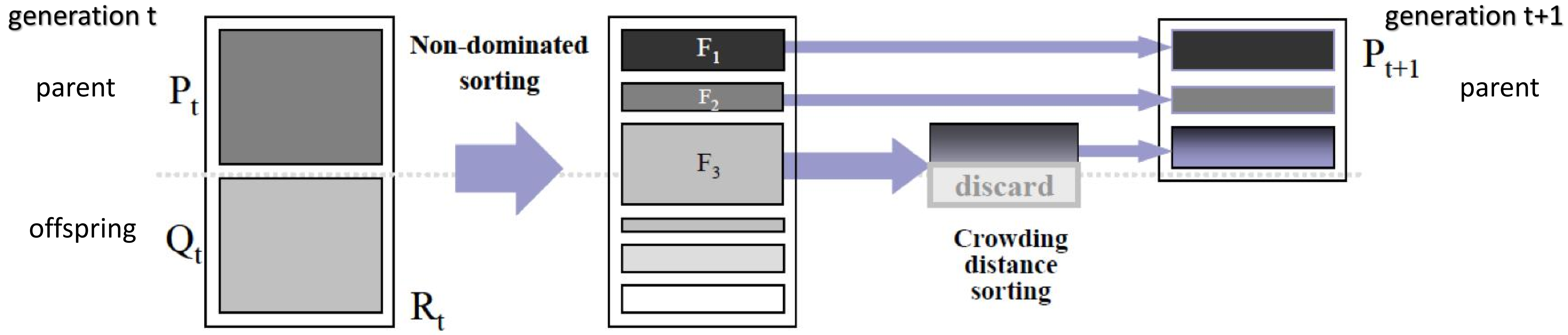


Figure 5: Image source: “Multi-Objective Optimization” by K. Deb.

[2] Kalyanmoy Deb et al. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: IEEE transactions on evolutionary computation 6.2 (2002), pp. 182–197

Elitist Non-Dominated Sorting GA (NSGA II)

- Step 1
  - ✓Create offspring population Q_t from P_t by using the crowded tournament selection, crossover and mutation operators
- Step 2
  - ✓Combine parent P_t and offspring Q_t populations $R_t = P_t \cup Q_t$
  - ✓Perform a non-dominated sorting to R_t and find different fronts F_i
- Step 3
  - ✓Set new population $P_{t+1} = \varnothing$ and set $i = 1$
  - ✓Until $|P_{t+1}| + |F_i| < N$, perform $P_{t+1} = P_{t+1} \cup F_i$ and increase i
- Step 4
  - ✓Include the most widely spread solutions $(N-|P_{t+1}|)$ of F_i in P_{t+1} using the crowding distance values

Non-dominated Sorting GA [2]

- Advantages
  - ✓The diversity among non-dominated solutions is maintained using the crowding procedure: No extra diversity control is needed.
  - ✓Elitism protects an already found Pareto-optimal solution from being deleted.
- Disadvantages
  - ✓When there are more than N members in the first non-dominated set, some Pareto-optimal solutions may give their places to other non-Pareto-optimal solutions.

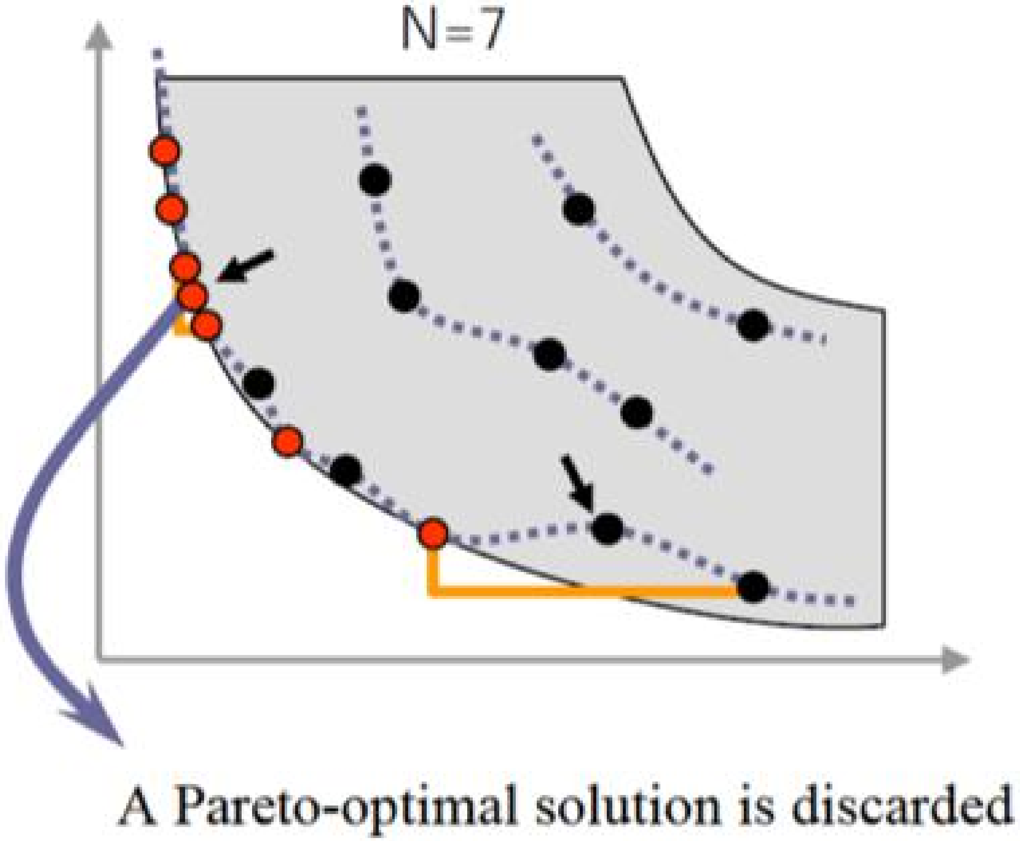


Figure 7: Image source: “Multi-Objective Optimization” by K. Deb.

Outline of This Lecture

- Multi-Objective Optimisation and Pareto Dominance
  - ✓Multi-Objective Optimisation (MOO)
  - ✓Pareto Dominance
- Multi-Objective Evolutionary Algorithms (MOEAs)
  - ✓Introduction to MOEAs
  - ✓Non-dominated Sorting GA (NSGA II)
- **From Multi- to Many Objective Optimisation**
  - ✓**Many Objective Optimisation**
  - ✓**Two Arch2**
- Reading List

Many Objective Optimisation

- Almost all MOEAs break down when the number of objectives is more than 3 (*so many solutions are non-dominated solutions, i.e., Pareto dominance is ineffective on many objective optimisation problems*).
- New techniques and algorithms are needed in handling such a larger number of objectives.
- In order to highlight the challenges of many objectives, a new term is coined

  → many objective optimisation.

Improved Two-Archive Algorithm: Two Arch2 [4, 3] -*Main idea*

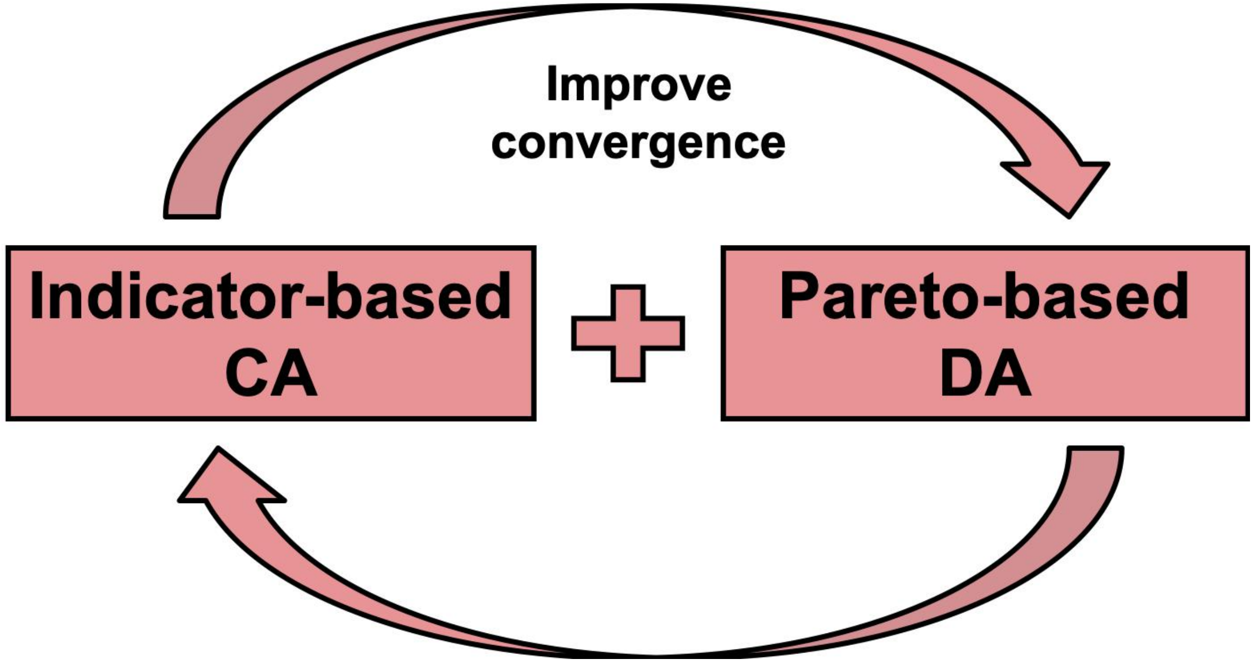


Figure 8: CA = Convergence Archive; DA = Diversity Archive.

[4] Handing Wang, Licheng Jiao, and Xin Yao. "Two Arch2: An improved two-archive algorithm for many-objective optimization". In: IEEE Transactions on Evolutionary Computation 19.4 (2015), pp. 524–541
[3] Zhenshou Song et al. "A Kriging-Assisted Two-Archive Evolutionary Algorithm for Expensive Many-Objective Optimization". In: IEEE Transactions on Evolutionary Computation 25.6 (2021), pp. 1013–1027. DOI: 10.1109/TEVC.2021.3073648

Two Arch2: Main Steps [4]

1. Initialisation.
2. Output DA if the stopping criterion is met, otherwise continue.
3. Generate new solutions from CA and DA by crossover and mutation.
4. Update CA and DA separately, go to 2.

Convergence Archive (CA)

- The quality indicator $I_{\varepsilon+}$ used in Indicator-Based EA (IBEA) is used in selection of CA. $I_{\varepsilon+}$ is an indicator that describes the minimum distance that one solution needs to dominate another solution in the objective space

$$I_{\varepsilon+}(\mathbf{x}_1, \mathbf{x}_2) = \min_{\varepsilon} \left(f_i(\mathbf{x}_1) - \varepsilon \leq f_i(\mathbf{x}_2), 1 \leq i \leq m \right),$$

  where m is the number of objectives.

- The fitness is assigned as below, the solution with the smallest fitness is removed from CA first.

$$F(\mathbf{x}_1) = \sum_{\mathbf{x}_2 \in Population/\{\mathbf{x}_1\}} -e^{-I_{\varepsilon+}(\mathbf{x}_2, \mathbf{x}_1)/0.05}$$

Diversity Archive (DA)

- Update DA:
  - ✓ When DA overflows, boundary solutions (solutions with maximal or minimal objective values) are firstly selected.
  - ✓ In the iterative process, the most different solution from the current DA is added until reaching the size.
- L_p-norm distance is adopted as the similarity measure in DA.
- DA is used as the final output of Two Arch2.

Degraded Euclidean Distance in High-Dimensional Space

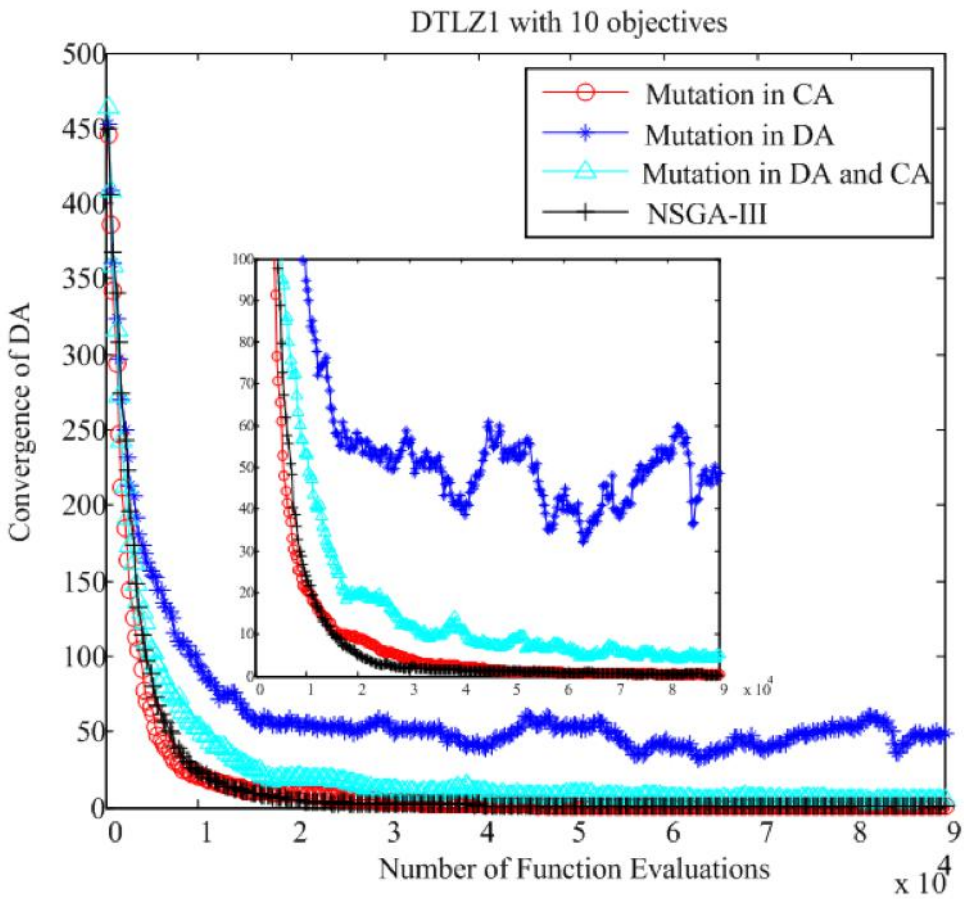
- The Euclidean distance (L_2-norm) degrades its similarity indexing performance in a high-dimensional space (distance concentration).
- Most of existing diversity maintenance methods use the Euclidean distance to measure similarity among solutions for ManyOPs.

Similarity in High-Dimensional Space

- The fractional distances (L_p-norm, $p < 1$) perform better in a high-dimensional space (experimantal results shown in Fig.3 [4]).
- $L_{1/m}$-norm is employed in Two Arch2, where m is the number of objectives.

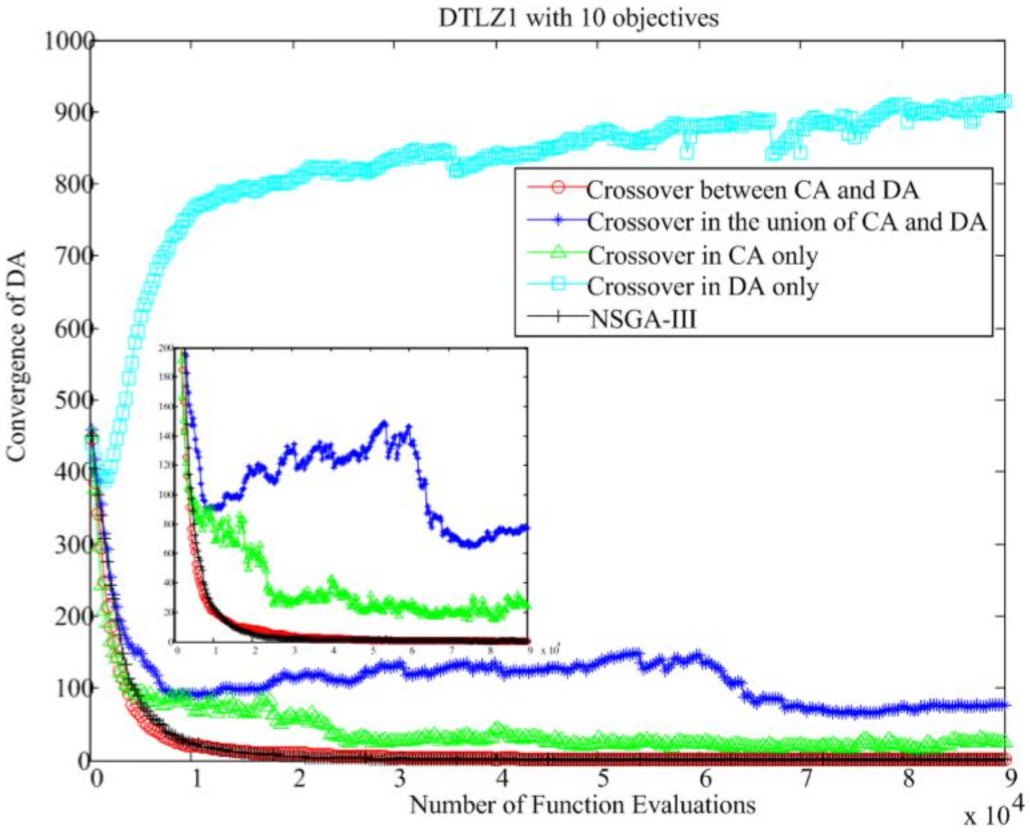
Interaction between CA and DA: Mutation [4]

- Mutation to DA does not speed up convergence, and disturbs the guidance of CA to DA.
- Mutation is applied to CA only in Two Arch2.
- CA leads convergence.



Interaction between CA and DA: Crossover [4]

- The crossover between CA and DA is employed in Two Arch2.



Outline of This Lecture

- Multi-Objective Optimisation and Pareto Dominance
  - ✓Multi-Objective Optimisation (MOO)
  - ✓Pareto Dominance
- Multi-Objective Evolutionary Algorithms (MOEAs)
  - ✓Introduction to MOEAs
  - ✓Non-dominated Sorting GA (NSGA II)
- From Multi- to Many Objective Optimisation
  - ✓Many Objective Optimisation
  - ✓Two Arch2
- **Reading List**

References for This Lecture I

1. Urvesh Bhowan et al. “Reusing genetic programming for ensemble selection in classification of unbalanced data”. In: IEEE Transactions on Evolutionary Computation 18.6 (2014), pp. 893–908.
2. Kalyanmoy Deb et al. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: IEEE transactions on evolutionary computation 6.2 (2002), pp. 182–197.
3. Zhenshou Song et al. “A Kriging-Assisted Two-Archive Evolutionary Algorithm for Expensive Many-Objective Optimization”. In: IEEE Transactions on Evolutionary Computation 25.6 (2021), pp. 1013–1027. doi: 10.1109/TEVC.2021.3073648.
4. Handing Wang, Licheng Jiao, and Xin Yao. “Two Arch2: An improved two-archive algorithm for many-objective optimization”. In: IEEE Transactions on Evolutionary Computation 19.4 (2015), pp. 524–541.
5. Xin Yao and Yong Liu. “Making use of population information in evolutionary artificial neural networks”. In: IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 28.3 (1998), pp. 417–425.

Reading List for Next Lecture

1. T. B¨ack, D. B. Fogel, and Z. Michalewicz (eds.), Handbook of Evolutionary Computation, IOP Publ. Co. & Oxford University Press, 1997. Sections C5.1 - C5.6.
2. T. P. Runarsson and X. Yao, “Stochastic Ranking for Constrained Evolutionary Optimization,” IEEE Transactions on Evolutionary Computation, 4(3):284-294, September 2000.
3. T. Runarsson and X. Yao, “Search Bias in Constrained Evolutionary Optimization,” IEEE Transactions on Systems, Man, and ybernetics, Part C, 35(2):233-243, May 2005.