Lecture 8: Multi-Objective Evolutionary Optimisation

CSE5012: Evolutionary Computation and Its Applications

Xin Yao

CSE, SUSTech

17 April 2023

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 (MOO)

- **Compared to "optimisation" that we have seen previously:**
  - **More than one objective to be optimised,**
  - **with or without constrains.**

$$
\begin{aligned}
\min / \max \quad & F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})) \\
s.t. \quad & g_j(\mathbf{x}) \geq 0, j = 1, 2, \ldots, J \\
& h_k(\mathbf{x}) = 0, k = 1, 2, \ldots, K \\
& x_i^{(L)} \leq x_i \leq x_i^{(U)}, i = 1, 2, \ldots, I
\end{aligned}
$$

where

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

Pareto (帕雷托) Dominance

- $\mathbf{x}_a$ **dominates** $\mathbf{x}_b$ **if**
  - **Solution $\mathbf{x}_a$ is no worse than $\mathbf{x}_b$ in all objectives.**
  - **Solution $\mathbf{x}_a$ is strictly better than $\mathbf{x}_b$ in at least one objective.**
  - **Denoted as $\mathbf{x}_a \preceq \mathbf{x}_b$ if minimisation.**
- **$\mathbf{x}_a$ dominated $\mathbf{x}_b \Leftrightarrow \mathbf{x}_b$ is dominated by $\mathbf{x}_a$**

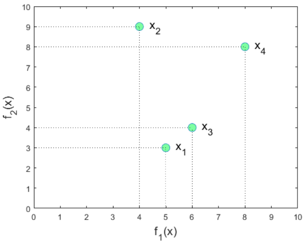


Figure 1: Example: minimise $F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))$.

Pareto Front

- **Among a set of solutions $\mathcal{P}$, the non-dominated solution set is a set of solutions that are not dominated by any member of $\mathcal{P}$.**
- **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 is called the Pareto optimal front.**

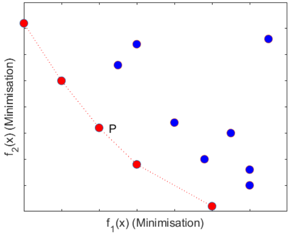


Figure 2: Pareto optimal: red points. Pareto optimal front: dashed red curve.

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).**

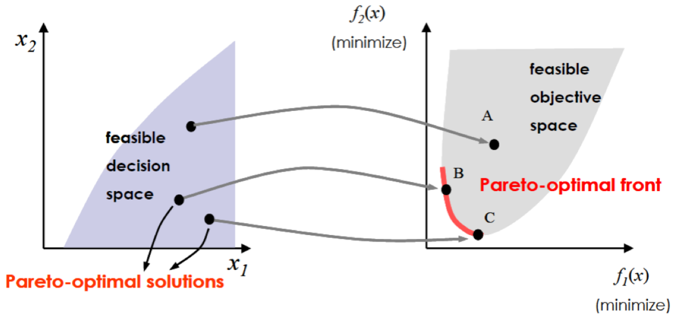


Figure 3: Image source: "Multi-Objective Optimization" by K. Deb.

Main Goals of MOO

1. **To find a set of solutions as close as possible to the Pareto optimal front.**
   $\rightarrow$ **Convergence (收敛性).**
2. **To find a set of solutions as diverse as possible.**
   $\rightarrow$ **Diversity (多样性).**

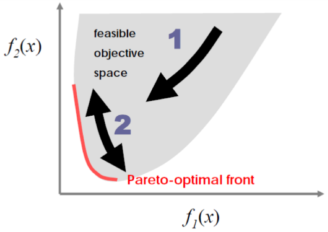


Figure 4: Image source: "Multi-Objective Optimization" by K. Deb.

Pareto Dominance Relation

- **Reflexive?**
  $\rightarrow$ **No. Any solution $\mathbf{x}$ does not dominate itself.**
- **Symmetric?**
  $\rightarrow$ **No.** $\mathbf{x}_a \preceq \mathbf{x}_b \nRightarrow \mathbf{x}_b \preceq \mathbf{x}_a$
- **Antisymmetric?**[1]
  $\rightarrow$ **No.**
- **Transitive?**
  $\rightarrow$ **Yes. If $\mathbf{x}_a \preceq \mathbf{x}_b$ and $\mathbf{x}_b \preceq \mathbf{x}_c$ then $\mathbf{x}_a \preceq \mathbf{x}_c$.**
- $\mathbf{x}_a \npreceq \mathbf{x}_b \nRightarrow \mathbf{x}_b \preceq \mathbf{x}_a$

[1] Antisymmetric Relation: A binary relation R is antisymmetric iff: If $R(a, b)$ and $R(b, a)$ then $a = b$.

How to Solve a MOO Problem?

1. **Provide one solution:**
   - **Straightforward solution: Convert it to a single-objective problem. E.g., The weighted sum approach.**

2. **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

Main idea

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

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

  - **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.**

  $\Rightarrow$ **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.**

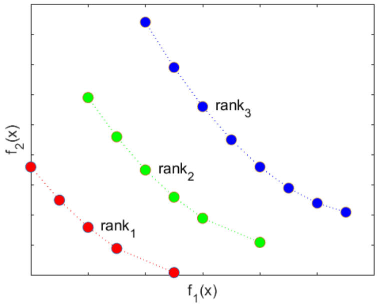
Outline of This Lecture

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 [4]

- **Classify the solutions into a number of mutually exclusive non-dominated sets.**
- $F = \cup_{i=1}^{3} rank_i$



[4] 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

Non-dominated Sorting GA

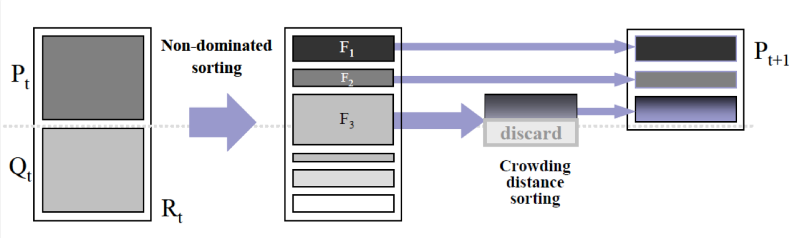


Figure 5: Image source: "Multi-Objective Optimization" by K. Deb.

[4] 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).**

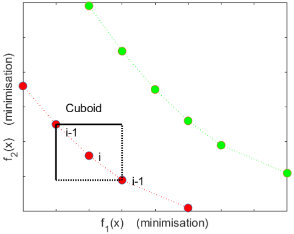
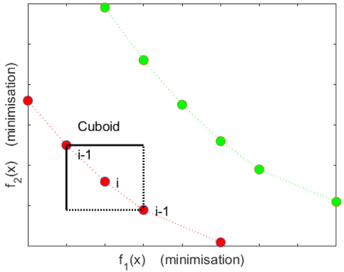


Figure 6: The crowding distance of the i^{th} solution in its front (red) is the average side-length of the cuboid (box).

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 [4]

$R_t = P_t \cup Q_t$	combine parent and children population
$\mathcal{F} = \texttt{fast-nondominated-sort}(R_t)$	$\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \ldots)$, all non-dominated fronts of R_t
until $\lvert P_{t+1} \rvert < N$	till the parent population is filled
$\quad \texttt{crowding-distance-assignment}(\mathcal{F}_i)$	calculate crowding distance in $\mathcal{F}_i$
$\quad P_{t+1} = P_{t+1} \cup \mathcal{F}_i$	include i-th non-dominated front in the parent pop
Sort$(P_{t+1}, \geq_n)$	sort in descending order using $\geq_n$
$P_{t+1} = P_{t+1}[0 : N]$	choose the first N elements of P_{t+1}
$Q_{t+1} = \texttt{make-new-pop}(P_{t+1})$	use selection,crossover and mutation to create a new population Q_{t+1}
$t = t + 1$	

Screenshot of [4] 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

Non-dominated Sorting GA [4]

- **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.**

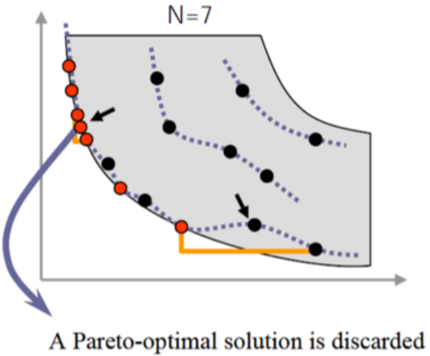


Figure 7: Image source: "Multi-Objective Optimization" by K. Deb.

Outline of This Lecture

Many Objective Optimisation

- **Almost all MOEAs break down when the number of objectives is more than 3.**
- **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 $\rightarrow$ many objective optimisation.**

Improved Two-Archive Algorithm: Two_Arch2 [6, 5]

Main idea

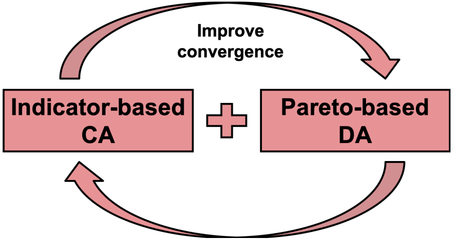


Figure 8: CA = Convergence Archive; DA = Diversity Archive.

[6] 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] 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 [6]

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 (IDEA) 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

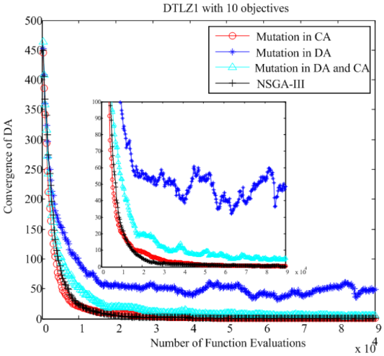
- **The Euclidean distance (L_2-norm) degrades its similarity indexing performance in a high-dimensional space (distance concentration).**
- **Most of existing diversity maintenance methods use the Euclidean distance to measure similarity among solutions for MaOPs.**

Similarity in High-Dimensional Space

- **The fractional distances (L_p-norm, $p < 1$) perform better in a high-dimensional space. Why?**
- **$L_{1/m}$-norm is employed in Two_Arch2, where m is the number of objectives.**

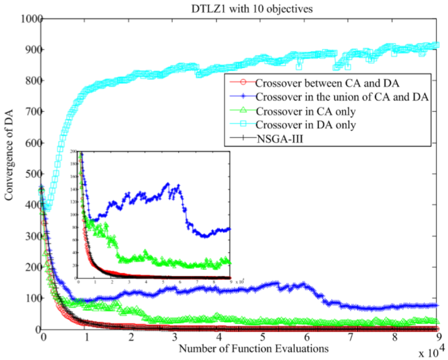
Interaction between CA and DA: Mutation [6]

- **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 [6]

- **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

Multi-Objective Learning
- **Introduction to Multi-Objective Learning**
- **Diverse and Accurate Ensemble Learning Algorithm**
- **Class Imbalance Learning**

Reading Lists

An Example Error Function

- **Negative correlation learning defines a simple error function for each network i as follows (N is the size of training set):**

$$\varepsilon_i = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{2}(f_i(\mathbf{x}_n) - y_n)^2 + \lambda p_i(\mathbf{x}_n)\right),$$

where $p_i(\mathbf{x}_n) = (f_i(\mathbf{x}_n) - F(\mathbf{x}_n))\sum_{j\neq i}(f_j(\mathbf{x}_n) - F(\mathbf{x}_n))$.

Where Are Multiple Objectives?

- **There are many methods for learning diverse and accurate ensembles, e.g., boosting, bagging, negative correlation learning, etc.**
- **In general: 1/Error = Accuracy + λDiversity**
- **We would like to maximise both the accuracy of each individual learners and the diversity among individuals.**
- **These are in essence two separate criteria/objectives.**

Multi-objective Learning [2]

- **Multi-objective learning treats accuracy and diversity as two separate but key objectives in learning.**
- **Multi-objective optimisation algorithms, such as multi-objective evolutionary algorithms (MOEAs), are used as learning algorithms.**
- **The result from such an MOEA is a non-dominated set of solutions (i.e., learners), which ideally form the ensemble we are interested.**

[2] Arjun Chandra and Xin Yao. "Ensemble learning using multi-objective evolutionary algorithms". In: *Journal of Mathematical Modelling and Algorithms* 5.4 (2006), pp. 417–445

Flexibility and Generality

- **Multi-objective learning offers a highly flexible and general framework for considering different requirements in learning.**
- **For example, we can include an additional regularisation term, as an additional objective [3]. Thus, three objectives are optimised:**
  1. **objective of performance;**
  2. **objective of diversity;**
  3. **objective of regularisation.**

$$\varepsilon_i = \frac{1}{M}\sum_{n=1}^{N}(f_i(\mathbf{x}_n) - y_n)^2 - \frac{\lambda}{M}\sum_{n=1}^{N}(f_i(\mathbf{x}_n) - f_{ens}(\mathbf{x}_n))^2 + \alpha_i w_i^T w_i$$

- $f_i(\mathbf{x}_n)$ **is the** i^{th} **base learner's output for a training sample** $\mathbf{x}_n$**;**
- y_n **is the desired output (true value) for a training sample** $\mathbf{x}_n$**;**
- $f_{ens}(\mathbf{x}_n)$ **is the ensemble's output for a training sample** $\mathbf{x}_n$**.**

[3] Huanhuan Chen and Xin Yao. "Multiobjective neural network ensembles based on regularized negative correlation learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22.12 (2010), pp. 1738–1751

Diverse and Accurate Ensemble Learning Algorithm (DIVACE) [2]

- **Two objectives:**
  1. **Accuracy:**

$$\max Accuracy_i = -\frac{1}{N}\sum_{n=1}^{N}(f_i(\mathbf{x}_n) - y_n)^2$$

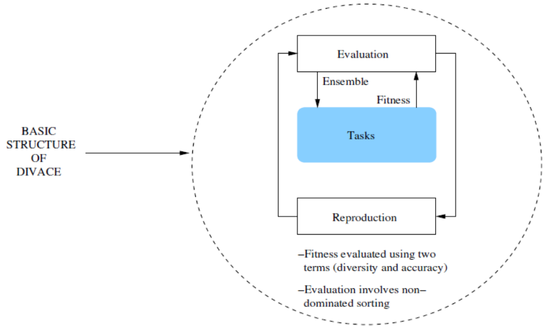
  2. **Diversity:**

$$\max Diversity_i = \sum_{n=1}^{N}(f_i(\mathbf{x}_n) - f_{ens}(\mathbf{x}_n))\left(\sum_{j\neq i, j=1}^{M}(f_j(\mathbf{x}_n) - f_{ens}(\mathbf{x}_n))\right)$$

  - $f_i(\mathrm{x}_n)$ **is the i^{th} base learner's output for a training sample x_n;**
  - y_n **is the desired output (true value) for a training sample x_n;**
  - $f_{ens}(\mathrm{x}_n)$ **is the ensemble's output for a training sample x_n.**

[2] Arjun Chandra and Xin Yao. "Ensemble learning using multi-objective evolutionary algorithms". In: *Journal of Mathematical Modelling and Algorithms* 5.4 (2006), pp. 417–445

DIVACE: Basic Structure [2]



[2] Arjun Chandra and Xin Yao. "Ensemble learning using multi-objective evolutionary algorithms". In: *Journal of Mathematical Modelling and Algorithms* 5.4 (2006), pp. 417–445

DIVACE: Main Steps [2]

1. **Initialise a random population of M networks, initialise the weights to uniformly distributed random values in the range of $(0, 1)$.**
2. **Apply Back-Propagation (BP) to all individuals in the population.**
3. **Repeat the following until stopping condition(s) is(are) met:**
   3.1 **Evaluate the population in accordance with the two objective functions and label the $S_{NonDominated}$ using Non-Dominate Sorting algorithm.**
   3.2 **If $|S_{NonDominated}| < 3$ then a repair rule is used [2].**
   3.3 **Delete all dominated individuals from the population.**
   3.4 **Repeat the following until the population size is M:**
      3.4.1 **Variance update: self-adaptive crossover operator updates itself: $\sigma^2 = max\{2 - \frac{1}{1+exp(anneal\_time - generation)}, 1\}$, where $anneal\_time = 50$ is a parameter signifying exploration time/#generations for which the search process is to be explorative.**
      3.4.2 **Select 3 parents uniformly at random from the population.**
      3.4.3 **Perform crossover (similar to Differential Evolution).**
      3.4.4 **Perform mutation (additive Gaussian noise).**
      3.4.5 **Apply BP to child and add it to the population.**

DIVACE: Main Steps Copied/Pasted from [2] I

Step 1: Create a random initial population[1] (size M) of networks, the weights for each assigned uniformly distributed random values $U(0,1)$.
Step 2: Apply Back-Propagation (BP) to all individuals in the population.
Step 3: Repeat until termination condition (a certain number of generations in our case)

1. Evaluate the individuals in accordance with the two objective functions and label the non-dominated set (Non-dominated sorting algorithm used here.)
2. If the number of non-dominated individuals is less than 3 then a repair rule similar to that used in MPANN (Abbass [5]) is used.
3. All dominated solutions are deleted from the population.

DIVACE: Main Steps Copied/Pasted from [2] II

4. Repeat until population size is M

   – Variance update: updating the variance value for the Gaussian distribution used in crossover. We do it according to,

$$\sigma^2 = 2 - \left(\frac{1}{1 + e^{(\text{anneal\_time} - \text{generation})}} \right) \tag{5}$$

   where anneal_time is a parameter signifying exploration time/ number of generations for which the search process is to be explorative after which the value of σ^2 decreases exponentially to finally reach a fixed value of 1 and it remains 1 until the final iteration. In our experiments, we use a value of 50 for the anneal_time parameter.

[1] For training, we take all the networks in the population as our ensemble but for testing, we only use the final pareto set as the ensemble.

DIVACE: Main Steps Copied/Pasted from [2] III

- Select 3 parents at random from the population. Let α_1 be the main parent and α_2 and α_3 be the supporting parents.

- Perform crossover: Produce a child which has an architecture which is similar to the parents but weights given by,

$$w_{hi} = w_{hi}^{\alpha_1} + N\left(0, \sigma^2\right)\left(w_{hi}^{\alpha_2} - w_{hi}^{\alpha_3}\right) \tag{6}$$

$$w_{oh} = w_{oh}^{\alpha_1} + N\left(0, \sigma^2\right)\left(w_{oh}^{\alpha_2} - w_{oh}^{\alpha_3}\right) \tag{7}$$

- Perform mutation: Mutate the child with probability $1/|pop|$ ($|pop|$ being the size of the population) according to,

$$w_{hi} = w_{hi} + N(0, 0.1) \tag{8}$$

$$w_{oh} = w_{oh} + N(0, 0.1) \tag{9}$$

- Apply BP to child and add it to the population.

Class Imbalance Learning

- **Class imbalance learning** refers to learning from imbalanced data sets, in which some classes of examples (minority) are highly under-represented comparing to other classes (majority).
- **Learning difficulty**: poor generalisation on the minority class.
- **Learning objective**: obtaining a classifier that will provide high accuracy for the minority class without severely jeopardising the accuracy of the majority class.

Multi-class Imbalance Learning [7]

- **Multi-class imbalance: there are more than two classes with uneven class distributions.**
  - **E.g. In software defect prediction: there are different types of defects.**
- **Most existing imbalance learning techniques are only designed for and tested in two-class scenarios.**
- **Existing methods are not effective or even cause a negative effect when there is more than one minority/majority class.**

[7] Shuo Wang and Xin Yao. "Multiclass imbalance problems: Analysis and potential solutions". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42.4 (2012), pp. 1119–1130

Multi-class Imbalance Learning [7]

- **Multi-objective learning treats single class performances as separate objectives.**
- **Multi-objective optimisation algorithms, such as multi-objective evolutionary algorithms (MOEAs), are used as learning algorithms.**
- **The result from such an MOEA is a non-dominated set of solutions (i.e., learners), which ideally form an ensemble we are interested.**

Ensemble Member Selection

- **Sometimes it is unnecessary to include the entire set of classifiers found by MOEAs in an ensemble. A subset would be sufficient, or even better [8].**
- **There are various methods in the literature for selecting a diverse subset of classifiers from a large set [1].**

[8] 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
[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

Software Effort Estimation (SEE)

- **Problem description:**
  - **Estimation of the effort required to develop a software project (e.g., in person-hours).**
    $\leftarrow$ Very useful to you for succeeding in CSE!
  - **Based on features such as**
    - **functional size (numerical),**
    - **required reliability (ordinal),**
    - **programming language (categorical),**
    - **development type (categorical),**
    - **team expertise (ordinal), etc.**
- **Importance:**
  - **Main factor influencing project cost.**
  - **Overestimation vs. underestimation.**

Machine Learning in SEE

- **Uses completed projects as training examples for creating SEE models, e.g.,**
  - **Multi-Layer Perceptrons (MLPs).**
  - **Radial Basis Function networks (RBFs).**
  - **Regression Trees (RTs).**

- **Can be used as decision support tools.**

Different Performance Measures in SEE

- **Mean Magnitude of the Relative Error (MMRE):**

$$MMRE = \frac{1}{T}\sum_{i=1}^{T} MRE_i,$$

where $MRE_i = \frac{|\hat{y}_i - y_i|}{y_i}$, $\hat{y}_i$ is the predicted effort and y_i is the actual effort.

- **Percentage of estimations within 25% of the actual values:**

$$PRED(25) = \frac{1}{T}\sum_{i=1}^{T} \begin{cases} 1, & \textbf{if } MRE_i \le \frac{25}{100} \\ 0, & \textbf{otherwise.} \end{cases}$$

- **Logarithmic Standard Deviation (LSD):**

$$LSD = \sqrt{\frac{\sum_{i=1}^{T}(e_i + \frac{s^2}{2})^2}{T-1}},$$

where s^2 is an estimator of the variance of the residual $e_i = \ln y_i - \ln \hat{y}_i$.

Current Situation

- **There is no universally agreed single performance measure.**
- **The relationship among different measures in SEE is not well understood.**
- **Existing SEE approaches use at most one measure during the learning procedure. It is unclear whether a model/learner trained using one measure would still perform well under a different measure.**
- **Many papers did not even report the measure they used in training!**

SEE by Multi-objective Learning

- **How about viewing SEE as a multi-objective learning problem?**
- **Each performance measure is considered explicitly as a separate objective in learning.**

But Why?

- **A multi-objective algorithm can be used to create SEE models that are generally good in terms of all objective measures, and present different trade-offs among these measures.**
- **These different trade-offs can help us to understand to what extent different measures behave differently and what the relationship among these measures is.**
- **They help to enhance the robustness of the models.**

Some Research Questions

- **What is the relationship among different performance measures for SEE?**
- **Can we use different performance measures as a source of diversity to create SEE ensembles? In particular, can that improve on the performance measures used as objectives with respect to a standard learning algorithm for the same type of base model?**
- **Is it possible to outperform the state-of-the-art?**

Multi-objective Ensemble Learning

- **Multi-objective ensemble learning does improve the performance of single objective learning.**
- **The use of different measures as separate objectives helped to increase the diversity in the ensemble and improve ensemble learning performance.**
- **The ensembles did well even on those performance measures that were not used in multi-objective learning, which provides an evidence demonstrating the robustness of the result.**

Concluding Remarks for Multi-Objective Learning

- **Multi-objective learning fits naturally with ensembles.**
- **There are different forms of multi-objective learning, e.g., different objectives.**

Outline of This Lecture

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] Arjun Chandra and Xin Yao. "Ensemble learning using multi-objective evolutionary algorithms". In: *Journal of Mathematical Modelling and Algorithms* 5.4 (2006), pp. 417–445.

[3] Huanhuan Chen and Xin Yao. "Multiobjective neural network ensembles based on regularized negative correlation learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22.12 (2010), pp. 1738–1751.

[4] 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.

[5] 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.

References for This Lecture II

[6] 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.

[7] Shuo Wang and Xin Yao. "Multiclass imbalance problems: Analysis and potential solutions". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42.4 (2012), pp. 1119–1130.

[8] 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. Bäck, 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 Cybernetics, Part C*, 35(2):233-243, May 2005.**