CONCLUSIONS AND SUGGESTIONS FOR FUTURE STUDIES

A. Conclusions

This thesis focuses on addressing the following two major issues. The first involves developing algorithms to improve in balancing convergence and diversity in multi-objective optimization problems, and the second is using those methods to resolve classification issues. Following is a summary of the major contributions made in this thesis.

(*) Balancing convergence and diversity in multi-objective optimization problems

- Proposing a DPP-based co-operative co-evolutionary approach for balancing the convergence and diversity.[C1].
- Proposing a DPP-based competitive co-evolutionary approach for balancing the convergence and diversity.[J1].

(*) Applying multi-objective co-evolutionary methods for classification with imbalanced data

- Proposing a multi-objective competitive co-evolutionary approach for imbalanced dataset classification (named IBDPPCP)[C7].
- Proposing a multi-objective co-operative co-evolutionary approach (named IBMCCA) for solving classification with imbalanced data [J2, C2, C3, C6].

B. Future Studies

Although the Co-evolution was studied widely in the literature, there are still several possible open problems which require further investigations in order to have a full understanding about their applicability as follows.

- Developing the DPP-based models using both of co-operative and competitive for multi-objective optimization problems.

- Developing a multi-objective multi-population for machine learning problems.

INTRODUCTION

In real life, there are many practical problems in which often-conflicted objectives need to be optimized simultaneously, especially in machine learning, where we are seeking a model with the best performance in both accuracy and generalization measures. These problems are called multi-objective optimization problems (MOPs). In multi-objective optimization (MOO), finding a set of solutions that satisfy both criteria: as close as possible to the Pareto-optimal front and as diverse as possible is a vital but time-consuming task. ***Maintaining a balance between diversity and convergence*** is a key concern in the field of multi-objective optimization.

In general, using only a single algorithm to solve the problem of balancing convergence and diversity in MOPs is not easy. Therefore, the current trend is to combine multiple algorithms and a ***co-evolutionary algorithm (CoEA)*** is one of them. The general idea of CoEA is to break down a problem into a set of sub-problems and use multiple populations to optimize different sub-problems.

The ***diversity and accuracy*** (i.e., convergence) are also keys to ***ensemble learning methods***. However, there is always a trade-off between classifier diversity and accuracy. From this point, it can be seen that multi-objective evolutionary algorithms in general and ***co-evolutionary algorithms in particular are ideal for ensemble learning*** because they can identify a collection of solutions that ensure both convergence and diversity. In this thesis, the author will concentrate on resolving two significant issues: first, proposing co-evolutionary algorithms for conventional multi-objective optimization issues (i.e., balancing diversity and convergence). Second, applying these co-evolutionary methods to machine learning issues (i.e., classification)

The dissertation is organised into three chapters except for introduction, conclusion, future work, bibliography and appendix. Chapter 1 gives the backgrounds related to this research. Chapter 2 presents the proposed methods of diversity and convergence, and Chapter 3 introduces the proposed methods of applying these co-evolutionary methods to imbalanced dataset classification problems.

Chapter 1

BACKGROUNDS

1.1 Multi-objective optimization

A multi-objective optimization problem (MOP) can be defined as follows:

Minimize:

$$F(x) = (f_1(x), ..., f_m(x))^T \tag{1.1}$$

Subject to: $g_i(x) \leq 0; \forall i = 1, ..., p$. $h_j(x) = 0; \forall j = 1, ..., q$.

Where, a solution $x = (x_1, ..., x_n) \in \Omega$ is a vector of decision variables; is the decision variable space or simply the decision space. $g_i(x)$ and $h_j(x)$ are called constraint functions. If any solution x satisfies all constraints and variable bounds, it is known as a feasible solution, otherwise, it is called an infeasible solution. There are m objective functions $F(x) = (f_1(x), ..., f_m(x))^T$; $F : \Omega \rightarrow \Re^m_+$.

Where $\Re^m_+$ is called the objective space. For each solution x in the decision variable space, there exists a point in the objective space.

1.2 Co-evolutionary Algorithms (CoEA)

Co-evolution is reciprocally generated evolutionary change between two or more species or populations. Traditional evolutionary algorithms (EAs) evaluate an individual's fitness objectively, separate from the population environment in which they are located. CoEAs operate similarly to standard EAs, with the exception that fitness evaluations are evaluated through its interactions with other individuals in the evolutionary system. The key benefit of CoEA over regular EA is its divide-and-conquer deconstruction approach. The CoEA primarily has four benefits. First, by breaking the problem down into smaller components, parallelism can accelerate the optimization process. Second, each subproblem is resolved by a different subpopulation, maintaining a wide variety of solutions. Third, breaking a system down into smaller components makes it more resilient to mistakes and failures in individual modules, which improves its capacity to be reused in dynamic contexts . Finally, if the issue is correctly decomposed, the rapid decrease in performance with a rise in the number of decision variables can be somewhat mitigated.

Scenario 3: Compare the proposed algorithm with ensemble learning algorithms

The detailed experimental results are shown in Figure 3.2. It is clear that the algorithms using the sampling solution give better results than the traditional algorithms. The difference is significant. Looking at this figure also shows the complete superiority of the proposed algorithms over the rest of the algorithms. Through this experimental scenario, the following conclusions can be drawn: Algorithms using sampling solutions give better and more stable results than conventional algorithms. The proposed algorithm still gives the best results when compared to these ensemble learning algorithms. This proves that ***selecting subsets from the original dataset by using the co-evolutionary methods is better than the sampling with replacement mechanism that is commonly used by ensemble machine learning algorithms.***

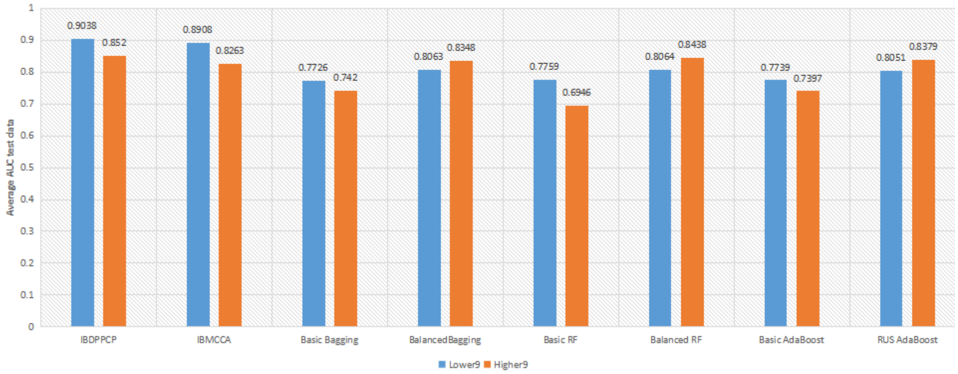

Figure 3.2: Experimental results of the proposed algorithm and ensemble learning algorithms (*Figures 3.17-3.18 in the thesis*)

3.4 Summary

In this chapter, the authors propose a competitive and a cooperative co-evolutionary algorithm for imbalanced data classification. The proposed algorithms take advantage of their strengths in creating sets of individuals that have both convergence and diversity factors to generate a collection of subsets of data that are used to generate classifiers in ensemble learning algorithms. Combined with hybrid data sampling solutions, IBDPPCP and IBMCCA have shown a good ability to handle problems related to imbalanced data.

Table 3.3: The Friedman test results for IBDPPCP and the state-of-the-art algorithms on two datasets , *Chi2* is the Chi-square value (*Table 3.4 in the thesis*)

Dataset	**Chi2**	**P-value**
Higher9	45.84983	3.17150e-08
Lower9	65.49744	3.41412e-12

rithms

Experimental results with each dataset are shown in Figure 3.1. It can be easily seen that the two proposed algorithms give better results in most of the test cases, except for the case of datasets with IR 9 > 9, where CNN gives a better result than IBMCCA. In the data set having $IR < 9$, *IBDPPCP* and *IBMCCA* give superior results compared to other algorithms. From the above experimental results, it can be concluded that the proposed algorithms and the CNN algorithm give more stable and better classification results than other algorithms on the datasets. However, in terms of the overlapping factor, the proposed algorithms give superior results to all. Thereby proving that ***the proposed algorithms are capable of handling the overlapping phenomenon of imbalanced data***.

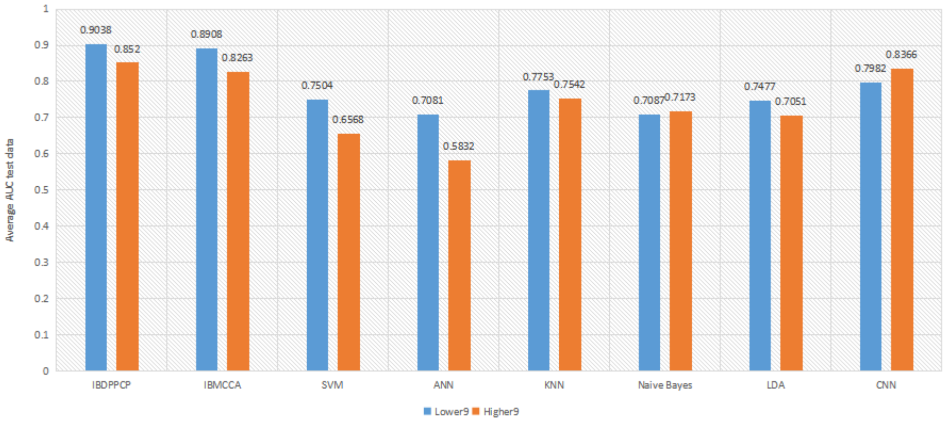

Figure 3.1: Experimental results of the proposed algorithm and machine learning algorithms (*Figures 3.15-3.16 in the thesis*)

CoEA can be divided into two main categories: competitive co-evolution and cooperative co-evolution.

1.2.1 *Cooperative co-evolutionary algorithms*

Cooperative co-evolutionary algorithms (CCEA - Algorithm 1) are frequently employed when an issue can be organically divided into smaller components (or sub-components). CCEA uses a different population (or species) for each of these sub-components. Since each individual in a given population only represents a portion of a possible solution to the issue. Therefore, to calculate fitness, a collaborator is chosen from the other populations to represent the other sub-components. The objective function is assessed once the individual is merged with this collaborator to form a complete solution. How successfully a subpopulation "cooperates" with other species to achieve beneficial outcomes is a measure of its fitness.

1.2.2 *Competetive co-evolutionary algorithms*

In competitive co-evolution (Algorithm 2), the first population makes an effort to fit into the second group's environment. The members of the second population will simultaneously make an effort to fit into the environment that the first population has built. The relative fitness evaluation function for each member of both populations will then be computed. The level of adaptation of a member of this population to the environment produced by one or a few members of the other population is represented by this relative evaluation function. Better fit individuals will be chosen for the following generation based on these relative fitness scores. Competitive co-evolution may result in an arms race when populations compete against one another to outperform one another and overcome more difficult issues.

1.3 The imbalanced data classification problem

A dataset is said to be imbalanced when a class or a set of classes is represented in a smaller number than the other classes. The majority class, also known as the negative class, is the set of data that contains the greatest number of instances, whereas the minority class, also known as the positive class, contains the fewest examples.

To address this issue, numerous methods are now being considered. Three

Algorithm 1: Cooperative co-evolutionary algorithms (CCEA)

```
Data: P← {$P_1, P_2, ..., P_N$}
Result: P
for population $p_s \in P$, all population do
    Initialize population $p_s$
for population $p_s \in P$, all population do
    Evaluate $p_s$
t:= 0
do
    for population $p_s \in P$, all population do
        Select parents from population $p_s$
        Generate offspring from parents
        Select collaborators from P
        Make a complete solution via combining offspring with
          collaborators
        Evaluate offspring via the fitness of complete solution
        Select survivors for new population $p_s$
    t:= t+1
until Terminating criteria is met
```

major groupings of these solutions can be identified: data-level algorithms, algorithm-level algorithms and algorithms based on cost-sensitive learning. Algorithm-level algorithms consist of developing brand-new algorithms or improving already-existing ones to cope with uneven datasets. Cost-sensitive learning is a strategy combining data- and algorithmic-level approaches while taking into account larger costs for misclassifying samples from the positive class in comparison to the negative ones. Currently, a group of data-level algorithms is most commonly applied. Resampling can be done in three different ways: (a) undersampling the majority class; (b) oversampling the minority class. (c) a hybrid strategy that incorporates (a) and (b). Synthetic minority over-sampling technique (SMOTE), Edited Nearest Neighbor (ENN), Tomek link, and Random Undersampling (RU) are typical resampling algorithms.

Table 3.1: Experimental results with IR less than 9 (*Table 3.7 in the thesis*).

Data	IBDPPCP	IBDPPCP2	IBDPP2	IBMCCA	EFIS_MOEA	DEMOA	SMEN_C45
yeast3	0.9548 ± 0.0022 (1)	0.9515 ± 0.0011 (2)	0.8827 ± 0.0317 (7)	0.9452 ± 0.0049 (3)	0.9395 ± 0.0055 (5)	0.9446 ± 0.0048 (4)	0.9326 ± 0.0000 (6)
yeast1	0.7543 ± 0.0039 (2)	0.7564 ± 0.0036 (1)	0.7096 ± 0.0126 (7)	0.7397 ± 0.0068 (5)	0.7439 ± 0.0055 (4)	0.7457 ± 0.0051 (3)	0.7138 ± 0.0000 (6)
wisconsin	0.9823 ± 0.0013 (1)	0.9815 ± 0.0017 (2)	0.9649 ± 0.0040 (6)	0.9709 ± 0.0032 (3)	0.9700 ± 0.0035 (4)	0.9699 ± 0.0022 (5)	0.9596 ± 0.0000 (7)
vehicle3	0.8205 ± 0.0049 (1)	0.8041 ± 0.0054 (4)	0.7580 ± 0.0124 (6)	0.7984 ± 0.0089 (5)	0.8095 ± 0.0079 (3)	0.8110 ± 0.0070 (2)	0.7452 ± 0.0000 (7)
vehicle2	0.9815 ± 0.0024 (1)	0.9801 ± 0.0012 (2)	0.9335 ± 0.0162 (7)	0.9689 ± 0.0075 (5)	0.9719 ± 0.0051 (3)	0.9712 ± 0.0052 (4)	0.9434 ± 0.0000 (6)
vehicle1	0.8054 ± 0.0036 (1)	0.7863 ± 0.0055 (4)	0.7581 ± 0.0147 (7)	0.7842 ± 0.0080 (5)	0.7969 ± 0.0056 (3)	0.7978 ± 0.0056 (2)	0.7591 ± 0.0000 (6)
vehicle0	0.9678 ± 0.0025 (2)	0.9687 ± 0.0022 (1)	0.9328 ± 0.0129 (6)	0.9597 ± 0.9597 (5)	0.9607 ± 0.0048 (4)	0.9619 ± 0.0037 (3)	0.9117 ± 0.0000 (7)
segment0	0.9958 ± 0.0007 (1)	0.9918 ± 0.0007 (5)	0.9893 ± 0.0020 (7)	0.9934 ± 0.0013 (2)	0.9906 ± 0.0017 (6)	0.9925 ± 0.0014 (3)	0.9920 ± 0.0000 (4)
pima	0.7889 ± 0.0037 (2)	0.7892 ± 0.0042 (1)	0.7562 ± 0.0126 (6)	0.7753 ± 0.0064 (4)	0.7746 ± 0.0070 (5)	0.7773 ± 0.0074 (3)	0.7376 ± 0.0000 (7)
newthyroid2	0.9885 ± 0.0031 (1)	0.9679 ± 0.0071 (4)	0.9511 ± 0.0144 (7)	0.9727 ± 0.0063 (3)	0.9564 ± 0.0107 (6)	0.9635 ± 0.0094 (5)	0.9794 ± 0.0000 (2)
newthyroid1	0.9953 ± 0.0034 (1)	0.9834 ± 0.0057 (4)	0.9613 ± 0.0094 (7)	0.9869 ± 0.0049 (3)	0.9741 ± 0.0128 (6)	0.9771 ± 0.0109 (5)	0.9887 ± 0.0000 (2)
iris0	0.9976 ± 0.0025 (1)	0.9965 ± 0.0024 (2)	0.9795 ± 0.0123 (7)	0.9908 ± 0.0066 (3)	0.9806 ± 0.0089 (6)	0.9842 ± 0.0101 (5)	0.9900 ± 0.0000 (4)
haberman	0.6394 ± 0.0112 (2)	0.6709 ± 0.0083 (1)	0.6324 ± 0.0237 (3)	0.6142 ± 0.0156 (5)	0.6131 ± 0.011 (6)	0.6219 ± 0.0131 (4)	0.5872 ± 0.0000 (7)
glass6	0.9500 ± 0.0048 (1)	0.9498 ± 0.0030 (2)	0.9148 ± 0.0163 (5)	0.9191 ± 0.0128 (3)	0.9085 ± 0.0213 (6)	0.8973 ± 0.0246 (7)	0.916 ± 0.0000 (4)
glass1	0.7966 ± 0.0105 (1)	0.7839 ± 0.0131 (3)	0.7183 ± 0.0265 (6)	0.7817 ± 0.0177 (4)	0.7810 ± 0.0112 (5)	0.7919 ± 0.0124 (2)	0.6987 ± 0.0000 (7)
glass0	0.8622 ± 0.0092 (2)	0.8644 ± 0.0067 (1)	0.8067 ± 0.0218 (7)	0.8391 ± 0.014 (5)	0.8471 ± 0.0159 (3)	0.8471 ± 0.0147 (3)	0.8140 ± 0.0000 (6)
glass0123vs4565	0.9551 ± 0.0063 (1)	0.9468 ± 0.0059 (5)	0.9098 ± 0.0202 (7)	0.9522 ± 0.0097 (4)	0.9529 ± 0.0071 (2)	0.9527 ± 0.0083 (3)	0.9184 ± 0.0000 (6)
ecoli3	0.9102 ± 0.0061 (1)	0.9013 ± 0.0047 (2)	0.8856 ± 0.0186 (7)	0.8996 ± 0.0107 (3)	0.8968 ± 0.0106 (4)	0.8905 ± 0.0144 (5)	0.8861 ± 0.0000 (6)
ecoli2	0.9249 ± 0.0052 (1)	0.9181 ± 0.0046 (2)	0.8932 ± 0.0117 (7)	0.9103 ± 0.0072 (5)	0.9104 ± 0.0067 (4)	0.9116 ± 0.0071 (3)	0.9000 ± 0.0000 (6)
ecoli1	0.9246 ± 0.0024 (1)	0.9216 ± 0.004 (3)	0.9016 ± 0.0224 (7)	0.9219 ± 0.0046 (2)	0.9071 ± 0.0096 (5)	0.9090 ± 0.0079 (4)	0.9034 ± 0.0000 (6)
ecoli0vs15	0.9841 ± 0.0012 (2)	0.9844 ± 0.0010 (1)	0.9720 ± 0.0090 (7)	0.9832 ± 0.001 (3)	0.9832 ± 0.0000 (3)	0.983 ± 0.0005 (6)	0.9832 ± 0.0000 (3)
AVERAGE	0.9038 (1.29)	0.8999 (2.48)	0.8672 (6.48)	0.8908 (3.81)	0.889 (4.43)	0.8906 (3.86)	0.8695 (5.48)

Table 3.2: Experimental results with IR higher than 9 (*Table 3.7 in the thesis*).

Dataset	IBDPPCP	IBDPPCP2	IBDPP2	IBMCCA	EFIS_MOEA	DEMOA	SMEN_C45
vowel0	**0.981 ± 0.0023 (1)**	**0.9688 ± 0.0023 (3)**	**0.9468 ± 0.0123 (6)**	**0.97 ± 0.0057 (2)**	**0.9645 ± 0.0083 (4)**	**0.9622 ± 0.0072 (5)**	**0.9461 ± 0.0000 (7)**
shuttlec2vsc4	**1.0000 ± 0.0000 (1)**	**1.0000 ± 0.0000 (1)**	0.995 ± 0.0045 (7)	**1.0000 ± 0.0000 (1)**	**1.0000 ± 0.0000 (1)**	0.9996 ± 0.0008 (6)	**1.0000 ± 0.0000 (1)**
shuttlec0vsc4	0.9999 ± 0.0001 (2)	**1.0000 ± 0.0000 (1)**	0.9989 ± 0.0019 (6)	0.9989 ± 0.0001 (6)	0.9998 ± 0.0003 (3)	0.9998 ± 0.0002 (3)	0.9997 ± 0.0000 (5)
glass5	0.911 ± 0.0115 (2)	**0.9431 ± 0.0087 (1)**	0.8591 ± 0.0374 (6)	0.8907 ± 0.0248 (3)	0.8807 ± 0.0312 (5)	0.8893 ± 0.0288 (4)	0.8195 ± 0.0000 (7)
glass4	**0.9483 ± 0.0099 (1)**	**0.8818 ± 0.0214 (4)**	0.8917 ± 0.0317 (2)	0.8857 ± 0.0181 (3)	0.8794 ± 0.0188 (5)	0.866 ± 0.0283 (6)	0.8646 ± 0.0000 (7)
glass2	**0.7921 ± 0.0207 (1)**	**0.6348 ± 0.0255 (7)**	0.7382 ± 0.0542 (6)	0.7646 ± 0.0361 (3)	0.7754 ± 0.0384 (2)	0.7595 ± 0.0366 (4)	0.7452 ± 0.0000 (5)
glass016vs5	**0.938 ± 0.028 (1)**	**0.934 ± 0.0145 (2)**	0.8905 ± 0.0433 (4)	0.8978 ± 0.0426 (3)	0.8817 ± 0.0313 (6)	0.8895 ± 0.0399 (5)	0.8657 ± 0.0000 (7)
glass016vs2	**0.7515 ± 0.029 (1)**	**0.597 ± 0.034 (7)**	0.6897 ± 0.0421 (5)	0.7068 ± 0.0363 (4)	0.738 ± 0.0306 (3)	0.7423 ± 0.0312 (2)	0.6575 ± 0.0000 (6)
ecoli4	**0.9534 ± 0.0056 (1)**	**0.927 ± 0.0028 (2)**	0.8845 ± 0.0335 (5)	0.9015 ± 0.0209 (4)	0.8727 ± 0.019 (7)	0.8779 ± 0.0214 (6)	0.9084 ± 0.0000 (3)
ecoli0137vs26	0.857 ± 0.0106 (2)	0.8886 ± 0.0067 (1)	0.8118 ± 0.0416 (7)	0.8211 ± 0.0162 (6)	0.8335 ± 0.0148 (4)	0.8312 ± 0.007 (5)	0.8345 ± 0.0000 (3)
abalone918	0.7488 ± 0.0107 (2)	**0.7752 ± 0.0073 (1)**	0.7105 ± 0.0374 (7)	0.7321 ± 0.0206 (3)	0.7186 ± 0.0143 (6)	0.7214 ± 0.015 (5)	0.7295 ± 0.0000 (4)
abalone19	0.6247 ± 0.0213 (2)	**0.7005 ± 0.009 (1)**	0.6183 ± 0.0054 (3)	0.6134 ± 0.0231 (5)	0.6147 ± 0.0166 (4)	0.6127 ± 0.0245 (6)	0.5159 ± 0.0000 (7)
yeast6	0.8894 ± 0.0045 (2)	**0.9013 ± 0.0036 (1)**	0.8623 ± 0.0209 (3)	0.8623 ± 0.0128 (3)	0.8091 ± 0.0187 (7)	0.8385 ± 0.0177 (5)	0.8148 ± 0.0000 (6)
yeast5	0.9745 ± 0.0029 (1)	0.9728 ± 0.0009 (2)	0.9547 ± 0.0178 (7)	0.9613 ± 0.0045 (5)	0.9627 ± 0.0074 (4)	0.9644 ± 0.006 (3)	0.956 ± 0.0000 (6)
yeast4	0.8414 ± 0.0075 (2)	**0.87 ± 0.0028 (1)**	0.8181 ± 0.0226 (3)	0.8064 ± 0.0138 (5)	0.7988 ± 0.0119 (6)	0.8176 ± 0.014 (4)	0.7636 ± 0.0000 (7)
yeast2vs8	0.8412 ± 0.0125 (1)	0.8066 ± 0.0153 (5)	0.7488 ± 0.0604 (7)	0.8334 ± 0.0174 (4)	0.839 ± 0.0181 (2)	0.8365 ± 0.0145 (3)	0.804 ± 0.0000 (6)
yeast2vs4	0.9392 ± 0.007 (2)	**0.9561 ± 0.0017 (1)**	0.9046 ± 0.0207 (6)	0.9225 ± 0.0131 (4)	0.9178 ± 0.0126 (5)	0.9301 ± 0.0125 (3)	0.9012 ± 0.0000 (7)
yeast1vs7	0.7258 ± 0.0154 (2)	0.7246 ± 0.0147 (3)	0.6855 ± 0.0399 (7)	0.7 ± 0.0379 (4)	0.6959 ± 0.029 (6)	0.6994 ± 0.0282 (5)	0.7572 ± 0.0000 (1)
yeast1458vs7	0.6054 ± 0.0116 (1)	0.5994 ± 0.0184 (2)	0.5923 ± 0.041 (3)	0.5566 ± 0.0288 (7)	0.5638 ± 0.0205 (5)	0.5751 ± 0.0207 (4)	0.5624 ± 0.0000 (6)
yeast1289vs7	0.7200 ± 0.0161 (1)	0.6635 ± 0.0271 (5)	0.6635 ± 0.0466 (5)	0.6992 ± 0.0247 (2)	0.6986 ± 0.0175 (3)	0.6908 ± 0.0273 (4)	0.6215 ± 0.0000 (7)
yeast05679vs4	0.8488 ± 0.0054 (1)	0.8226 ± 0.0048 (5)	0.8203 ± 0.0191 (6)	0.8275 ± 0.0099 (2)	0.8242 ± 0.0124 (4)	0.8274 ± 0.0109 (3)	0.806 ± 0.0000 (7)
AVERAGE	**0.8520 (1.43)**	0.8365 (2.67)	0.8136 (5.29)	0.8263 (3.76)	0.8223 (4.38)	0.8253 (4.33)	0.8035 (5.48)

pling with replacement mechanisms. This comparison illustrates ***how useful it is to use co-evolutionary solutions to find subsets or decision trees***.

3.3.3 Results and analysis

Scenario 1: Compare the proposed algorithm with the state-of-the-art algorithms

+ *Question 1: Is IBDPPCP algorithm capable of solving the problem of imbalanced data if the sampling method is not used?*

Through the comparison of the two algorithms, IBDPPCP and IBDPPCP2, this answer will be clarified. Table 3.1 and 3.2 show the comparison results between the two algorithms. It is easy to see that these two algorithms give better results than the rest. However, looking at the average AUC results, it can be seen that the difference between these two algorithms is not significant. In the first dataset, IBDPPCP results in an average AUC of 0.9038, the value for IBDPPCP2 is 0.8999. In second dataset, the corresponding values for the two algorithms are 0.8520 and 0.8365. ***In summary, it can be said that the data sampling method enhances the outcomes. However, this difference is not significant. Even without data sampling, the IDPPCP algorithm is capable of handling imbalanced data.***

+**Statistical test for comparing performance**

Table 3.3 shows the results of the Friedman statistical test. For each dataset, the Friedman test returns a chi-square statistic and a p-value. The p-value is calculated based on the chi-square statistic and the degrees of freedom. The significance level is 0.05. The null hypothesis (i.e., Ho) assumes that there is no significant difference between the rankings of the algorithms being compared. As can be seen, the p-value is less than the significance level (3.17150e-08 and 3.41412e-12), so we can reject Ho and conclude that there is evidence of a significant difference among the algorithms.

Next, the author uses the Wilcoxon signed-rank test as a post-hoc test to determine which algorithms differ significantly from each other. The results of this statistical test show that most of the p-values are smaller than the significance level (i.e., 0.05). This indicates that there is strong evidence against the null hypothesis and that the two algorithms are likely to be different.

Scenario 2: Compare the proposed algorithm with machine learning algo-

Algorithm 2: Competitive co-evolutionary algorithms (CPEA)

```
Data: P_s : SolutionPopulation; P_t : TestPopulation; P
Result: P
P ← {P_s, P_t}
for population p ∈ P, all population do
    Initialize population p
for population p ∈ P, all population do
    Evaluate p
t:= 0
do
    for population p ∈ P, all population do
        Select parents from population p
        Generate offspring from parents
        if p is P_s then
            Select competitors from P_t
    else
        Select competitors from P_s
    Evaluate offspring via competing against collaborators
    Select survivors for new population p
    t:= t+1
until Terminating criteria is met
```

1.4 Summary

In this chapter, the author introduces fundamental knowledge related to the contents used in the thesis. Specifically, multi-objective optimization, co-evolution (specifically, co-operative and competitive co-evolution), classification with imbalanced data, approaches to solving it are presented.

Chapter 2

THE DUAL-POPULATION CO-EVOLUTIONARY METHODS FOR SOLVING MULTI-OBJECTIVE PROBLEMS

2.1 The dual-population paradigm (DPP)

In 2015, Ke Li et.al. dealt with the problem of balancing convergence and diversity in MOOPs by employing ***a dual-population cooperative co-evolution paradigm (named ED/DPP or abbreviated as DPP)***. DPP employed two co-evolving populations. The Pareto-based mechanism is used in the first population (named A_p) and the decomposition-based mechanism is used in the second population (named A_d). These populations engage in parallel evolution. At each generation, a restricted mating selection mechanism (RMS) allows them to interact with each other. In the RMS, the mating parent includes three solutions, of which two are selected from A_d and the remaining one is selected from A_p. Thanks to this way, the parents could pass on all the positive characteristics (i.e., convergence and diversity) to the offspring. To update both A_p and A_d, the offspring utilizes the corresponding archiving mechanism. In case there is no solution in the selected sub-region in A_p, an alternative solution is chosen by the RMS in the corresponding one in A_d. In case there is more than one solution found in the sub-region, only one solution is selected.

This algorithm gives some promising results. However, there are two areas for possible improvement, as discussed below:

1. Restricted mating selection method:

In DPP, a neighborhood of a sub-region is defined as a set of its several closest sub-regions. To take advantage of neighborhood information, the authors specify the neighborhood of each sub-region based on the Euclidean distance between unit vectors. The authors restrict the mating parents to neighboring sub-regions with a high probability (and there is only a low probability that these mating parents will be selected from the whole population). However, they only randomly select a neighboring sub-region from A_p regardless of whether this sub-region contains any solutions in the A_p or not. This leads to a high

algorithms

The two proposed algorithms are compared with five other algorithms.

+ **IBDPPCP2** is another version of IBDPPCP. The difference is that IBDPPCP2 does not use data sampling. The purpose of this comparison is to check the effectiveness of using the data sampling method as well as answer the question: ***Is the IBDPPCP capable of solving the problem of imbalanced data if the sampling method is not used?***.

+ **IBDPP2** is another version of the IBDPPCP algorithm. The difference is that IBDPP2 uses **DPP2 algorithm** instead of DPPCP in the co-evolutionary process. The purpose of this comparison is to check ***how efficient the two algorithms are then applied to the imbalanced data classification problem***.

+ **EFIS_MOEA** is the premise algorithm chosen by the author for improvement. Comparing the two proposed algorithms to determine ***whether the proposed algorithms are superior to the premise research***.

+ **DEMOA** [C6] is an algorithm that uses a decomposition mechanism for classification problems with imbalanced data. DEMOA uses only one population and utilizes MOEA/D as a decomposition algorithm. This comparison helps determine ***whether a dual population method is better than a single population method***.

+ **SMEN_C45** is an algorithm that uses only the data sampling method (i.e., SMOTE-ENN) to generate the balanced data without using the co-evolutionary process or ensemble learning. This comparison shows ***the effect of using co-evolution in combination with ensemble learning***.

+ Scenario 2: Compare the proposed algorithms with machine learning algorithms

This case study compares the performance of the proposed method against some of the most widely used machine learning algorithms, including conventional (i.e. SVM, ANN, KNN, Naive Bayes, and LDA) and deep learning methods(i.e. CNN).

+Scenario 3: Compare the proposed algorithm with the group of ensemble learning algorithms.

The main difference between the proposed methods and common ensemble learning algorithms is the way subsets are generated. In the proposed algorithms, subsets are created from individuals found through the co-evolutionary process; in the ensemble learning algorithm, the subsets are created using sam-

Algorithm 6: The IBMCCA algorithm

```
input  : DataSet
output: FinalResult
#Step 1: Data preprocessing
DataSet ← $PreProcessing(DataSet)$;
#Step 2: Population initiation
$P1 \leftarrow InitializePopulation1(N)$
$P2 \leftarrow InitializePopulation2(N)$
$Elite1 \leftarrow [11 \ldots 1]$
$ElitePool \leftarrow \Theta; BestArchive \leftarrow \Theta$
ElitePool.Add(Elite1);
CalculateFitness(P2, Elite1, DataSet);
Eltite2 ← $Sort(P2)$;
Pool.Add(Elite2);
#Step 3: The Co-evolutionary process
while Stop condition false do
    Reproduction(P1, Eltite2;
    Eltite1 ← $Sort(P1)$;
    Pool.Update(Elite1);
    Reproduction(P2, Eltite1);
    Eltite2 ← $Sort(P2)$;
    Pool.Update(Elite2);
#Step 4: Ensemble learning
BestArchive= Union ($\mathbf{Front0}_{P1}$, ($\mathbf{Front0}_{P2}$));
Classifiers = BuildTrees(BestArchive, Dataset)
FinalResult = EnsembleLearning(Classifiers, Dataset)
Return FinalResult;
```

3.3.2 Test scenarios

To evaluate the performance of the two proposed algorithms, in this study, the author conducts some experiments as follows:

+Scenario 1: Compare the proposed algorithm with the state-of-the-art

possibility that the selected sub-region does not contain any solutions (so an alternative solution has to be borrowed from the corresponding sub-region in A_d). ***This may lead to an imbalance between the two populations.***

2. The interaction between two co-evolving populations:

In DPP, the authors select x_{r3}^G and x_{r1}^G from A_d and x_{r2}^G from A_p to create a new offspring using the differential evolution (DE) operator: $x_i^{G+1} = x_{r3}^G + F * (x_{r1}^G - x_{r2}^G)$ with the hope that x_{r1}^G has good convergence properties and x_{r2}^G has promising diversity and offspring has a large chance of having both advantages. ***However, there still exist two major drawbacks:***

(+) Choosing two out of three solutions from the A_d and only one from the A_p may cause an imbalance in the co-evolutionary process.

(+) Since the direction vector is made up of two solutions in two different populations, it could lead to unpromising outcomes, especially when the two populations are imbalanced (i.e., the convergence of one population is much better than the other).

Inspired by the co-evolution paradigm in DPP, this study attempts to address the aforementioned drawbacks. The author proposes two new dual-population competitive co-evolutionary algorithm named DPP2 and DPPCP (*The dual-population competitive co-evolutionary algorithm*). These two proposed algorithms (i.e., DPP2 and DPPCP) are explained in more detail in the next sections.

2.2 A dual-population co-operative co-evolutionary method for solving multi-objective problems (DPP2)

The pseudo-code of this algorithm is shown in Algorithm.3. In the first step, A_p and A_d are randomly initialized. N solutions in A_d are evenly assigned to N sub-regions (according to N unit vectors). Later, in the process of evolution, each sub-region always has only one solution. This is to guarantee that A_d always has an even distribution (i.e., diversity) in objective space. Whereas, N solutions in A_p will be randomly assigned to N sub-regions. This means that more than one solution can be in the same sub-region, and there are also sub-regions that don't contain any solutions. Next, each solution specifies the T closest neighborhood sub-regions based on the Euclidean distance between unit vectors.

The author uses a new RMS mechanism (named RMS2) to select two mating parents. After the selection process, two mating parents (denoted to x_{r2} and

x_{r3}) are selected to generate new offspring by using DE operator. One thing to be underlined here is that the new offspring need to be assigned to a certain sub-region. In this research, this offspring belongs to the sub-region that has the minimum Euclidian distance between its unit vector and the offspring's objective vector. Finally, the new offspring is used to update each of A_p and A_d, respectively.

In general, the DPP2 has three main differences from the DPP:

First, when choosing one solution in A_p, instead of just selecting from a selected neighborhood sub-region, the author select from all T neighborhood sub-regions. By doing this, the probability of finding one solution in A_p will be much higher than in RMS.

Second, in case all T neighborhood sub-regions do not contain any solution. Instead of choosing an alternative solution in Ad, the author randomly selects a solution in A_p. In this way, the offspring are generated from parents in different populations, so they can take advantage of all the advantages of both parents (i.e., diversity and convergence).

Third, the update procedure for A_p is different from the original DPP. In particular, whenever a new offspring is generated, it will be stored in an offspring list (i.e., $offSpringAp$ in Algorithm.3) instead of being updated right away to A_p. After a generation finishes, $offSpringAp$ will be combined with A_p and the author uses the crowding distance sorting method (CDSM) in the combined population to select the N best solutions for the new population. The reason is that the CDSM is a really time-consuming method.

2.3 The dual-population competitive co-evolutionary method for solving multi-objective problems (DPPCP)

The pseudo-code of the proposed algorithm DPPCP is shown in Algorithm 4. Like DPP2, this algorithm employs two co-evolving populations:A_p and A_d. At each generation, the author uses a neighbor-based selection mechanism (NBSM) to select three candidate solutions from each of the populations. After that, the author uses DE operator to create two offspring named $Child_{Ap}$ (i.e., the offspring in population A_p) and $Child_{Ad}$ (the offspring in population A_d). Next, let $Child_{Ad}$ compete with $Child_{Ap}$ using Pareto dominance-based metrics and choose the winner to update A_p. Similarly, let $Child_{Ap}$ compete with $Child_{Ad}$ using decomposition-based metrics and use the winner to update A_d. At the end of the co-evolution process, the final population is a combination of

viduals representing different ways of selecting features. The second one (called the instance population or population 2) contains individuals that each represents a subset of the original dataset. In the process of co-evolution, in order to calculate the fitness value, each individual in the first population needs to be associated with the individual in the second population, and vice versa. Undergoing a co-evolutionary process, the output is a combination of the best individuals from two populations.

Objective functions: Because of the different purposes of each population, the objective functions are varied. The purpose of the feature population is to find individuals that have not only the least number of selected features but also the highest AUC value. Therefore, two chosen objectives for this population are the AUC and the number of selected features. Meanwhile, the second population (i.e., the instance population) tries to find individuals with the least number of selected instances as well as the highest AUC value. Therefore, AUC and the number of selected samples are chosen as two objectives for this population. IS stands for instance set, IS$_i$ is a binary value (0 or 1) converted from the probability of selection of an instance. FS stands for feature set, FS$_i$ is a binary value (0 or 1) converted from the probability of selection of a feature. Suppose D is the number of features and S is the number of samples. The formula for the objective functions of the two populations is as follows:

$$Population1: \begin{cases} OBJ_1 = & -AUC \\ OBJ_2 = & \sum_{i=0}^{D-1} FS_i \end{cases} \tag{3.2}$$

$$Population2: \begin{cases} OBJ_1 = & -AUC \\ OBJ_2 = & \sum_{i=0}^{S-1} IS_i \end{cases} \tag{3.3}$$

3.3 Experimental results

3.3.1 Experimental datasets

The experimental datasets consist of 42 standard imbalanced datasets. These data sets are divided into two groups: imbalance ratios lower than 9 and imbalance ratios higher than 9.

Algorithm 5: The IBDPPCP algorithm

```
input  : Dataset
output: FinalResult

$BestArchive \leftarrow \Theta$
#Step 1: Data preprocessing
Dataset ← $DataSampling(DataSet)$;
Dataset ← $DuplicateRemoving(DataSet)$;
#Step 2: The Co-evolutionary process
BestArchive= DPPCP (Dataset);
#Step 3: Ensemble learning
Classifiers = BuildTrees(BestArchive, Dataset)
FinalResult = EnsembleLearning(Classifiers, Dataset)
Return FinalResult;
```

$$\begin{cases} OBJ_1 = \quad AUC \\ OBJ_2 = \sum_{i=0}^{N-1} IS_i \end{cases} \tag{3.1}$$

where N is the number of samples of the training dataset; IS$_i$ is a binary value (0 or 1) converted from the probability of selection. The task now is to minimize these two functions. This implies that for any solution, the smaller the IS and the higher the AUC, the better it is.

3.2 A multi-objective cooperative co-evolutionary method for classification with imbalanced data (IBMCCA)

The pseudo-code for this algorithm is presented in Algorithm 6. There are two main differences between IBDPPCP and IBMCCA algorithms. The first one is individual encoding. In IBDPPCP, the individuals in the two populations use the same encoding (i.e., the two substrings FS and IS). In IBMCCA, each individual is a separate substring. Objective functions are the second difference. Because the role of each population is different, in IBMCCA, each of them uses different objective functions, whereas, in IBDPPCP, both populations use the same objective functions. There are two populations that evolve simultaneously to solve two tasks: feature selection and instance selection. The first population (called the feature population or population 1) includes indi-

Algorithm 3: DPP2 algorithm

```
input  : Maximum number of generations (M)
         Neighborhood Size (T)
         Population size (N)
output: Final Population P

$[A_p, A_d] = initializePopulation()$
$W = InitializeUniformWeight()$
$B = InitializeNeighborhood()$
$Z^* = InitializeIdealPoint()$
$Z^{nad} = InitializeNadirPoint()$
$m \leftarrow 0$
while $m < M$ do
    $offspringAp \leftarrow \varnothing$
    for $i \leftarrow 1$ to $N$ do
        Q= RMS2 ($A_p$, $A_d$, m, Bm) (Algorithm)
        $Child = CoOperativeMating(Q)$
        Mutate(Child);
        Update Sub-Region index for Child
        Update Idea point $Z^*$ and nadir point $Z^{nad}$
        Update $A^d$
        Add Child to offspringAp
        m++;
    $U$ = Union($offspringAp, A_p$)
    $Ap$ = crowdingDistanceSelection($U$)
```

both A_p and A_d populations. The reason for this decision is that each of them uses a different optimal mechanism. While A_p uses the true Pareto front, A_d utilizes the idea point (a solution with the best objective values known since running the algorithm) as the best goal to achieve. The roles of the two populations are the same. Therefore, in order to preserve the good properties of both populations (i.e., diversity and convergence), the author decided to keep both populations in the final selected population.

There are two differences between the DPPCP and other co-evolutionary methods (i.e. DPP and DPP2): *First*, in the DPPCP, the author does not use a cooperative co-evolutionary mechanism. In other words, this study has eliminated the mating parents' steps to generate offspring. Instead, this study uses a competitive mechanism to make two offspring interact with each other. *Second*, this study uses the NBSM mechanism to select three solutions in each population and use them to create two separate offspring. In general, the model is divided into four main steps: *Initialization, NBSM selection, Competitive process,*

Algorithm 4: DPPCP Algorithm

```
input  : M: The number of generations.
         T: The neighboring numbers
         N: The population size
output: Final Population $A_p$ and $A_d$
$[A_p, A_d]$ = initializePopulation()
$W$ = InitializeUniformWeight()
$B$ = InitializeNeighborhood()
$Z^*$ = InitializeIdealPoint()
$Z^{nad}$ = InitializeNadirPoint()
$m \leftarrow 0$
while $m < M$ do
    $offspringAp \leftarrow \varnothing$
    for $i \leftarrow 1$ to $N$ do
        $Child_{Ap}, Child_{Ad}$ = NBSMSelection($A_p, A_d, i, Bi$)
          ($Algorithm$)
        $Winner1$ == CompeteDominate($Child_{Ap}, Child_{Ad}$)
        $Winner2$ == CompeteDecompostion($Child_{Ap}, Child_{Ad}$)
        UpdateAp(Winner1, $A_p$); ($Algorithm$)
        UpdateAd(Winner2, $A_d$);
        Update $Z^*$ and $Z^{nad}$
        $m$++;
Return $P \leftarrow A_p \cup A_d$
```

Chapter 3

THE APPLICATION OF MULTI-OBJECTIVE CO-EVOLUTIONARY OPTIMIZATION METHODS FOR CLASSIFICATION PROBLEMS

3.1 A multi-objective competitive co-evolutionary method for classification with imbalanced data (IBDPPCP)

The proposed algorithm model is presented in Algorithm 5. There are three main phases: *Data pre-processing, the co-evolutionary process, and ensemble-based decision-making*. The general idea of the algorithm is as follows: Encoding each individual into a couple of feature sets (FS) and instance sets (IS) with the hope of finding the optimal ones that have both key features as well as important instances to help solve the imbalanced dataset problem. The multi-objective competitive co-evolutionary algorithm (i.e., DPPCP) is utilized to find the set of optimal individuals, then combine these individuals with an ensemble learning algorithm. It should be noted that to boost the performance of the ensemble learning algorithm, the weak learners should satisfy two criteria: *having diversity as well as good classification performance*. The multi-objective optimization algorithm helps generate weak learners that satisfy both of these two criteria. After the evolutionary process, all individuals in the final population are used as weak learners in the ensemble learning algorithm. A voting mechanism is used to determine the final result. This study uses a DPP-based algorithm (named IBDPPCP) as the multi-objective optimization algorithm and utilizes the C4.5 algorithm as the base learner to solve this problem.

Objective functions: There are two key objectives the author wishes to accomplish with this study. Increasing identification across all data classes (including minority and majority) is the first one. The second is to minimize the number of samples selected, or, in other words, to maximize the removal of bad samples. Two objective functions as follows:

2.5.3 *Statistical test for comparing performance*

In the previous comparisons, the proposed algorithms yield better average performance than the other algorithms. To further strengthen this claim, the authors conducted statistical evaluations to determine whether there is a significant difference between the algorithms. Specifically, the Friedman test, a non-parametric test, is used to check whether there are significant differences among the results. The null hypothesis (Ho) is that there is no difference between the algorithms. If the p-value is smaller than a significance level (i.e., 0.05), the null hypothesis is rejected (or there are significant differences between the algorithms), and vice versa. Table 2.3 shows the Friedman statistic of the IGD metric considering reduction performance (distributed according to chi-square with 4 degrees of freedom: 34.787), and the p-value is approximately 1.084e-06. From these p-values, it can be concluded that there is a significant difference between the compared algorithms and the DPPCP algorithm gives the best results.

Table 2.3: Average ranking of the algorithms using the IGD metric (*Table 2.7 in the thesis*).

Algorithm	Ranking
NSGAII	3.5806
MOEAD	3.1290
DPP	3.7742
DPP2	2.8710
DPPCP	1.6452

2.6 Summary

In this chapter, the author presents two DPP-based algorithms for balancing convergence and diversity in MOEAs. Specifically, a modified dual-population-based co-evolutionary algorithm (DPP2) and a dual-population competitive co-evolutionary (named DPPCP) algorithm are presented. The empirical results pointed out the efficacy of the co-evolutionary methods in balancing diversity and convergence for solving MOPs.

and Update population.

Another major difference between the two RMS and NBSM mechanisms is the solution selection procedure in A_p. For each small partition, this study conducts a search across the entire T neighborhood sub-regions instead of just choosing a random sub-region, as in the RMS mechanism. This way, the probability of finding three solutions is much higher. In the case that any solutions cannot be found in the neighborhood sub-regions, this study borrows from the A_d.

Table 2.1: Performance comparisons between the DPPCP and baseline algorithms using the IGD metric (*Table 2.4 in the thesis*).

	DPP	DPP2	DPPCP
ZDT1	$3.945162e-04_{6.4e-04}$	$5.554228e-05_{2.2e-07}$	$3.093510e-05_{0.0e+00}$
ZDT2	$4.602254e-05_{3.6e-05}$	$4.661050e-05_{7.3e-08}$	$3.241247e-05_{0.0e+00}$
ZDT3	$6.907624e-05_{8.2e-05}$	$8.860561e-05_{8.5e-07}$	$2.623904e-05_{0.0e+00}$
ZDT4	$3.273414e-01_{2.5e-01}$	$5.866687e-05_{4.8e-07}$	$3.104248e-05_{0.0e+00}$
ZDT6	$3.147239e-05_{8.0e-07}$	$4.628070e-05_{6.9e-09}$	$3.135962e-05_{0.0e+00}$
UF1	$1.219677e-03_{4.7e-04}$	$6.864305e-05_{3.0e-06}$	$5.686509e-05_{0.0e+00}$
UF2	$1.983361e-03_{6.9e-04}$	$4.221505e-04_{1.3e-04}$	$1.700001e-04_{0.0e+00}$
UF3	$6.117055e-03_{1.7e-03}$	$2.379576e-04_{5.5e-04}$	$1.820430e-03_{0.0e+00}$
UF4	$2.055704e-03_{3.4e-04}$	$1.955396e-03_{2.4e-04}$	$1.676999e-03_{0.0e+00}$
UF5	$1.319009e-01_{4.5e-02}$	$6.085614e-02_{2.8e-02}$	$1.547979e-01_{0.0e+00}$
UF6	$1.023728e-02_{2.6e-03}$	$6.511947e-03_{4.9e-03}$	$2.268674e-02_{0.0e+00}$
UF7	$7.577647e-04_{4.1e-04}$	$1.073177e-04_{3.7e-05}$	$1.185823e-04_{0.0e+00}$
UF8	$1.105829e-03_{3.2e-04}$	$1.033388e-03_{2.1e-04}$	$8.438084e-04_{0.0e+00}$
UF9	$2.296300e-03_{2.4e-04}$	$2.186628e-03_{1.5e-04}$	$2.269156e-03_{0.0e+00}$
UF10	$1.254666e-02_{4.0e-03}$	$4.860551e-03_{5.9e-04}$	$4.938679e-03_{0.0e+00}$
WFG1	$4.091125e-03_{2.0e-03}$	$2.672370e-04_{1.8e-05}$	$7.811839e-05_{0.0e+00}$
WFG2	$3.898765e-04_{1.9e-04}$	$6.061874e-04_{2.3e-05}$	$8.474260e-05_{0.0e+00}$
WFG3	$1.837856e-04_{8.8e-05}$	$5.480337e-05_{3.4e-08}$	$3.343249e-05_{0.0e+00}$
WFG4	$2.109915e-04_{2.9e-05}$	$6.344580e-05_{1.9e-06}$	$3.390037e-05_{0.0e+00}$
WFG5	$9.304804e-04_{2.1e-06}$	$9.328236e-04_{8.8e-07}$	$9.303355e-04_{0.0e+00}$
WFG6	$1.249467e-04_{9.1e-05}$	$9.143875e-05_{2.0e-07}$	$5.394634e-05_{0.0e+00}$
WFG7	$2.832908e-05_{3.6e-06}$	$4.054432e-05_{2.5e-08}$	$2.255773e-05_{0.0e+00}$
WFG8	$3.479606e-03_{9.6e-04}$	$3.172940e-03_{2.8e-03}$	$8.095268e-04_{0.0e+00}$
WFG9	$5.823859e-05_{3.0e-06}$	$4.076749e-05_{3.5e-07}$	$2.269694e-05_{0.0e+00}$
DTLZ1	$2.274694e-02_{1.4e-02}$	$3.471784e-04_{1.4e-06}$	$2.526425e-04_{0.0e+00}$
DTLZ2	$3.282544e-04_{9.9e-06}$	$4.301280e-04_{1.9e-06}$	$3.343429e-04_{0.0e+00}$
DTLZ3	$1.647616e-01_{3.3e-01}$	$7.229934e-04_{7.5e-06}$	$5.305681e-04_{0.0e+00}$
DTLZ4	$4.906612e-04_{2.2e-05}$	$8.454214e-04_{2.2e-04}$	$5.417136e-04_{0.0e+00}$
DTLZ5	$2.934037e-05_{6.4e-06}$	$1.518117e-05_{1.8e-07}$	$3.674228e-06_{0.0e+00}$
DTLZ6	$1.138830e-05_{1.1e-06}$	$3.454072e-05_{2.2e-08}$	$8.785615e-06_{0.0e+00}$
DTLZ7	$1.091168e-03_{5.5e-05}$	$2.612434e-03_{2.2e-04}$	$1.181486e-03_{0.0e+00}$

2.4 Test scenarios

In this thesis, the author presents a comparison between the proposed algorithms and other algorithms such as with some baseline algorithms (i.e., NSGA-II and MOEA/D-DE) and the state-of-the-art algorithms (i.e. DPP). Via the comparison results, it can be seen how good the performance of the proposed methods are when compared to the others. Additionally, the thesis also conducts some other experiments that are not presented in this document such as comparing with a variant named *DPPCP-Variant1* to know the effects of competitiveness, comparing with two other variants named *DPPCP-Variant2* and *DPPCP-Variant3* to know the effects of the NBSM mechanism and comparing with two other variants named *DPPCP-Ap* and *DPPCP-Ad* to know the interaction between two co-evolving populations.

2.5 Results and discussions

2.5.1 Comparing with the baseline algorithms

The results in Table 2.1 show that the DPPCP is clearly better than DPP and DPP2 (it gives a better metric value in 24 out of 31 comparisons). In ZDT instances, DPPCP gives better results than DPP in all instances, especially in ZDT4, where DPPCP outperforms DPP about 10,000 times. In UF instances, DPP achieves better performance on UF5 and UF6 instances. However, DPPCP obtains better IGD metric values in other UF instances; even with UF1, it is better about 100 times. Similar to WFG instances, DPPCP achieves better metric values in all of the comparisons (except WFG5). These findings demonstrate that the competitive co-evolution model proposed in this study outperforms other cooperative co-evolution methods.

2.5.2 DPPCP with baseline algorithms

Tables 2.2 provide the performance comparisons of DPPCP, MOEA/D-DE, and NSGA-II on 31 test instances using the IGD metric. Based on experimental results, it can be seen that DPPCP achieves a better outcome than both NSGA-II and MOEA-D/DE. It wins 26 out of 31 comparisons. It is worth noting that although NSGA-II is the worst among the three candidates, it achieves the best IGD metric values on the UF4 and the UF5. Meanwhile, MOEA/D-DE obtains the best IGD metric values on the UF3, UF6, UF9, and WFG5. By contrast, DPPCP shows a poor result on the UF5 test instance. However, DPPCP shows better performance than the baseline algorithm on all the ZDT and DTLZ instances. These results indicate the effectiveness of DPPCP in achieving both convergence and diversity criteria.

Table 2.2: Performance comparisons between the DPPCP and baseline algorithms using the IGD metric. The metric value with the highest mean is emphasized by being displayed in bold font with a gray background (*Table 2.6 in the thesis*).

	NSGAII	MOEAD	DPPCP
ZDT1	$5.788071e-05_{4.8e-06}$	$5.556843e-05_{3.3e-07}$	$3.093510e-05_{0.0e+00}$
ZDT2	$5.968199e-05_{2.9e-06}$	$4.660528e-05_{4.7e-08}$	$3.241247e-05_{0.0e+00}$
ZDT3	$4.146423e-05_{1.9e-06}$	$8.838159e-05_{7.9e-07}$	$2.623904e-05_{0.0e+00}$
ZDT4	$5.699439e-05_{2.2e-06}$	$5.906846e-05_{5.0e-07}$	$3.104248e-05_{0.0e+00}$
ZDT6	$7.378607e-05_{4.2e-06}$	$4.628025e-05_{6.2e-09}$	$3.135962e-05_{0.0e+00}$
UF1	$3.546947e-03_{6.0e-04}$	$6.903693e-05_{5.0e-06}$	$5.686509e-05_{0.0e+00}$
UF2	$1.066904e-03_{2.9e-04}$	$3.608501e-04_{2.4e-04}$	$1.700001e-04_{0.0e+00}$
UF3	$7.154636e-03_{1.8e-03}$	$1.702418e-04_{1.5e-04}$	$1.820430e-03_{0.0e+00}$
UF4	$1.358224e-03_{2.4e-05}$	$1.940253e-03_{2.3e-04}$	$1.676999e-03_{0.0e+00}$
UF5	$4.394656e-02_{8.3e-03}$	$6.607502e-02_{1.7e-02}$	$1.547979e-01_{0.0e+00}$
UF6	$8.797878e-03_{3.8e-03}$	$3.479782e-03_{8.8e-03}$	$2.268674e-02_{0.0e+00}$
UF7	$1.859278e-03_{1.7e-03}$	$1.085036e-04_{2.4e-05}$	$1.185823e-04_{0.0e+00}$
UF8	$2.981191e-03_{1.7e-04}$	$9.841127e-04_{4.3e-04}$	$8.438084e-04_{0.0e+00}$
UF9	$2.732983e-03_{2.0e-03}$	$2.165702e-03_{1.5e-03}$	$2.269156e-03_{0.0e+00}$
UF10	$5.161469e-03_{3.7e-03}$	$4.986122e-03_{6.2e-04}$	$4.938679e-03_{0.0e+00}$
WFG1	$3.200666e-04_{2.3e-05}$	$2.702204e-04_{2.7e-05}$	$7.811839e-05_{0.0e+00}$
WFG2	$1.174109e-04_{1.1e-05}$	$6.057198e-04_{6.6e-06}$	$8.474260e-05_{0.0e+00}$
WFG3	$6.512669e-05_{3.3e-06}$	$5.476262e-05_{9.3e-08}$	$3.343249e-05_{0.0e+00}$
WFG4	$5.717426e-05_{2.8e-06}$	$6.238275e-05_{1.3e-06}$	$3.390037e-05_{0.0e+00}$
WFG5	$9.330129e-04_{5.7e-07}$	$9.337936e-04_{3.8e-07}$	$9.303355e-04_{0.0e+00}$
WFG6	$1.122812e-04_{6.5e-05}$	$9.139109e-05_{1.4e-07}$	$5.394634e-05_{0.0e+00}$
WFG7	$3.877603e-05_{1.7e-06}$	$4.054086e-05_{1.9e-08}$	$2.255773e-05_{0.0e+00}$
WFG8	$2.747423e-03_{2.2e-03}$	$3.174261e-03_{2.4e-03}$	$8.095268e-04_{0.0e+00}$
WFG9	$4.337451e-05_{5.0e-06}$	$4.071753e-05_{1.1e-07}$	$2.269694e-05_{0.0e+00}$
DTLZ1	$3.201110e-04_{1.9e-05}$	$3.474762e-04_{1.3e-06}$	$2.526425e-04_{0.0e+00}$
DTLZ2	$4.355620e-04_{2.4e-05}$	$4.306742e-04_{3.0e-06}$	$3.343429e-04_{0.0e+00}$
DTLZ3	$6.953162e-04_{2.1e-05}$	$7.211632e-04_{6.2e-06}$	$5.305681e-04_{0.0e+00}$
DTLZ4	$7.724318e-04_{1.2e-04}$	$7.911537e-04_{1.3e-04}$	$5.417136e-04_{0.0e+00}$
DTLZ5	$6.185971e-06_{4.8e-07}$	$1.516055e-05_{1.1e-07}$	$3.674228e-06_{0.0e+00}$
DTLZ6	$3.789467e-04_{2.9e-04}$	$3.453845e-05_{3.1e-08}$	$8.785615e-06_{0.0e+00}$
DTLZ7	$1.204211e-03_{6.4e-05}$	$2.612999e-03_{1.7e-04}$	$1.181486e-03_{0.0e+00}$

- Developing Spatial-based co-evolutionary algorithms for solving spatial challenges such as spatial forest planning, groundwater management, etc.

PUBLICATIONS

J1. **Van Truong VU**, Lam Thu BUI, and Trung Thanh NGUYEN. "*A competitive co-evolutionary approach for the multi-objective evolutionary algorithms.*" IEEE Access 8 (2020): 56927-56947, **SCIE, Q1, IF: 4.64**.

J2. **Van Truong VU**, Lam Thu BUI, and Trung Thanh NGUYEN. "*An Ensemble Co-Evolutionary based Algorithm for Classification Problems.*" Journal of Research and Development on Information and Communication Technology 2019.1, 10/2019.

C1. **Van Truong VU**, Lam Thu Bui, Trung Thanh Nguyen, "*A modified dual-population approach for solving multi-objective problems.*" 2017 21st Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES). IEEE, 11/2017 [**SCOPUS Conference**].

C2. **Van Truong VU**, Lam Thu Bui, Trung Thanh Nguyen, *A Coevolutionary approach for classification problems: Preliminary results.* In: 5th NAFOSTED Conference on Information and Computer Science, NICS 2018, 23 November 2018 through 24 November 2018 [**SCOPUS Conference**].

C3. **Van Truong VU**, Lam Thu Bui, Trung Thanh Nguyen, *A multi-objective cooperative coevolutionary approach for remote sensing image classification*, 2019 11th International Conference on Knowledge and Systems Engineering (KSE), 10/2019 [**SCOPUS Conference**].

C4. **Van Truong VU**; Lam Thu BUI; Trung Thanh NGUYEN, *A multi-objective competitive co-evolutionary approach for classification problems*, The NAFOSTED Conference on Information and Computer Science (NICS), 08/2019 [**SCOPUS Conference**].

C5. Lam Thu Bui, **Van Truong VU**, Bich Van Pham, Viet Anh Phan, *A multi-population coevolutionary approach for Software defect prediction with imbalanced data*, 2022 14th International Conference on Knowledge and Systems Engineering (KSE), 10/2022.

C6 Lam Thu Bui, **Van Truong VU**, Huong Dinh Thi Thu, Hang Le Minh, *A multi-objective co-operative co-evolutionary method for classification with imbalanced data*, 2023 15th International Conference on Knowledge and Systems Engineering (KSE), 10/2023 [**Accepted Paper**].

C7. ***Van Truong VU****, Lam Thu Bui, A decomposition-based ensemble multi-objective optimization algorithm for imbalanced dataset classification problems*, Hội thảo quốc gia lần thứ XXIV "Một số vấn đề chọn lọc của công nghệ thông tin và truyền thông, 12/2021.

C8. . Thi Thu Huong Dinh **, Van Truong VU**, Lam Thu Bui, *An ensemble multi-objective particle swarm optimization approach for exchange rates forecasting problem*, The 4th International Conference on Machine Learning and Soft Computing (ICMLSC 2020), 1/2020 [**SCOPUS Conference**].