EE682
HW5 - Performance Comparison between MOPBIL and MQEA

Herman Kolstad Jakobsen
20196493

December 6, 2019

Contents

1 Introduction **1**

2 MOPBIL **1**

3 MQEA **1**

4 Comparison of performance **2**

5 Conclusion **3**

1 Introduction

The task is to compare the performance of *Multi-Objective Population-based Incremental Learning (MOPBIL)* and *Multiobjective Quantum-inspired Evolutionary Algorithm (MQEA)* for the DTLZ2 test function. The performance of the algorithms will be compared using diversity and hypervolume as metrics. First, an overview of MOPBIL will be given. Then, MQEA will be presented. Finally, the performance of MOPBIL will be compared to the performance of MQEA.

2 MOPBIL

Multi-Objective Population-based Incremental Learning can be regarded as a method of combining genetic algorithms and competitive learning. Competitive learning is mostly used to cluster a number of unlabeled points into distinct groups. This is done by determining some desired features and then cluster the points into groups based on these features. During this process, the system learns and gets better by adjusting different weights.

The main difference between genetic algorithms and Multi-Objective Population-based Incremental Learning is that the latter evolves the population as a whole from one generation to the next, instead of evolving the solutions individually. Due to this, the whole probability distribution for the population is represented in Multi-Objective Population-based Incremental Learning algorithms. This probability distribution is represented by a probability vector, where its elements represents the probability of setting a bit in a chromosome equal to "1". In the case of real number application, generated solutions are decoded to real numbers and then evaluated by a fitness measure. An archive containing the best solutions is then created by taking the union of the previous archive and the nondominated solutions from the current generation. If the archive exceeds its maximum size, some solutions having smaller "nearest neighbour distance" are truncated. A smaller value of distance to its nearest neighbour represents that the solution is clustered together with other solutions. Due to the desire of having a largest possible diversity among solutions, the truncation is necessary. Finally, the probability vector is updated using the current archive as a reference. The algorithm is illustrated in fig. 1.

3 MQEA

In order to encompass multiple objectives, the Quantum-inspired Evolutionary Algorithm is extended with multiple subpopulations which evolve independently. An illustration of the new structure is shown in fig. 2. In order to decide which generated solution is better, MQEA use the following two conditions:

- if one solution is better than the other based on Pareto dominance or;

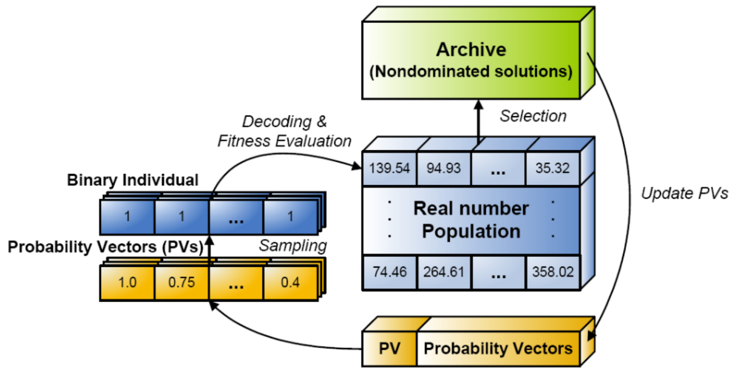

Figure 1: Multi-Objective Population-based Incremental Learning

- if one solution is closer to the Pareto-optimal front than the other when they are indifferent.

The global population is then filled with the best solutions from the subpopulations, where the archive will consist of the nondominated solutions from the global population. Next, the reference binary solutions are updated with randomly selected solutions from the archive. Finally, the Q-bits are updated using the reference solutions and quantum gates.

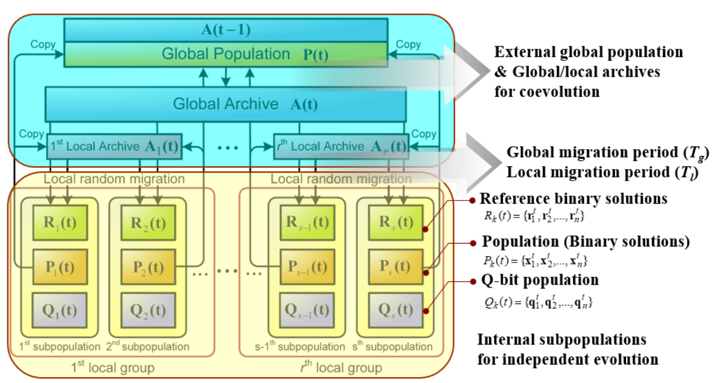

Figure 2: Multiobjective Quantum-inspired Evolutionary Algorithm

4 Comparison of performance

Before comparing the algorithms, it should be stated that there are several ways to modify their performance. For MOPBIL, several parameters can be tuned. This includes:

- Learning rate, LR - decide to what extent the probability vector should be modified based on the best solution.
- Mutation, ms - Decide to what extent the probability vector should be randomly modified.
- Probability of mutation, pm - Set the probability of a mutation happening.

Due to little experience and knowledge of MOPBIL, the parameters were kept at their default values.

For the MQEA, however, the performance is depending mainly on which quantum gates the algorithm uses and how the reference binary solutions are chosen. In the last homework, different tunings of the rotation gate and the square-root-of-NOT (SRN) gate were tested out. The different gates did not give a better performance. Hence, the default quantum gate was used for this assignment. Instead of selecting reference solutions randomly, the algorithm might benefit from implementing a more preference-based selection method. This could aid the algorithm to converge towards preferred solutions faster.

That being said, the raw output from MOPBIL and MQEA can be found in their respective *Results/* folders. The performance metrics for the two algorithms are shown in table 1. While MOPBIL has a slightly larger hypervolume, MQEA has a much larger diversity. This means that the algorithms covers almost the same size of dominated space. However, the nondominated solutions from MQEA are more spread out. It can therefore be concluded that MQEA performs better than MOPBIL.

Table 1: Overview of performance for MOPBIL and MQEA

Performance metrics		
	MOPBIL	MQEA
Diversity	89.2097	150.6162
Hypervolume	977.59	947.48

The difference in diversity between the two algorithms can be a direct effect of the probabilistic models imposed by Q-bits. Rather than representing the population as points, as MOPBIL does, a quantum population consists of probability distributions of sampling the search space. In that way, a Q-bit individual can represent multiple solutions and therefore cover a wider search area, which will result in more diverse solutions.

5 Conclusion

Multi-Objective Population-based Incremental Learning can be regarded as a method of combining genetic algorithms and competitive learning. Instead of evolving the solutions individually like genetic algorithms, Multi-Objective

Population-based Incremental Learning evolves the population as a whole from one generation to the next. In order to encompass multiple objectives, the Quantum-inspired Evolutionary Algorithm is extended with multiple subpopulations which evolve independently. MQEA then use two conditions to distinguish between superior and inferior solutions. There are several ways to modify the performance of the algorithms. For MOPBIL different parameters can be tuned, while the performance of MQEA is heavily dependent on how the quantum gates are chosen. The two algorithms gave solutions that resulted in nearly the same hypervolume, but the solutions of MQEA were more diverse. Therefore, it can be concluded that MQEA performed better compared to MOPBIL for the DTLZ2 test function.