An Effective Genetic Algorithm with Uniform Crossover for Bi-objective Unconstrained Binary Quadratic Programming Problem

Chao Huo[1], Rong-Qiang Zeng[1,2(✉)], Yang Wang[3], and Ming-Sheng Shang[4]

[1] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, Sichuan, People's Republic of China
chaohuo0811@gmail.com

[2] School of Mathematics, Southwest Jiaotong University, Chengdu 610031, Sichuan, People's Republic of China
zrq@home.swjtu.edu.cn

[3] School of Management, Northwestern Polytechnical University, Xi'an 710072, Shanxi, People's Republic of China
sparkle.wy@gmail.com

[4] Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, People's Republic of China
msshang@cigit.ac.cn

Abstract. The unconstrained binary quadratic programming problem is one of the most studied NP-hard problem with its various practical applications. In this paper, we propose an effective multi-objective genetic algorithm with uniform crossover for solving bi-objective unconstrained binary quadratic programming problem. In this algorithm, we integrate the uniform crossover within the hypervolume-based multi-objective optimization framework for further improvements. The computational studies on 10 benchmark instances reveal that the proposed algorithm is very effective in comparison with the original multi-objective optimization algorithms.

Keywords: Multi-objective optimization · Hypervolume contribution · Genetic Algorithm · Uniform crossover · Unconstrained Binary Quadratic Programming problem

1 Introduction

The Unconstrained Binary Quadratic Programming (UBQP) problem is a classic NP-hard combinatorial problem with a number of applications [15]. The UBQP problem is extended into multi-objective case in [13], where the multiple objectives are to be maximized simultaneously, then the multi-objective UBQP problem can mathematically be formulated as follows [14]:

$$f_k(x) = x'Q^k x = \sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}^k x_i x_j \tag{1}$$

H. Yin et al. (Eds.): IDEAL 2016, LNCS 9937, pp. 58–67, 2016.
DOI: 10.1007/978-3-319-46257-8_7

where $Q^k = (q^k_{ij})$ is an $n \times n$ matrix of constants and x is an n-vector of binary (zero-one) variables, i.e., $x_i \in \{0, 1\}$ $(i = 1, \ldots, n)$, $k \in \{1, \ldots, m\}$.

UBQP is notable for its ability to formulate a wide range of other practical problems in different fields, such as machine scheduling [1], traffic management [8], computer aided design [12], financial analysis [16], etc. Many heuristic and metaheuristic algorithms have been proposed to tackle UBQP in the literature [11], such as simulated annealing [2], scatter search [3], tabu search [9], memetic algorithms [17], etc.

Moreover, A. Liefooghe et al. [13] proposed a hybrid metaheuristic algorithm to solve the multi-objective UBQP problem, which combines an elitist evolutionary multi-objective optimization algorithm with an effective single-objective tabu search procedure based on the scalarizing function. In [14], they proposed the multi-objective local search algorithms with three different strategies to solve the bi-objective UBQP problem more efficiently.

In this paper, we propose a multi-objective genetic algorithm to solve the bi-objective UBQP problem. This algorithm integrates the uniform crossover within the hypervolume-based multi-objective optimization framework for further improvements. Actually, there are two main components: hypervolume contribution selection and genetic algorithm with uniform crossover. The hypervolume contribution selection procedure iteratively improve the Pareto approximation set until it can not be improved any more. Then, the uniform crossover is used to further improve the whole quality of the Pareto approximation set.

This paper is organized as follows. In the next section, we introduce the basic notations and definitions of multi-objective optimization. In Sect. 3, we briefly review the previous work related to the uniform crossover and the multi-parent crossover. Afterwards, we present our proposed multi-objective genetic algorithm in Sect. 4. Section 5 provides the computational results and analyze the behavior of the proposed algorithm. Finally, the conclusions are provided in the last section.

2 Multi-objective Optimization

In this section, we present the basic notations and definitions of multi-objective optimization. Let X denote the search space of the optimization problem under consideration and Z the corresponding objective space. Without loss of generality, we assume that $Z = \Re^n$ and that all n objectives are to be maximized. Each $x \in X$ is assigned exactly one objective vector $z \in Z$ on the basis of a vector function $f : X \rightarrow Z$ with $z = f(x)$, and the mapping f defines the evaluation of a solution $x \in X$ [7]. Actually, we are often interested in those solutions that are Pareto optimal with respect to f. The relation $x_1 \succ x_2$ means that the solution x_1 is *preferable* to x_2. The dominance relation between two solutions x_1 and x_2 is often defined as follows [7]:

Definition 1 (Pareto Dominance). A decision vector x_1 is said to dominate another decision vector x_2 (written as $x_1 \succ x_2$), if $f_i(x_1) \geq f_i(x_2)$ for all $i \in \{1, \ldots, n\}$ and $f_j(x_1) > f_j(x_2)$ for at least one $j \in \{1, \ldots, n\}$.

Definition 2 (Pareto Optimal Solution). $x \in X$ is said to be Pareto optimal if and only if there does not exist another solution $x' \in X$ such that $x' \succ x$.

Definition 3 (Pareto Optimal Set). S is said to be a Pareto optimal set if and only if S is composed of all the Pareto optimal solutions.

Definition 4 (Non-Dominated Solution). $x \in S$ $(S \subset X)$ is said to be non-dominated if and only if there does not exist another solution $x' \in S$ such that $x' \succ x$.

Definition 5 (Non-Dominated Set). S is said to be a non-dominated set if and only if any two solutions $x_1 \in S$ and $x_2 \in S$ such that $x_1 \not\succ x_2$ and $x_2 \not\succ x_1$.

Since there does not exist the total order relation among all the solutions in multi-objective optimization, the aim is to generate the Pareto optimal set, which keeps the best compromise among all the objectives. Nevertheless, in most cases, it is impossible to generate the Pareto optimal set in a reasonable time. Therefore, we are interested in finding a non-dominated set which is as close to the Pareto optimal set as possible, and the whole goal is often to identify a good Pareto approximation set.

3 Previous Work

The Genetic Algorithms (GA) have a good potential of solving many different combinatorial optimization problems, which are often integrated into the hybrid metaheuristics as an important part for further improvements [7]. In this section, we provide a literature review of the studies related to the uniform crossover and the multi-parent crossover.

U. Benlic et al. [6] proposed a powerful population-based memetic algorithm for solving quadratic assignment problem, which integrates an effective local optimization algorithm within the evolutionary computing framework based on a uniform crossover. In this algorithm, the uniform crossover is used to further enforce the search capacity of the proposed algorithm. The extensive computational studies reveal that their proposed algorithm is very competitive.

E. D. Paolo et al. [10] proposed an efficient genetic algorithm with uniform crossover for solving the multi-objective airport gate assignment problem. In this algorithm, a new defined uniform crossover operator is used to generate high-quality offsprings, which is crucial to the successful implementations. The extensive simulation studies illustrate the advantages of the proposed GA scheme with the uniform crossover operator.

Zhipeng Lü et al. [15] presented a multi-parent hybrid genetic-tabu algorithm for dealing with unconstrained binary quadratic programming problem, which incorporates the tabu search procedure into the genetic algorithm framework. In this algorithm, a multi-parent crossover operator called MSX is jointly employed with the conventional uniform crossover to generate diversified new solutions. The computational results on 10 large benchmark instances indicate that the proposed algorithm is very effective.

Yang Wang et al. [18] integrated four multi-parent crossover operators (called MSX, Diagonal, U-Scan and OB-Scan) within the memetic algorithm framework for dealing with unconstrained binary quadratic programming problem. Their proposed algorithms apply these crossover operators to further improve the results generated by the tabu search procedure. The experimental results and the analysis on the behavior of the algorithm provide the evidences and the insights as to key role of the crossover operators.

4 Multi-Objective Genetic Algorithm

The Multi-Objective Genetic Algorithm (MOGA) is proposed to solve the bi-objective UBQP problem, which is composed of two main components: hypervolume contribution selection and genetic algorithm with the uniform crossover. The general scheme of this algorithm is described in Algorithm 1.

Algorithm 1. Multi-Objective Genetic Algorithm

```
Input: $N$ (Population size)
Output: $A$: (Pareto approximation set)
Step 1 - Initialization: $P \leftarrow N$ randomly generated solutions
Step 2: $A \leftarrow \Phi$
Step 3 - Fitness Assignment: Compute a fitness value for each solution $x \in P$
Step 4:
while Running time is not reached do
  repeat
    1) Hypervolume Contribution Selection: $x \in P$
  until all neighbors of $x \in P$ are explored
    2) $A \leftarrow$ Non-dominated solutions of $A \bigcup P$
    3) Genetic Algorithm: $y \in A$
end while
Step 5: Return $A$
```

In MOGA, all the solutions in an initial population are randomly generated, i.e., each variable of one solution is randomly assigned a value 0 and 1 (Step 1). Then, each solution is computed a fitness value by the Hypervolume Contribution (HC) indicator defined in [5] (Step 3).

After realizing the fitness assignment, we optimize the initial population with the hypervolume contribution selection procedure in order to obtain a Pareto approximation set. Afterwards, we apply the uniform crossover operator to generate the offsprings, in order to update the Pareto approximation set A for further improvements.

4.1 Hypervolume Contribution Selection

In order to generate a set of efficient individuals used for generating high-quality offsprings with the uniform crossover operator, we apply the Hypervolume

Algorithm 2. Hypervolume Contribution Selection (HCS)

```
Steps:
    1) $x^* \leftarrow$ one randomly chosen unexplored neighbors of $x$
    2) $P \leftarrow P \bigcup x^*$
    3) compute $x^*$ fitness: $HC(x^*, P)$
    4) update all $z \in P$ fitness values
    5) $\omega \leftarrow$ worst solution in $P$
    6) $P \leftarrow P \backslash \{\omega\}$
    7) update all $z \in P$ fitness values
    8) if $\omega \neq x^*$, Progress $\leftarrow$ True
```

Contribution Selection (HCS) procedure presented in Algorithm 2 to the initial population.

For the UBQP problem, we flip the value 0 (or 1) of the k^{th} variable of each solution $x \in P$ to 1 (or 0) to obtain a new solution x^* as the neighbor of x. Then, we compute the objective function values of this new solution with the fast incremental neighborhood evaluation formula [15] below:

$$\Delta_i = (1 - 2x_i)(q_{ii} + \sum_{j \in N, j \neq i, x_j = 1} q_{ij}) \tag{2}$$

Actually, the move value Δ_i can be computed in linear time, which makes the local search procedure more efficiently.

In the HCS procedure, one solution x^*, which is one of the unexplored neighbors of x in the population P, is assigned to a fitness value by the HC indicator. If x^* is dominated, the fitness values of all the solutions in P keep unchanged. If x^* is non-dominated, we need to update the fitness values of non-dominated neighbors of x^* in the objective space.

Afterwards, the solution ω with the worst fitness value is removed from the population P. If ω is dominated, the fitness values of the other solutions keep unchanged. If ω is non-dominated, the fitness values of the non-dominated neighbors of ω need to be updated. The whole procedure will repeat until the termination criterion is satisfied.

4.2 Genetic Algorithm

The uniform crossover operator is widely used to solve many combinatorial optimization problems, such as quadratic assignment problem [6], gate assignment problem [10], single-objective UBQP problem [15], and so on. In this work, we employ the uniform crossover operator to improve the Pareto approximation set A generated by the HCS procedure. The exact steps are given in Algorithm 3.

In the genetic algorithm, we randomly select two non-dominated solutions as two parents from the Pareto approximation set A. For the crossover process, we employ the standard uniform crossover operator to recombine the selected parents.

Algorithm 3. Genetic Algorithm (GA)

```
Steps:
    1) randomly select two parents: $y_i \in A$ and $y_j \in A$
    2) generate an offspring $z \leftarrow$ Crossover$(y_i, y_j)$
    3) $z' \leftarrow$ Mutation$(z)$
    4) $A \leftarrow$ HCS$(z')$
```

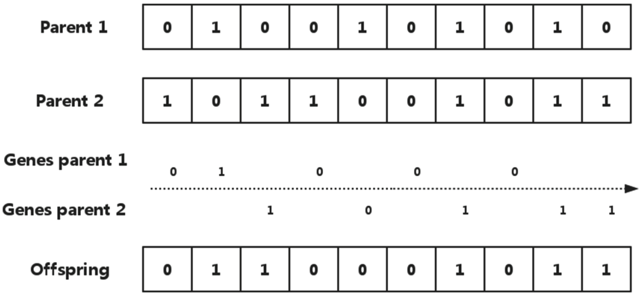

Fig. 1. An example of the uniform crossover for UBQP.

Furthermore, the elements of the parents are scanned from left to right, and each element in the offspring keeps the value x_i (0 or 1) in either of these two parents with the equal probability. The unassigned variables in the offspring is given 0 or 1 randomly. An example of this crossover process is illustrated in Fig. 1.

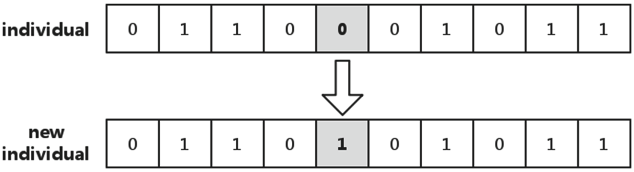

Fig. 2. An example of the mutation for UBQP.

Afterwards, we apply the mutation procedure to the generated offspring by randomly flipping the value of one variable to the opposite value (0 or 1), in order to generate a new offspring. An example of the mutation procedure is illustrated in Fig. 2. Then, we insert this new offspring into the Pareto approximation set A with the HC indicator for further improvements.

5 Experiments

In this section, we first present the parameter settings for the MOGA algorithm. Then, we introduce a performance assessment protocol used to evaluate the effectiveness of multi-objective optimization algorithm. Finally, we provide the computational results and the performance analysis.

5.1 Parameters Settings

The MOGA algorithm is programmed in C++ and compiled using Dev-C++ 5.0 compiler on a PC running Windows 7 with Core 2.50 GHz CPU and 4 GB RAM. In order to evaluate the efficiency of our proposed algorithm, we carry out the experiments on 10 benchmark instances of bi-objective UBQP problem, which are generated by the tools provided in [13].

Besides, this algorithm requires to set a few parameters, we mainly discuss two important ones: the running time and the population size. The precise information about the instances and the parameter settings is presented in Table 1.

Table 1. Parameter settings used for bi-objective UBQP instances: instance dimension (D), population size (P) and running time (T).

	Dimension (D)	Population (P)	Time (T)
bubqp_1000_01	1000	10	$100''$
bubqp_1000_02	1000	10	$100''$
bubqp_2000_01	2000	20	$200''$
bubqp_2000_02	2000	20	$200''$
bubqp_3000_01	3000	30	$300''$
bubqp_3000_02	3000	30	$300''$
bubqp_4000_01	4000	40	$400''$
bubqp_4000_02	4000	40	$400''$
bubqp_5000_01	5000	50	$500''$
bubqp_5000_02	5000	50	$500''$

5.2 Performance Assessment Protocol

In this paper, we evaluate the effectiveness of multi-objective optimization algorithms using a test procedure that has been undertaken with the performance assessment package provided by Zitzler et al.[1].

The quality assessment protocol works as follows: we first create a set of 20 runs with different initial populations for each algorithm and each benchmark instance. Afterwards, we calculate the reference set PO^* in order to determine the quality of k different sets $A_0 \ldots A_{k-1}$ of non-dominated solutions. Furthermore, we define a reference point $z = [w_1, w_2]$, where w_1 and w_2 represent the worst values for each objective function in $A_0 \cup \cdots \cup A_{k-1}$. Then, the evaluation of a set A_i of solutions can be determined by finding the hypervolume difference between A_i and PO^* [19], and this hypervolume difference has to be as close as possible to zero.

[1] http://www.tik.ee.ethz.ch/pisa/assessment.html.

5.3 Computational Results

In this subsection, we provide the computational results obtained by our proposed MOGA algorithm, the indicator-based multi-objective local search algorithm (IBMOLS) proposed in [4] and the hypervolume-based multi-objective local search algorithm (HBMOLS) proposed in [5].

The computational results are summarized in Table 2. Each line in this table contains a value both **in bold** and **in grey box**, which is the best result obtained on the considered instance. The another two values refer to that the corresponding algorithms are statistically outperformed by the algorithm which obtains the best result (with a confidence level greater than 95 %).

Table 2. The computational results on bi-objective UBQP problem obtained by the algorithms: IBMOLS, HBMOLS and MOGA

Instance	Algorithm		
	IBMOLS	HBMOLS	MOGA
bubqp_1000_01	0.527656	0.510992	**0.179459**
bubqp_1000_02	0.184199	0.116860	**0.116035**
bubqp_2000_01	0.560875	0.516558	**0.220095**
bubqp_2000_02	0.618150	0.634531	**0.112132**
bubqp_3000_01	0.570740	0.557831	**0.213661**
bubqp_3000_02	0.639882	0.653772	**0.134274**
bubqp_4000_01	0.533230	0.568231	**0.103679**
bubqp_4000_02	0.629849	0.627149	**0.069898**
bubqp_5000_01	0.506434	0.541808	**0.136980**
bubqp_5000_02	0.630504	0.636295	**0.044327**

According to Table 2, we can see that all the best results are obtained by MOGA, which statistically outperforms the algorithms IBMOLS and HBMOLS on all the instances. Moreover, the most significant result is achieved on the instance bubqp_5000_02, where the average hypervolume difference value obtained by MOGA is much smaller (about 15 times) than the values obtained by IBMOLS and HBMOLS.

Compared with IBMOLS and HBMOLS, we can see the evident contribution of the uniform crossover in MOGA. We assume that IBMOLS and HBMOLS could be often trapped in the local optima after a certain amount of running time, since they only focus on the local search procedure without any strategy used for jumping out of the local optima.

For MOGA, the uniform crossover is used to generate two new solutions, which are inserted into the Pareto approximation set for further improvements. In fact, these new generated solutions make MOGA have a chance to jump out of the local optima. The computational results indicate the uniform crossover evidently improve the quality of the Pareto approximation set. Therefore, MOGA has a better performance on all the instances.

6 Conclusion

In this paper, we have presented a simple and effective multi-objective genetic algorithm for the well-known bi-objective unconstrained binary quadratic programming problem. The proposed algorithm has combined the hypervolume-based optimization with the standard uniform crossover for the further improvements on the Pareto approximation set. We have evaluated this algorithm on the set of 10 benchmark instances, and the computational results indicate that the MOGA algorithm performs very well on these instances.

Acknowledgments. The work in this paper was supported by the Fundamental Research Funds for the Central Universities (Grant No. A0920502051408-25), supported by the Research Foundation for International Young Scientists of China (Grant No. 61450110443), supported by the Scientific Research Foundation for the Returned Overseas Chinese Scholars (Grant No. 2015S03007) and supported by National Natural Science Foundation of China (Grant No. 61370150 and 71501157).

References

1. Alidaee, B., Kochenberger, G.A., Ahmadian, A.: 0–1 quadratic programming approach for the optimal solution of two scheduling problems. Int. J. Syst. Sci. **25**, 401–408 (1994)
2. Alkhamis, T.M., Hasan, M., Ahmed, M.A.: Simulated annealing for the unconstrained binary quadratic pseudo-Boolean function. Eur. J. Oper. Res. **108**, 641–652 (1998)
3. Amini, M., Alidaee, B., Kochenberger, G.: A scatter search approach to unconstrained quadratic binary programs. New Meth. Optim. **108**, 317–330 (1999)
4. Basseur, M., Liefooghe, A., Le, K., Burke, E.: The efficiency of indicator-based local search for multi-objective combinatorial optimisation problems. J. Heuristics **18**(2), 263–296 (2012)
5. Basseur, M., Zeng, R.-Q., Hao, J.-K.: Hypervolume-based multi-objective local search. Neural Comput. Appl. **21**(8), 1917–1929 (2012)
6. Benlic, U., Hao, J.-K.: Memetic search for the quadratic assignment problem. Expert Syst. Appl. **42**, 584–595 (2015)
7. Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation). Springer, New York (2007)
8. Gallo, G., Hammer, P.: Quadratic knapsack problems. Math. Program. **12**, 132–149 (1980)

9. Glover, F., Kochenberger, G., Alidaee, B.: Adaptive memory tabu search for binary quadratic programs. Manag. Sci. **44**, 336–345 (1998)
10. Hu, X.-B., Paolo, E.D.: An efficient genetic algorithm with uniform crossover for the multi-objective airport gate assignment problem. Multi Objective Memetic Algorithm **171**, 71–89 (2009)
11. Kochenberger, G., Hao, J.-K., Glover, F., Lewis, M., Lü, Z., Wang, H., Wang, Y.: The unconstrained binary quadratic programming problem: a survey. J. Comb. Optim. **28**, 58–81 (2014)
12. Krarup, J.: Computer aided layout design. Math. Program. Study **9**, 75–94 (1978)
13. Liefooghe, A., Verel, S., Hao, J.-K.: A hybrid metaheuristic for multiobjective unconstrained binary quadratic programming. Appl. Soft Comput. **16**, 10–19 (2014)
14. Liefooghe, A., Verel, S., Paquete, L., Hao, J.-K.: Experiments on local search for bi-objective unconstrained binary quadratic programming. In: Proceedings of the 8th International Conference on Evolutionary Multi-criterion Optimization (EMO 2015), Guimarães, Portugal, pp. 171–186 (2015)
15. Lü, Z., Hao, J.-K., Glover, F.: A study of memetic search with multi-parent combination for UBQP. In: Prodeedings of the 10th International Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP 2010), Istanbul, Turkey, pp. 154–165 (2010)
16. McBride, R.D., Yormark, J.S.: An implicit enumeration algorithm for quadratic integer programming. Manag. Sci. **26**, 282–296 (1980)
17. Merz, P., Katayama, K.: Memetic algorithms for the unconstrained binary quadratic programming problem. BioSystems **78**, 99–118 (2004)
18. Wang, Y., Lü, Z., Hao, J.-K.: A study of multi-parent crossover operators in a memetic algorithm. In: Prodeedings of the 11th International Conference on Parallel Problem Solving from Nature (PPSN XI), Krakow, Poland, pp. 556–565 (2010)
19. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. Evol. Comput. **3**, 257–271 (1999)