A Multi-parent Crossover Based Genetic Algorithm for Bi-Objective Unconstrained Binary Quadratic Programming Problem

Chao Huo[1], Rongqiang Zeng[1,2(✉)], Yang Wang[3], and Mingsheng 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. In this paper, we present a multi-parent crossover based genetic algorithm for the bi-objective unconstrained binary quadratic programming problem, by integrating the multi-parent crossover within the framework of hypervolume-based multi-objective optimization algorithm. The proposed algorithm employs a multi-parent crossover operator to generate the offspring solutions, which are used to further improve the quality of Pareto approximation set. Experimental results on 10 benchmark instances demonstrate the efficacy of our proposed algorithm compared with the original multi-objective optimization algorithms.

Keywords: Multi-objective optimization · Hypervolume contribution · Genetic algorithm · Multi-parent crossover · Unconstrained binary quadratic programming problem

1 Introduction

The Unconstrained Binary Quadratic Programming (UBQP) problem is one of the most studied NP-hard problem with its various practical applications. The multi-objective UBQP problem can be mathematically formulated as follows [15]:

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

where $f_k(x)$ ($k \in \{1, \ldots, m\}$) is the kth objective and to be maximized, $Q^k = (q_{ij}^k)$ is an $n \times n$ matrix of constants and x is an n-vector of binary (zero-one) variables, i.e., $x_i \in \{0, 1\}$ ($i = 1, \ldots, n$).

The formulation of UBQP is notable for its ability to represent a wide range of important combinatorial optimization problems, including traffic management [9], financial analysis [20], molecular conformation [25], cellular radio channel allocation [26], and so on. The literature reports a large number of heuristic and metaheuristic algorithms to deal with the UBQP problem [14], which include scatter search [2], directed local search [7], simulated annealing [1,13], evolutionary algorithms [6,17,22], tabu search [10,23,24], etc.

Moreover, Liefooghe et al. [15] first extended the single UBQP problem into the multi-objective case and proposed a hybrid metaheuristic algorithm to solve the multi-objective UBQP problem. In [16], they further proposed three versions of multi-objective local search algorithms with different search strategies to solve the bi-objective UBQP problem.

In the current paper, we study a multi-parent crossover based genetic algorithm for the bi-objective UBQP problem, which integrates a multi-parent crossover within the framework of hypervolume-based multi-objective optimization algorithm. The proposed algorithm consists of two main procedures: hypervolume contribution selection procedure and genetic algorithm with multi-parent crossover. The hypervolume contribution selection procedure iteratively improves the Pareto approximation set until it can not be improved any more. Then, the multi-parent crossover is used to further improve the entire quality of the Pareto approximation set.

The remaining part of the 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 describe our proposed multi-objective genetic algorithm with multi-parent crossover in Sect. 4. Section 5 is dedicated to the computational results and concluding remarks are given 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 $f : X \to Z$ with $z = f(x)$, and the mapping f defines the evaluation of a solution $x \in X$ [8].

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 [8]:

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 $x_1 \nsucc x_2$ and $x_2 \nsucc x_1$.

In fact, there does not exist the total order relation among all the solutions in multi-objective optimization. Thus, 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 overall goal is often to identify a good Pareto approximation set.

3 Related Work

The uniform crossover and its variants are usually integrated into the hybrid metaheuristics as an important part for further improvement, which are widely used to solve many combinatorial optimization problems, such as quadratic assignment problem [5], gate assignment problem [12], single-objective UBQP problem [14]. In this section, we briefly review the literature on solving the UBQP problem with the uniform crossover and the multi-parent crossover.

Merz and Freisleben [21] proposed a hybrid genetic algorithm, which incorporates a simple local search into the traditional genetic algorithm. A variant of uniform crossover is used to generate offspring solutions based on Hamming distance from the parents. Computational results on the UBQP problem show that the proposed algorithm is sufficient to find best known results for the problem instances with less than 200 variables, but not very effective on the problem instances with large size.

Lodi et al. [17] presented an effective evolutionary method for solving the UBQP problem. In this algorithm, a uniform crossover operator is used to produce the offspring solutions, where the variables with common values in parental solutions are temporarily fixed in the current round of local search. Computational results on the problem instances with up to 500 variables show the attractiveness and the effectiveness of the proposed method, especially on the small problem instances.

Lü et al. [18] proposed a hybrid metaheuristic approach for solving the UBQP problem, which incorporates a tabu search procedure into the framework of evolutionary algorithms. In this algorithm, a uniform crossover operator and a

diversification-guided combination operator are used to generate offspring solutions in order to further enforce the search capacity of the proposed algorithm. The extensive computational studies on problem instances with up to 7000 variables reveal that their proposed algorithm is very competitive.

Wang et al. [27] 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-parent Crossover Based Genetic Algorithm

The Multi-Parent Crossover based Genetic Algorithm (MPCGA) is proposed to solve the bi-objective UBQP problem, which consists of two main procedures: hypervolume contribution selection and genetic algorithm with the multi-parent crossover. The general architecture of the MPCGA algorithm is described in Algorithm 1.

Algorithm 1. Multi-Parent Crossover based Genetic Algorithm

    Input: N (Population size)
    Output: A: (Pareto approximation set)
    Step 1: $P \leftarrow N$ randomly generated individuals
    Step 2: $A \leftarrow \Phi$
    Step 3: Calculate a fitness value for each individual $x \in P$ with HC indicator
    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: $z \in A$
    end while
    Step 5: Return A

In MPCGA, all the individuals in an initial population are randomly generated, i.e., each variable of an individual is randomly assigned a value 0 and 1 (Step 1). Then, each individual is calculated a fitness value by the Hypervolume Contribution (HC) indicator defined in [4] (Step 3) and optimized by the hypervolume contribution selection procedure. Afterwards, we employ the multi-parent crossover operator proposed in [19] to produce the offsprings, in order to further improve the quality of Pareto approximation set.

4.1   Hypervolume Contribution Selection

After the fitness assignment for each individual, we apply the Hypervolume Contribution Selection (HCS) procedure [4] presented in Algorithm 2 to the initial population, in order to generate a set of efficient individuals.

Algorithm 2. Hypervolume Contribution Selection (HCS)

Steps:

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

In the HCS procedure, an individual 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 individuals in P remain unchanged. If x^* is non-dominated, we need to update the fitness values of non-dominated neighbors of x^*.

Actually, the neighborhood of UBQP is usually defined by the simple one-flip move, which flips the value 0 (or 1) of the kth variable of each solution $x \in P$ to 1 (or 0) to obtain a new individual x^* as the neighbor of x [11]. Then, we calculate the objective function values of this new neighbor with the fast incremental neighborhood evaluation formula [19] below:

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

Afterwards, the individual ω with the worst fitness value is deleted from the population P. If ω is dominated, the fitness values of the other individuals remain unchanged. If ω is non-dominated, the fitness values of the non-dominated neighbors of ω need to be updated. The HCS procedure will repeat until the termination criterion is satisfied.

4.2   Genetic Algorithm

The main idea of uniform crossover is to assign values to the variables of offspring that represent assignments made in common by both parents, and to randomly assign values to remaining variables of the offspring solution [18]. Based on this idea, a multi-parent crossover operator called MSX is proposed to solve the UBQP problem [19]. In this work, we employ the MSX crossover operator to

---

Algorithm 3. Genetic Algorithm (GA)

Steps:
    1) randomly select a subset of individuals E from A
    2) $z \leftarrow$ MSX_Crossover_Operator(E)
    3) $A \leftarrow$ HCS(z)

---

improve the Pareto approximation set A generated by the HCS procedure. The exact steps are presented in Algorithm 3.

In our algorithm, we randomly select a set E ($|E| = s$) of non-dominated individuals from the Pareto approximation set A. Let $E = \{x^{(1)}, x^{(2)}, \ldots, x^{(s)}\}$, where $x^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}\}$ and the individuals in E are ordered in terms of their fitness values, i.e., $x^{(1)}$ is the best individual in E and $x^{(s)}$ is the worst individual in E. As suggested in [19], we set s to be a random number between 4 and 8. Then, the MSX crossover operator is defined below [27]:

MSX Crossover Operator: we define a weight $w(i)$ for the individual $x^{(i)}$ and a strength value $Strength(j)$ for variables x_j as: $w(i) = 1/\sum sum(i) = 1/\sum_{j=1}^{n} x_j^{(i)}$ and $Strength(j) = \sum_{i=1}^{s} w(i)x_j^{(i)}$.

The value $Strength(j)$ gives a relative indication of the tendency of the individuals in E to favor $x_j = 1$ or $x_j = 0$. Furthermore, we take an advantage of the $sum(i)$ values over E to get a value for the number of x_j components that should be 1 in an average individual, denoted by $Avg = \sum_{i=1}^{s} sum(i)/s$ [19]. Then, the variables with the first Avg largest $Strength$ values receive assignment 1 and other variables receive assignment 0. Afterwards, a new offspring is generated and inserted into the Pareto approximation set A with the HC indicator for further improvement.

5   Experiments

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 [15]. The MPCGA 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.

5.1   Parameters Settings

The MPCGA algorithm requires to set a few parameters, we mainly discuss two important ones: the running time and the population size. The exact 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 efficiency 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^* [28], and this hypervolume difference has to be as close as possible to zero.

5.3   Computational Results

In this subsection, we present the computational results obtained by our proposed MPCGA algorithm, the indicator-based multi-objective local search algorithm (IBMOLS) proposed in [3] and the hypervolume-based multi-objective local search algorithm (HBMOLS) proposed in [4].

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 values both **in italic** and **in bold** mean that the corresponding algorithms are **not** 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 MPCGA

Instance	Algorithm		
	IBMOLS	HBMOLS	MPCGA
bubqp_1000_01	0.175422	*0.102101*	**0.100887**
bubqp_1000_02	0.125044	0.093831	**0.091997**
bubqp_2000_01	0.102348	0.100722	**0.094091**
bubqp_2000_02	0.122507	0.094502	**0.093091**
bubqp_3000_01	0.103768	0.102531	**0.066613**
bubqp_3000_02	0.105664	*0.093186*	**0.093022**
bubqp_4000_01	0.120652	*0.104065*	**0.103641**
bubqp_4000_02	0.931491	0.930591	**0.901945**
bubqp_5000_01	0.944998	0.959954	**0.287479**
bubqp_5000_02	0.115645	*0.101692*	**0.101490**

From Table 2, we can observe that all the best results are obtained by MPCGA, and the most significant result is achieved on the instance bubqp_5000_01, where the average hypervolume difference value obtained by MPCGA is much smaller than the values obtained by IBMOLS and HBMOLS.

However, the values on the instances (bubqp_1000_01, bubqp_3000_02, bubqp_4000_01 and bubqp_5000_02) obtained by HBMOLS are not statistically outperformed by MPCGA. Actually, the new offsprings generated by the MSX crossover operator evidently improve the entire quality of Pareto approximation set, which makes the MPCGA algorithm have a chance to search the high-quality individuals in the objective space. Thus, MPCGA has a better performance on all the instances.

6   Conclusion

In this paper, integrating the multi-parent crossover within the hypervolume-based multi-objective optimization algorithm to further improve the overall quality of Pareto approximation set, the MPCGA is proposed to deal with the bi-objective unconstrained binary quadratic programming problem. The computational results of MPCGA on 10 benchmark instances have shown the feasibility of the improvements and the effectiveness of MPCGA for the bi-objective UBQP problem.

References

1. 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)
2. Amini, M., Alidaee, B., Kochenberger, G.: A scatter search approach to unconstrained quadratic binary programs. New Methods Optim. **108**, 317–330 (1999)
3. 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)
4. Basseur, M., Zeng, R.-Q., Hao, J.-K.: Hypervolume-based multi-objective local search. Neural Comput. Appl. **21**(8), 1917–1929 (2012)
5. Benlic, U., Hao, J.-K.: Memetic search for the quadratic assignment problem. Expert Syst. Appl. **42**, 584–595 (2015)
6. Borgulya, I.: An evolutionary algorithm for the binary quadratic problems. Adv. Soft Comput. **2**, 3–16 (2005)
7. Boros, E., Hammer, P.L., Tavarse, G.: Local search heuristics for quadratic unconstrained binary optimization (QUBO). J. Heuristics **13**, 99–132 (2007)
8. Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-objective Problems (Genetic and Evolutionary Computation). Springer-Verlag New York Inc., Secaucus (2007)
9. Gallo, G., Hammer, P., Simeone, B.: Quadratic knapsack problems. Math. Program. **12**, 132–149 (1980)
10. Glover, F., Kochenberger, G., Alidaee, B.: Adaptive memory tabu search for binary quadratic programs. Manag. Sci. **44**, 336–345 (1998)
11. Hoos, H., Stützle, T.: Stochastic Local Search Foundations and Applications. Morgan Kaufmann, San Francisco (2004)
12. 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)
13. Katayama, K., Narihisa, H.: Performance of simulated annealing-based heuristic for the unconstrained binary quadratic programming problem. Eur. J. Oper. Res. **134**, 103–119 (2001)
14. 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)
15. Liefooghe, A., Verel, S., Hao, J.-K.: A hybrid metaheuristic for multiobjective unconstrained binary quadratic programming. Appl. Soft Comput. **16**, 10–19 (2014)
16. Liefooghe, A., Verel, S., Paquete, L., Hao, J.-K.: Experiments on local search for bi-objective unconstrained binary quadratic programming. In: Gaspar-Cunha, A., Henggeler Antunes, C., Coello, C.C. (eds.) EMO 2015. LNCS, vol. 9018, pp. 171–186. Springer, Heidelberg (2015). doi:10.1007/978-3-319-15934-8_12
17. Lodi, A., Allemand, K., Liebling, T.M.: An evolutionary heuristic for quadratic 0–1 programming. Eur. J. Oper. Res. **119**(3), 662–670 (1999)

18. Lü, Z., Glover, F., Hao, J.-K.: A hybrid metaheuristic approach to solving the UBQP problem. Eur. J. Oper. Res. **207**, 1254–1262 (2010)
19. Lü, Z., Hao, J.-K., Glover, F.: A study of memetic search with multi-parent combination for UBQP. In: Cowling, P., Merz, P. (eds.) EvoCOP 2010. LNCS, vol. 6022, pp. 154–165. Springer, Heidelberg (2010). doi:10.1007/978-3-642-12139-5_14
20. McBride, R.D., Yormark, J.S.: An implicit enumeration algorithm for quadratic integer programming. Manag. Sci. **26**, 282–296 (1980)
21. Merz, P., Freisleben, B.: Genetic algorithms for binary quadratic programming. In: Proceedings of The 1st Genetic and Evolutionary Computation Conference (GECCO 1999), pp. 417–424, Orlando, Florida, USA (1999)
22. Merz, P., Katayama, K.: Memetic algorithms for the unconstrained binary quadratic programming problem. BioSystems **78**, 99–118 (2004)
23. Palubeckis, G.: Multistart tabu search strategies for the unconstrained binary quadratic optimization problem. Ann. Oper. Res. **131**, 259–282 (2004)
24. Palubeckis, G.: Iterated tabu search for the unconstrained binary quadratic optimization problem. Informatica **17**(2), 279–296 (2006)
25. Rosen, J.B., Phillips, A.T.: A quadratic assignment formulation of the molecular conformation problem. J. Glob. Optim. **4**, 229–241 (1994)
26. Sutter, A., Chardaire, P.: A decomposition method for quadratic zero-zero programming. Manag. Sci. **41**(4), 704–712 (1994)
27. Wang, Y., Lü, Z., Hao, J.-K.: A study of multi-parent crossover operators in a memetic algorithm. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN 2010. LNCS, vol. 6238, pp. 556–565. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15844-5_56
28. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. Evol. Comput. **3**, 257–271 (1999)