

# Average value of solutions for the bipartite boolean quadratic programs and rounding algorithms\*

Abraham P. Punnen<sup>†</sup>, Piyashat Sripratak<sup>‡</sup> and Daniel Karapetyan<sup>§</sup>

*Department of Mathematics, Simon Fraser University Surrey, 250-13450 102nd AV, Surrey, British Columbia, V3T 0A3, Canada*

## Abstract

We consider domination analysis of approximation algorithms for the bipartite boolean quadratic programming problem (BBQP) with  $m + n$  variables. A closed form formula is developed to compute the average objective function value  $\mathcal{A}$  of all solutions in  $O(mn)$  time. However, computing the median objective function value of the solutions is shown to be NP-hard. Also, we show that any solution with objective function value no worse than  $\mathcal{A}$  dominates at least  $2^{m+n-2}$  solutions and this bound is the best possible. Further, we show that such a solution can be identified in  $O(mn)$  time and hence the dominance ratio of this algorithm is at least  $\frac{1}{4}$ . We then show that for any fixed rational number  $\alpha > 1$ , no polynomial time approximation algorithm exists for BBQP with dominance ratio larger than  $1 - 2^{\frac{(1-\alpha)}{\alpha}(m+n)}$ , unless P=NP. We then analyze some powerful local search algorithms and show that they can get trapped at a local maximum with objective function value less than  $\mathcal{A}$ . One of our approximation algorithms has an interesting rounding property which provides a data dependent lower bound on the optimal objective function value. A new integer programming formulation of BBQP is also given and computational results with our rounding algorithms are reported.

**Keywords:** quadratic programming, boolean variables, heuristics, worst-case analysis, domination analysis.

## 1 Introduction

The *bipartite boolean quadratic programming problem* (BBQP) is to

$$\begin{aligned} & \text{Maximize } f(x, y) = x^T Q y + c x + d y \\ & \text{subject to } x \in \{0, 1\}^m, y \in \{0, 1\}^n \end{aligned}$$

where  $Q = (q_{ij})$  is an  $m \times n$  matrix,  $c = (c_1, c_2, \dots, c_m)$  is a row vector in  $R^m$ , and  $d = (d_1, d_2, \dots, d_n)$  is a row vector in  $R^n$ . Without loss of generality, we assume that  $m \leq n$ .

---

\*This research work was supported by an NSERC Discovery grant and an NSERC discovery accelerator grant awarded to Abraham P Punnen.

<sup>†</sup>apunnen@sfu.ca

<sup>‡</sup>psriprat@sfu.ca

<sup>§</sup>daniel.karapetyan@gmail.com

BBQP has applications in data mining, clustering and bioinformatics [35], approximating a matrix by a rank-one binary matrix [10, 32], mining discrete patterns in binary data [21, 32], solving fundamental graph theoretic optimization problems such as maximum weight biclique [4, 34], maximum weight cut problem on a bipartite graph [26], maximum weight induced subgraph of a bipartite graph [26], and computing approximations to the cut-norm of a matrix [3].

BBQP is closely related to the well-studied *boolean quadratic programming problem* (BQP) [7, 12, 39]:

$$\begin{aligned} & \text{Maximize } f(x) = x^T Q' x + c' x \\ & \text{subject to } x \in \{0, 1\}^n, \end{aligned}$$

where  $Q'$  is an  $n \times n$  matrix and  $c'$  is a row vector in  $R^n$ . BBQP can be formulated as a BQP with  $n+m$  variables [26] and hence the resulting cost matrix will have dimension  $(n+m) \times (n+m)$ . This increase in problem size is not desirable especially for large scale problems. On the other hand, we can formulate BQP as a BBQP by choosing

$$Q = Q' + 2MI, \quad c = \frac{1}{2}c' - Me \quad \text{and} \quad d = \frac{1}{2}c' - Me, \quad (1)$$

where  $I$  is an  $n \times n$  identity matrix,  $e \in R^n$  is an all one row vector and  $M$  is a very large number [26]. Thus, BBQP is a proper generalization of BQP which makes the study of BBQP further interesting. An instance of BBQP is completely defined by the matrix  $Q$  and vectors  $c$  and  $d$  and hence it is represented by  $\mathcal{P}(Q, c, d)$ . Thus,  $\mathcal{P}(Q, 0, 0)$  represents a BBQP with no terms  $cx$  or  $dy$  in the objective function. Such an instance is referred to as a *homogeneous* BBQP. Relationships between BBQP and its homogeneous version are considered in [26].

BBQP is trivial if the entries of  $Q, c$  and  $d$  are either all positive or all negative. BBQP is known to be NP-hard [25] since the maximum weight biclique problem (MWBP) is a special case of it. Approximation hardness results for MWBP are established by Ambuhl et al. [4] and Tan [34]. Performance ratio for approximation algorithms for some special cases of BBQP are discussed by Alon and Naor [3] and Raghavendra and Steurer [27]. Results extensive experimental analysis of algorithms for BBQP are reported by Karapetyan and Punnen [19] and Glover et al. [13]. Punnen, Sripratak, and Karapetyan [26] studied BBQP and identified various polynomially solvable special cases. Various classes of valid inequalities and facet defining inequalities for the polytope associated with BBQP are obtained by Sripratak and Punnen [33].

Worst case analysis of approximation algorithms (heuristics) are carried out normally through the measure of performance ratio [37]. Other important measures include differential ratio [8], dominance ratio [11, 17], dominance number [40, 11], comparison to average value of solutions [5, 36, 31, 30, 28] etc. Our focus in this paper is on domination analysis and average value based analysis of approximation algorithms for BBQP. Berend et al. [7] eloquently argues the importance of domination analysis in the study of approximation algorithms.

Let  $\mathcal{F}$  be the family of all solutions of BBQP and it is easy to see that  $|\mathcal{F}| = 2^{m+n}$ . The average objective function value  $\mathcal{A}(Q, c, d)$  of all the solutions of BBQP is given by  $\mathcal{A}(Q, c, d) = 2^{-(m+n)} \sum_{(x,y) \in \mathcal{F}} f(x, y)$ . The idea of comparing a heuristic solution to the average objective function value of all the solutions as a measure of heuristic quality for combinatorial optimization problems was originated in the Russian literature in the early 1970s. Most of these studies are focussed on the traveling salesman problem and the assignment problem (e.g. Rublineckii [28], Minina and Perekrest [22], Vizing [38], Sarvanov and Doroshko [29, 30]). In the western literature,

Gutin and Yeo [16], Grover [14], Punnen et al. [25], Punnen and Kabadi [24], Deneke and Woeginger [9] studied the traveling salesman problem and identified heuristics that guarantee a solution with objective function value no worse than the average value of all tours. Such a solution has interesting domination properties and hence the approach is also relevant in dominance analysis of heuristics. For recent developments on domination analysis, we refer to the excellent research papers [2, 7, 11, 15]. Gutin and Yeo [16], Sarvanov [31], and Angel et al. [5] studied heuristics for the quadratic assignment problem with performance guarantee in terms of average value of solutions. Similar analysis for the three-dimensional assignment problem was considered by Sarvanov [31], for the Maximum clique problem by Bendall and Margot [6], and for the satisfiability problem by Twitto [36]. Berend et al. [7] considered dominance analysis by including infeasible solutions. Other problems studied from the point of view of dominance analysis and average value based analysis include graph bipartition, variations of maximum clique and independent set problems [5, 14] and the subset-sum problem [7]. For information on dominance results and linkages with the development of heuristic algorithms based on very large scale neighborhood search, we refer to [1].

A solution with objective function value no worse than the average value of a solution with high probability can be obtained by repeated random sampling. However, it should be pointed out that even algorithms that performs well in practice could produce solutions with objective function value inferior to the average value of a solution. We observed this particularly in the case of the BBQP. Thus, a worst case performance of a heuristic that guarantees a solution with objective function value no worse than the average value of a solution is a useful measure to be included when studying worst case behavior of heuristic algorithms for combinatorial optimization problems.

Let  $(x, y), (x^0, y^0) \in \mathcal{F}$ . Then  $(x^0, y^0)$  *dominates*  $(x, y)$  if  $f(x, y) \leq f(x^0, y^0)$ . Let  $\Gamma$  be a heuristic algorithm for BBQP that produces a solution  $(x^\Gamma, y^\Gamma)$ . Define  $\mathcal{G}^\Gamma = \{(x, y) \in \mathcal{F} : f(x, y) \leq f(x^\Gamma, y^\Gamma)\}$ . Let  $I$  be the collection of all instances of BBQP. Then the *dominance number* and *dominance ratio* of  $\Gamma$  are defined respectively as

$$\inf_{\mathcal{P}(Q,c,d) \in I} |\mathcal{G}^\Gamma| \quad \text{and} \quad \inf_{\mathcal{P}(Q,c,d) \in I} \left\{ \frac{|\mathcal{G}^\Gamma|}{|\mathcal{F}|} \right\}.$$

The concept of dominance ratio in the analysis of heuristics was proposed by Glover and Punnen [11]. Prior to this work, Zemel [40] considered different measures to analyze heuristic algorithms one of which is equivalent to the dominance number. Hassin and Kuller [17] also considered similar measures in analyzing heuristic algorithms.

In this paper we obtain a closed form formula to compute  $\mathcal{A}(Q, c, d)$  in  $O(mn)$  time. We also show that any solution to BBQP with objective function value no less than  $\mathcal{A}(Q, c, d)$  dominates  $2^{m+n-2}$  solutions. Such a solution is called *no worse than average* solution. Two algorithms of complexity  $O(mn)$  are developed to compute no-worse than average solutions. Thus, the dominance ratio of these algorithms is at least  $\frac{1}{4}$ . One of these algorithms have interesting rounding property which provides data dependent lower bounds. The problem of computing a solution with objective function value no worse than the median of the objective function values of all solutions is shown to be NP-hard. Further, we show that, unless  $P=NP$ , for any fixed rational number  $\alpha > 1$ , no polynomial time approximation algorithm exists for BBQP with dominance ratio larger than  $1 - 2^{-\frac{(1-\alpha)}{\alpha}(m+n)}$ . We also analyze some very powerful local search algorithms and show that, in worst case, such algorithms could get trapped at a locally optimal solution with objective function value less than  $\mathcal{A}(Q, c, d)$ . Finally we provide a new integer programming formulation of BBQP and the

resulting LP relaxation solution could be used to initiate our rounding algorithms. Computational results are also provided using the rounding algorithms which establish that the algorithms are good candidates to obtain very fast starting solutions in complex metaheuristic algorithms.

Through out this paper, we use the following notations and naming conventions. We denote  $\mathcal{M} = \{1, 2, \dots, m\}$  and  $\mathcal{N} = \{1, 2, \dots, n\}$ . The  $i$ th component of a vector is represented simply by adding the subscript  $i$  to the name of the vector. For example, the  $i$ th component of the vector  $x^*$  is  $x_i^*$ . The set  $\{0, 1\}^n$  is denoted by  $\mathbb{B}^n$  and  $[0, 1]^n$  is denoted by  $\mathbb{U}^n$  for any positive integer  $n$ . For any positive integer  $m$ , an  $m$ -vector of all 1's is denoted by  $\mathbf{1}^m$  and an  $m$ -vector of all 0's is denoted by  $\mathbf{0}^m$ .

## 2 Average value of solutions and dominance properties

Note that there are  $2^m$  candidate solutions for  $x$  and  $2^n$  candidate solutions for  $y$ . Then the solutions in the family  $\mathcal{F}$  can be enumerated as  $\mathcal{F} = \{(x^k, y^\ell) : k = 1, 2, \dots, 2^m, \ell = 1, 2, \dots, 2^n\}$ . The next theorem gives a closed form expression to compute  $\mathcal{A}(Q, c, d)$  in  $O(mn)$  time.

**Theorem 1.**  $\mathcal{A}(Q, c, d) = \frac{1}{4} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} q_{ij} + \frac{1}{2} \sum_{i \in \mathcal{M}} c_i + \frac{1}{2} \sum_{j \in \mathcal{N}} d_j.$

*Proof.* Let  $\eta = 2^m$  and  $\nu = 2^n$ . Then

$$\begin{aligned} \mathcal{A}(Q, c, d) &= \frac{1}{2^{m+n}} \sum_{k=1}^{\eta} \sum_{\ell=1}^{\nu} f(x^k, y^\ell) \\ &= \frac{1}{2^{m+n}} \sum_{k=1}^{\eta} \sum_{\ell=1}^{\nu} \left( \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} q_{ij} x_i^k y_j^\ell + \sum_{i \in \mathcal{M}} c_i x_i^k + \sum_{j \in \mathcal{N}} d_j y_j^\ell \right) \\ &= \frac{1}{2^{m+n}} \left( \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} q_{ij} \sum_{k=1}^{\eta} x_i^k \sum_{\ell=1}^{\nu} y_j^\ell + \sum_{i \in \mathcal{M}} c_i \sum_{\ell=1}^{\nu} \sum_{k=1}^{\eta} x_i^k + \sum_{j \in \mathcal{N}} d_j \sum_{k=1}^{\eta} \sum_{\ell=1}^{\nu} y_j^\ell \right) \\ &= \frac{1}{2^{m+n}} \left( 2^{m-1} 2^{n-1} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} q_{ij} + \nu 2^{m-1} \sum_{i \in \mathcal{M}} c_i + \eta 2^{n-1} \sum_{j \in \mathcal{N}} d_j \right) \\ &= \frac{1}{4} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} q_{ij} + \frac{1}{2} \sum_{i \in \mathcal{M}} c_i + \frac{1}{2} \sum_{j \in \mathcal{N}} d_j. \end{aligned}$$

□

If either  $x = 0$  or  $y = 0$ , then  $x^T Q y = 0$  and such an  $(x, y)$  is called a *trivial solution*. Note that  $f(x, y)$  need not be equal to zero for trivial solutions. All remaining solutions are called *nontrivial solutions*. Maximizing  $f(x, y)$  over trivial solutions is straightforward and thus one can restrict attention to non-trivial solutions only. The number of nontrivial solutions is  $(2^m - 1)(2^n - 1)$ . Let  $\bar{A}(Q, c, d)$  denote the average value of all nontrivial solutions for  $\mathcal{P}(Q, c, d)$ .

**Corollary 2.**  $\bar{A}(Q, c, d) = \frac{2^{m-1}2^{n-1}}{(2^m-1)(2^n-1)} \left( \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} q_{ij} + \frac{2^n-1}{2^{n-1}} \sum_{i \in \mathcal{M}} c_i + \frac{2^m-1}{2^{m-1}} \sum_{j \in \mathcal{N}} d_j \right)$ . Fur-

ther,  $\bar{A}(Q, \mathbf{0}^m, \mathbf{0}^n) = \frac{2^{m+n}}{(2^m-1)(2^n-1)} \mathcal{A}(Q, \mathbf{0}^m, \mathbf{0}^n)$  and  $\lim_{n \rightarrow \infty} \lim_{m \rightarrow \infty} \bar{A}(Q, c, d) = \mathcal{A}(Q, c, d)$ .

From Corollary 2, the asymptotic behavior of  $\bar{A}(Q, c, d)$  and  $\mathcal{A}(Q, c, d)$  are the same. Thus, hereafter we focus our attention on  $\mathcal{A}(Q, c, d)$  only.

Let  $\mathcal{G} = \{(x, y) : x \in \{0, 1\}^m, y \in \{0, 1\}^n, f(x, y) \leq \mathcal{A}(Q, c, d)\}$ . Thus,  $\mathcal{G}$  consists of all solutions of BBQP that are no worse than average.

**Theorem 3.**  $|\mathcal{G}| \geq 2^{m+n-2}$ .

*Proof.* For any solution  $(x, y) \in \mathcal{F}$ , let  $P(x, y) = \{(x, y), (x, \mathbf{1}^n - y), (\mathbf{1}^m - x, y), (\mathbf{1}^m - x, \mathbf{1}^n - y)\}$ . It can be verified that  $P(x, y) = P(x, \mathbf{1}^n - y) = P(\mathbf{1}^m - x, y) = P(\mathbf{1}^m - x, \mathbf{1}^n - y)$  and  $P(x, y) \neq P(x', y')$  if  $x' \notin \{x, \mathbf{1}^m - x\}$  or  $y' \notin \{y, \mathbf{1}^n - y\}$ . Thus, we can partition the solution space  $\mathcal{F}$  into  $\frac{1}{4}2^{m+n} = 2^{m+n-2}$  disjoint sets  $P(x^k, y^k)$ ,  $k = 1, 2, \dots, 2^{m+n-2} = \omega$ , say. Note that

$$\begin{aligned} f(x, y) + f(\mathbf{1}^m - x, y) + f(x, \mathbf{1}^n - y) + f(\mathbf{1}^m - x, \mathbf{1}^n - y) &= \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} q_{ij} + 2 \sum_{i \in \mathcal{M}} c_i + 2 \sum_{j \in \mathcal{N}} d_j \\ &= 4\mathcal{A}(Q, c, d). \end{aligned} \quad (2)$$

From equation (2), it follows immediately that

$$\min \{f(x, y), f(\mathbf{1}^m - x, y), f(x, \mathbf{1}^n - y), f(\mathbf{1}^m - x, \mathbf{1}^n - y)\} \leq \mathcal{A}(Q, c, d).$$

Thus, from each  $P(x^k, y^k)$ ,  $k = 1, 2, \dots, \omega$ , choose a solution with smallest objective function value to form the set  $D_1$ . By construction,  $f(x, y) \leq \mathcal{A}(Q, c, d)$  for all  $(x, y) \in D_1$ . Since  $|D_1| = 2^{m+n-2}$ , the result follows.  $\square$

The lower bound on  $\mathcal{G}$  established in Theorem 3 is tight. To see this, consider the matrix  $Q$  defined by

$$q_{ij} = \begin{cases} -1 & \text{if } i = m, j = n, \\ 0 & \text{otherwise.} \end{cases}$$

and choose  $c$  and  $d$  as zero vectors in  $R^m$  and  $R^n$ , respectively. Then  $\mathcal{A}(Q, c, d) = \frac{-1}{4}$  and the set of solutions  $(x, y)$  with  $f(x, y) \leq \mathcal{A}(Q, c, d)$  is precisely  $\mathcal{G} = \{(x, y) | x_m = y_n = 1\}$ . Clearly,  $|\mathcal{G}| = 2^{m+n-2}$  and hence the bound obtained in Theorem 3 is the best possible.

Theorem 3 establishes that any algorithm that guarantees a solution with objective function value no worse than  $\mathcal{A}(Q, c, d)$  dominates  $2^{m+n-2}$  solutions of the BBQP  $\mathcal{P}(Q, c, d)$ . In other words, the domination ratio of such an algorithm is at least 1/4. Our next theorem establishes an upper bound on the dominance number of any polynomial time approximation algorithm for BBQP.

**Theorem 4.** *Unless  $P=NP$ , no polynomial time algorithm for BBQP can have dominance number more than  $2^{m+n} - 2^{\lfloor \frac{m+n}{\alpha} \rfloor}$  for any fixed rational number  $\alpha > 1$ .*

*Proof.* We show that a polynomial time algorithm  $\Omega$  for BBQP with dominance number at least  $2^{m+n} - 2^{\lfloor \frac{m+n}{\alpha} \rfloor} + 1$  can be used to compute an optimal solution to BBQP. Without loss of generality, assume  $\alpha = \frac{a}{b}$  where  $a$  and  $b$  are positive relatively prime integers with  $a > b$ . Consider an instance  $\mathcal{P}(Q, c, d)$  of BBQP. Let  $Q^* = (q_{ij}^*)$  be an  $abm \times abn$  matrix where

$$q_{ij}^* = \begin{cases} q_{ij} & \text{if } i \in \mathcal{M} \text{ and } j \in \mathcal{N}, \\ 0 & \text{otherwise.} \end{cases}$$

Likewise, let  $c^*$  and  $d^*$  be vectors in  $R^{abm}$  and  $R^{abn}$  such that

$$c_i^* = \begin{cases} c_i & \text{if } i \in \mathcal{M}, \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad d_j^* = \begin{cases} d_j & \text{if } j \in \mathcal{N}, \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to verify that from any optimal solution to the BBQP instance  $\mathcal{P}(Q^*, c^*, d^*)$  an optimal solution to  $\mathcal{P}(Q, c, d)$  can be recovered. The total number of solutions of  $\mathcal{P}(Q^*, c^*, d^*)$  is  $2^{ab(m+n)}$  of which at least  $2^{ab(m+n)-(m+n)}$  are optimal. So the maximum number of non-optimal solutions is  $2^{ab(m+n)} - 2^{ab(m+n)-(m+n)}$ . Solve the BBQP instance  $\mathcal{P}(Q^*, c^*, d^*)$  using  $\Omega$  and let  $(x^*, y^*)$  be the resulting solution. By hypothesis, the objective function value of  $(x^*, y^*)$  is not worse than that of at least  $2^{ab(m+n)} - 2^{\frac{ab(m+n)}{a/b}} + 1 = 2^{ab(m+n)} - 2^{b^2(m+n)} + 1$  solutions. Since  $a > b$ , we have  $2^{ab(m+n)} - 2^{b^2(m+n)} + 1 > 2^{ab(m+n)} - 2^{ab(m+n)-(m+n)}$ . Thus,  $(x^*, y^*)$  must be optimal for  $\mathcal{P}(Q^*, c^*, d^*)$ . From  $(x^*, y^*)$ , an optimal solution to  $\mathcal{P}(Q, c, d)$  can be recovered by simply taking the first  $m$  components of  $x^*$  and first  $n$  components of  $y^*$ . The result now follows from NP-completeness of BBQP.  $\square$

Theorem 4 implies that unless  $P=NP$ , no polynomial time approximation algorithm for BBQP can have dominance ratio more than  $1 - 2^{\frac{(1-\alpha)}{\alpha}(m+n)}$  for any fixed rational number  $\alpha > 1$ .

Although we have a closed form formula for computing the average value of all solutions to BBQP, we now show that computing the median value of all solutions is NP-hard.

Since  $|\mathcal{F}|$  is even, there are two values of median, say  $\theta_1$  and  $\theta_2$ , where  $\theta_1 \leq \theta_2$ . A median finding algorithm could simply produce  $\theta_1$  or  $\theta_2$ , but we may not know precisely, the output is either  $\theta_1$  or  $\theta_2$ .

**Theorem 5.** *Computing a median of the objective function values of BBQP is NP-hard.*

*Proof.* Suppose we have a polynomial time algorithm to compute a median of the objective function values of BBQP. We will show that this algorithm can be used to solve the PARTITION problem, which is defined as follows: Given  $n$  positive integers  $a_1, a_2, \dots, a_n$ , determine if there exists a partition  $S_1$  and  $S_2$  of  $N = \{1, 2, \dots, n\}$  such that  $\sum_{j \in S_1} a_j = \sum_{j \in S_2} a_j$ . From an instance of PARTITION, construct an instance of BBQP as follows: Choose  $c$  as the zero vector,  $d_j = a_j$  for  $j = 1, 2, \dots, n$ . Choose  $\mathcal{M} = \{1, 2\}$ . Define  $q_{1j} = a_j \epsilon$  and  $q_{2j} = -a_j \epsilon$ , where  $\epsilon$  is a very small positive number. For each subset  $H$  of  $N$ , let  $y^H$  be its characteristic vector, i.e.  $y^H \in \mathbb{B}^n$  and  $y_j^H = 1$  if and only if  $j \in H$ . For each choice of  $H$ , we can associate four choices for  $x$  as  $x = (0, 0)$ ,  $x = (1, 0)$ ,  $x = (0, 1)$  or  $x = (1, 1)$ . Thus, for each  $H$ , we get the following solutions  $F_H = \{(0, 0, y^H), (1, 0, y^H), (0, 1, y^H), (1, 1, y^H)\}$ . Now,  $f((0, 0), y^H) = \sum_{j \in H} a_j$ ,  $f((1, 0), y^H) = (1 + \epsilon) \sum_{j \in H} a_j$ ,  $f((0, 1), y^H) = (1 - \epsilon) \sum_{j \in H} a_j$  and  $f((1, 1), y^H) = \sum_{j \in H} a_j$ . Thus,  $f((0, 1), y^H) < f((0, 0), y^H) = f((1, 1), y^H) < f((1, 0), y^H)$ . There are  $2^n$  choices for  $H$  and hence there are  $2^{n+2}$

different solutions for the BBQP constructed. For each subset  $H$  of  $N$ , let  $g(H) = \sum_{j \in H} a_j$  and  $G = \{g(H) : H \subseteq N\}$ . We first observe that  $G$  has two medians and these median values are the same and equal to  $\frac{1}{2} \sum_{j \in N} a_j$  if and only if  $N$  has the required partition. This follows from the fact that for any  $H \subseteq N$ , either  $g(H) \leq \frac{1}{2} \sum_{j \in N} a_j \leq g(N \setminus H)$  or  $g(N \setminus H) \leq \frac{1}{2} \sum_{j \in N} a_j \leq g(H)$ .

Let  $\sigma_1 < \sigma_2 < \dots < \sigma_\kappa$  be an ascending arrangement of distinct  $g(H)$ ,  $H \subseteq N$  and let  $W_k = \{H \subseteq N : g(H) = \sigma_k\}$ . Note that  $|W_k| = |W_{\kappa+1-k}|$  and  $\sigma_k + \sigma_{\kappa+1-k} = \sum_{j \in N} a_j$ . Thus, the required partition exists if and only if median of  $\{\sigma_1, \sigma_2, \dots, \sigma_\kappa\} = \frac{1}{2} \sum_{j \in N} a_j$ . Consider an ascending arrangement of  $f(x, y)$  for all solutions  $(x, y)$  of the BBQP constructed. This can be grouped as blocks of values  $B_1 < B_2 < \dots < B_\kappa$  where the block  $B_k$  has the structure

$$\overbrace{\sigma_k(1-\epsilon) = \sigma_k(1-\epsilon) = \dots = \sigma_k(1-\epsilon)}^{\text{repeated } |W_k| \text{ times}} < \underbrace{\sigma_k = \sigma_k = \dots = \sigma_k}_{\text{repeated } 2|W_k| \text{ times}} < \overbrace{\sigma_k(1+\epsilon) = \sigma_k(1+\epsilon) = \dots = \sigma_k(1+\epsilon)}^{\text{repeated } |W_k| \text{ times}}$$

for  $k = 1, 2, \dots, \kappa$ . Thus, median of  $\{\sigma_1, \sigma_2, \dots, \sigma_\kappa\}$  is the same as median of the objective function values of BBQP. Thus, the required partition exists if and only both the median values are the same and equal to  $\frac{1}{2} \sum_{j \in N} a_j$ . The proof now follows from the NP-completeness of PARTITION.  $\square$

It may be noted that the above theorem does not rule out the possibility of a polynomial time algorithm with dominance number  $2^{m+n-1}$ .

### 3 Average value of solutions and local search

In this section we consider two natural local search heuristics for BBQP and show that the solution produced could have objective function value worse than  $\mathcal{A}(Q, c, d)$ . One of the popular heuristics for BBQP is the *alternating algorithm* proposed by many authors [21, 10, 19]. The algorithm starts with a candidate solution  $x^0$  and try to choose an optimal  $y^0$ . Then fix  $y^0$  and tries to find the best candidate for  $x$ , say  $x^1$  yielding a solution  $(x^1, y^0)$ . These operations can be carried out using the formulas

$$y_j^0 = \begin{cases} 1 & \text{if } \sum_{i \in \mathcal{M}} q_{ij} x_i^0 + d_j > 0, \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad x_i^1 = \begin{cases} 1 & \text{if } \sum_{j \in \mathcal{N}} q_{ij} y_j^0 + c_i > 0, \\ 0 & \text{otherwise,} \end{cases}$$

Now fix  $x = x^1$  and try to choose the best  $y = y^1$  and the process is continued until no improvement is possible by fixing either  $x$  variables or  $y$  variables. The algorithm terminates when a locally optimal solution is reached. From experimental analysis [19], it is known that this algorithm produces reasonably good solutions on average. To the best of our knowledge, worst-case behavior of this algorithm has not been investigated.

**Theorem 6.** *The objective function value of a locally optimal solution produced by the alternating algorithm could be arbitrarily bad and could be worse than  $\mathcal{A}(Q, c, d)$ .*

*Proof.* Choose  $m = n$ ,  $c = d = 0$ , set  $q_{11} = 1$ ,  $q_{nn} = M$  and  $q_{ij} = 0$  for all other combinations of  $i$  and  $j$ . Choose the starting solution  $x_1^0 = 1$  and  $x_i^0 = 0$  for  $i \neq 1$ . The algorithm will choose  $y_1^0 = 1$  and  $y_j^0 = 0$  for  $j \neq 1$ . Now  $(x^0, y^0)$  will be locally optimal solution with objective function value 1,

but the optimal objective function value is  $M + 1$  for any  $M > 0$ . The average cost of a solution is  $\frac{M+1}{4}$ . Thus, for  $M > 3$  the solution produced by the alternating algorithm is less than  $\mathcal{A}(Q, c, d)$ . In fact, by choosing  $M$  large, the solution can be made arbitrarily bad.  $\square$

Despite this example, it is easy to see that a solution produced by the alternating algorithm dominates at least  $2^m + 2^n - 1$  solutions.

Let us now consider a more general neighborhood, which is a variation of the  $k$ -exchange neighborhood studied for various combinatorial optimization problems. For any  $(x^0, y^0) \in \mathcal{F}$ , let  $\mathbb{N}^{h,k}$  be the set of solutions in  $\mathcal{F}$  obtained by flipping at most  $h$  components of  $x^0$  and at most  $k$  components of  $y^0$ . If  $h = m$  or  $k = n$ , we ignore “,” in the definition of  $\mathbb{N}^{h,k}$ . Note that  $|\mathbb{N}^{h,k}| = \left(\sum_{j=0}^h \binom{m}{j}\right) \left(\sum_{i=0}^k \binom{n}{i}\right)$ , and the best solution in this neighborhood can be identified in polynomial time for fixed  $h$  and  $k$ . A more powerful neighborhood is  $\mathbb{N}^\alpha = \mathbb{N}^{m\alpha} \cup \mathbb{N}^{\alpha n}$  and  $|\mathbb{N}^\alpha| = 2^m \sum_{j=0}^\alpha \binom{n}{j} + 2^n \sum_{i=0}^\alpha \binom{m}{i} - \sum_{i=0}^\alpha \binom{m}{i} \sum_{j=0}^\alpha \binom{n}{j}$ . Again, this neighborhood can also be searched for an improving solution in polynomial time for fixed  $\alpha$  [19]. It may be noted that a solution produced by the alternating algorithm is locally optimal with respect to the neighborhood  $\mathbb{N}^0 = \mathbb{N}^{m0} \cup \mathbb{N}^{0n}$ . Glover et al. [13] considered the neighborhoods  $\mathbb{N}^1$ ,  $\mathbb{N}^2$ , and  $\mathbb{N}^{1,1}$ . They provided fast and efficient algorithms for exploring these neighborhoods supported by detailed computational analysis. They also considered tabu search algorithms using these neighborhoods in a hybrid form. Computational results with these algorithms provided very high quality solutions, improving several benchmark instances. Nonetheless, our next theorem shows that even such very powerful local search algorithms could provide solutions with objective function values that are inferior to  $\mathcal{A}(Q, c, d)$  even if we allow  $\alpha$  to be a function of  $n$ .

**Theorem 7.** *A locally optimal solution to BBQP with respect to the neighborhood  $\mathbb{N}^\alpha = \mathbb{N}^{m\alpha} \cup \mathbb{N}^{\alpha n}$  could be worse than average for any  $\alpha \leq \lfloor \frac{n}{5} \rfloor$ .*

*Proof.* Consider the matrix  $Q$  defined as

$$q_{ij} = \begin{cases} \lambda & \text{if } i = m, j = n, \\ -1 & \text{if } i = m \text{ or } j = n \text{ but } (i, j) \neq (m, n), \\ a & \text{otherwise.} \end{cases} \quad (3)$$

and choose  $c$  and  $d$  as zero vectors. Without loss of generality, we assume  $m = n$ . Otherwise, we can extend the matrix  $Q$  into an  $n \times n$  matrix by adding  $n - m$  rows of zeros and extending the vector  $c$  into an  $n$ -vector by making the last  $n - m$  entries zeros. We also assume that  $n$  is a multiple of 5. Consider the solution  $(x^0, y^0)$  where  $x_n^0 = y_n^0 = 1$  and all other components are zero. Also, let  $\alpha = \frac{n}{5}$  and assume  $n \geq 6$ .

Let  $N_x^r(0)$  be the set of all  $x \in \mathbb{B}^n$  with  $x_n = 0$  obtained by flipping exactly  $r$  entries of  $x^0$  and  $N_x^r(1)$  be the set of all  $x \in \mathbb{B}^n$  with  $x_n = 1$  obtained by flipping exactly  $r$  entries of  $x^0$ . Define  $N_y^r(0)$  and  $N_y^r(1)$  analogously. Note that for any  $(x, y) \in \mathbb{N}^{\alpha n}$ ,  $x \in N_x^r(0) \cup N_x^r(1)$  and  $y \in N_y^s(0) \cup N_y^s(1)$  for some  $0 \leq r \leq \alpha, 0 \leq s \leq n$  and for any  $(x, y) \in \mathbb{N}^{\alpha n}$ ,  $x \in N_x^r(0) \cup N_x^r(1)$  and  $y \in N_y^s(0) \cup N_y^s(1)$  for some  $0 \leq r \leq n, 0 \leq s \leq \alpha$ . Thus, for  $(x, y) \in \mathbb{N}^\alpha$  we have

$$f(x, y) = \begin{cases} (r-1)(s-1)a & \text{if } x \in N_x^r(0), y \in N_y^s(0) \text{ for } (r, s) \in I_n^\alpha \cup I_\alpha^n \\ (ra-1)(s-1) & \text{if } x \in N_x^r(1), y \in N_y^s(0) \text{ for } (r, s) \in I_n^\alpha \cup I_\alpha^n \\ (r-1)(sa-1) & \text{if } x \in N_x^r(0), y \in N_y^s(1) \text{ for } (r, s) \in I_n^\alpha \cup I_\alpha^n \\ rsa-r-s+\lambda & \text{if } x \in N_x^r(1), y \in N_y^s(1) \text{ for } (r, s) \in I_n^\alpha \cup I_\alpha^n. \end{cases} \quad (4)$$

where  $I_q^p = \{0, 1, \dots, p\} \times \{0, 1, \dots, q\}$ .

Thus,  $(x^0, y^0)$  is locally optimal with respect to  $N^\alpha$  if and only if

$$(r-1)(s-1)a \leq \lambda \text{ for all } (r, s) \in I_n^\alpha \cup I_\alpha^n \quad (5)$$

$$(ra-1)(s-1) \leq \lambda \text{ for all } (r, s) \in I_n^\alpha \cup I_\alpha^n, \quad (6)$$

$$(r-1)(sa-1) \leq \lambda \text{ for all } (r, s) \in I_n^\alpha \cup I_\alpha^n, \quad (7)$$

$$rsa-r-s \leq 0 \text{ for all } (r, s) \in I_n^\alpha \cup I_\alpha^n. \quad (8)$$

It is not very difficult to verify that conditions (5), (6), (7), and (8), are satisfied if

$$(\alpha-1)(n-1)a \leq \lambda, \quad (9)$$

$$(n-1)(\alpha a-1) \leq \lambda, \quad (10)$$

$$(\alpha-1)(na-1) \leq \lambda, \text{ and} \quad (11)$$

$$a\alpha n - \alpha - n \leq 0. \quad (12)$$

Choose  $a = \frac{6}{n}$ . Then inequality (12) holds and inequality (9) implies inequalities (10) and (11). Choose  $\lambda = (\alpha-1)(n-1)\frac{6}{n}$ . Then  $(x^0, y^0)$  is locally optimal. Now,

$$\begin{aligned} \mathcal{A}(Q, c, d) - f(x^0, y^0) &= \frac{1}{4}((n-1)^2a - (2n-2) + \lambda) - \lambda \\ &= \frac{1}{4n}(\frac{2}{5}n^2 + \frac{58}{5}n - 12) > 0 \text{ for } n \geq 6. \end{aligned}$$

This completes the proof. □

As an immediate corollary, we have the following result.

**Corollary 8.** *For any fixed  $h$  and  $k$ , the objective function value of a locally optimal solution with respect to the neighborhood  $\mathbb{N}^{hk}$  could be worse than  $\mathcal{A}(Q, c, d)$  for sufficiently large  $m$  and  $n$ .*

*Proof.* Choose  $\alpha = \max\{h, k\}$ . Then a locally optimal solution with respect to  $\mathbb{N}^\alpha$  is not worse than a locally optimal with respect to  $\mathbb{N}^{hk}$ . The result now follows from Theorem 7. □

These examples motivates us to develop polynomial time algorithms for BBQP that guarantee a solution with objective function value no worse than  $\mathcal{A}(Q, c, d)$ .

## 4 Algorithms with no worse than average guarantee

We first consider a very simple algorithm to compute a solution with objective function value guaranteed to be no worse than  $\mathcal{A}(Q, c, d)$ . The algorithm simply takes fractional vectors  $x \in \mathbb{U}^n$  and  $y \in \mathbb{U}^n$  and applies a rounding scheme to produce a solution for BBQP. Let  $x \in \mathbb{U}^m$  and  $y \in \mathbb{U}^n$ . Extending the definition of  $f(x, y)$  for 0-1 vectors, define

$$f(x, y) = \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} q_{ij} x_i y_j + \sum_{i \in \mathcal{M}} c_i x_i + \sum_{j \in \mathcal{N}} d_j y_j.$$

Consider the solutions  $y^* \in \mathbb{B}^n$  and  $x^* \in \mathbb{B}^m$  given by

$$y_j^* = \begin{cases} 1 & \text{if } d_j + \sum_{i \in \mathcal{M}} q_{ij} x_i > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (13)$$

$$x_i^* = \begin{cases} 1 & \text{if } c_i + \sum_{j \in \mathcal{N}} q_{ij} y_j^* > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

Note that  $x^*$  is the optimal 0-1 vector when  $y$  is fixed at  $y^*$ , and equation (13) rounds the  $y$  to  $y^*$  using a prescribed rounding criterion. The process of constructing  $(x^*, y^*)$  from  $(x, y)$  thus called a *round- $y$  optimize- $x$  algorithm* or *RyOx-algorithm*. The next theorem establishes a lower bound on the objective function value of the solution produced by the RyOx-algorithm.

**Theorem 9.**  $f(x^*, y^*) \geq f(x, y)$ .

*Proof.*

$$\begin{aligned} f(x, y) &= \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} q_{ij} x_i y_j + \sum_{i \in \mathcal{M}} c_i x_i + \sum_{j \in \mathcal{N}} d_j y_j \\ &= \left[ \sum_{j \in \mathcal{N}} \left( \sum_{i \in \mathcal{M}} q_{ij} x_i + d_j \right) \right] y_j + \sum_{i \in \mathcal{M}} c_i x_i \\ &\leq \sum_{j \in \mathcal{N}} \left( \sum_{i \in \mathcal{M}} q_{ij} x_i + d_j \right) y_j^* + \sum_{i \in \mathcal{M}} c_i x_i && \text{(by construction of } y^*) \\ &= \sum_{j \in \mathcal{N}} d_j y_j^* + \sum_{i \in \mathcal{M}} \left( \sum_{j \in \mathcal{N}} q_{ij} y_j^* + c_i \right) x_i \\ &\leq \sum_{j \in \mathcal{N}} d_j y_j^* + \sum_{i \in \mathcal{M}} \left( \sum_{j \in \mathcal{N}} q_{ij} y_j^* + c_i \right) x_i^* && \text{(by construction of } x^*) \\ &= f(x^*, y^*). \quad \square \end{aligned}$$

Note that  $(x^*, y^*)$  can be constructed in  $O(mn)$  time whenever  $x$  and  $y$  are rational numbers. We can also round  $x$  first to obtain  $x^0 \in \mathbb{B}^m$  and choose optimal  $y = y^0$  by fixing  $x$  at  $x^0$ . This is done using the rounding scheme given by the following equations:

$$x_i^0 = \begin{cases} 1 & \text{if } c_i + \sum_{j \in \mathcal{N}} q_{ij} y_j > 0, \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad y_j^0 = \begin{cases} 1 & \text{if } d_j + \sum_{i \in \mathcal{M}} q_{ij} x_i^0 > 0, \\ 0 & \text{otherwise,} \end{cases}$$

**Theorem 10.**  $f(x^0, y^0) \geq f(x, y)$ .

The proof of Theorem 10 follows along the same line as Theorem 9 and hence omitted. The process of constructing  $(x^0, y^0)$  is called *round- $x$  optimize- $y$  algorithm* or RxOy-algorithm. The complexity of the RxOy-algorithm is also  $O(mn)$ .

**Corollary 11.** *A solution  $(\bar{x}, \bar{y})$  for BBQP satisfying  $f(\bar{x}, \bar{y}) \geq \mathcal{A}(Q, c, d)$  can be obtained in  $O(mn)$  time.*

*Proof.* Let  $x_i = 1/2$  for all  $i \in \mathcal{M}$  and  $y_j = 1/2$  for all  $j \in \mathcal{N}$ . Then it can be verified that  $f(x, y) = \mathcal{A}(Q, c, d)$ , but  $(x, y)$  is not feasible for BBQP. Now, choose  $(\bar{x}, \bar{y})$  as the output of either the RyOx-algorithm or the RxOy-algorithm. The result now follows from Theorems 9 or 10.  $\square$

In view of Corollary 11 and Theorem 9, the dominance ratio of the RyOx-algorithm and the RxOy-algorithm is at least  $\frac{1}{4}$ . By choosing an appropriate starting solution, we can establish improved dominance ratio for these algorithms.

We now discuss an unexpected upper bound on  $\mathcal{A}(Q, c, d)$ . As a consequence, we have yet another simple scheme to compute a solution that is not worse than average. Let  $\alpha = \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} q_{ij}$ ,  $\beta = \sum_{i \in \mathcal{M}} c_i$  and  $\gamma = \sum_{j \in \mathcal{N}} d_j$ .

**Theorem 12.**  $\mathcal{A}(Q, c, d) \leq \max\{\alpha + \beta + \gamma, \beta, \gamma, 0\}$ .

*Proof.* Let  $u$  and  $v$  be real numbers in  $[0, 1]$ . Choose  $x \in \mathbb{B}^m$ ,  $y \in \mathbb{B}^n$  be such that  $x_i = u$  for all  $i \in \mathcal{M}$  and  $y_j = v$  for all  $j \in \mathcal{N}$ . Then  $f(x, y) = \alpha uv + \beta u + \gamma v = \eta(u, v)$ , say. Note that  $\eta(1/2, 1/2) = \mathcal{A}(Q, c, d)$ . Thus,  $\max\{\eta(u, v) : (u, v) \in \mathbb{U}^2\} \geq \mathcal{A}(Q, c, d)$ . Since  $\eta(u, v)$  is bilinear, its maximum is attained at an extreme point of the square  $\mathbb{U}^2$ . Since these extreme points are precisely  $(0, 0), (1, 0), (0, 1), (1, 1)$ , the result follows.  $\square$

**Corollary 13.** *One of the solutions  $(\mathbf{1}^m, \mathbf{1}^n), (\mathbf{1}^m, \mathbf{0}^n), (\mathbf{0}^m, \mathbf{1}^n), (\mathbf{0}^m, \mathbf{0}^n)$  of BBQP have an objective function value no worse than  $\mathcal{A}(Q, c, d)$ .*

The proof of this corollary follows directly from Theorem 12. We can compute  $\alpha$ ,  $\beta$  and  $\gamma$  in  $O(mn)$  time and hence we have a solution no worse than average in  $O(mn)$  time. Interestingly, if  $\alpha, \beta$  and  $\gamma$  are given, then we can identify a solution to BBQP with objective function value no worse than  $\mathcal{A}(Q, c, d)$  in  $O(1)$  time. The solution produced by Corollary 13 is trivial and may not be of much practical value. Nevertheless, the simple upper bound on  $\mathcal{A}(Q, c, d)$  established by Theorem 12 is very interesting and have interesting consequences as discussed below.

Recall that the alternating algorithm starts with a solution  $(x^0, y^0)$ , fix  $x^0$  and find the best  $y$ , say  $y^1$ . Then fix  $y$  at  $y^1$  and compute the optimal  $x$  and so on. Since we initiate the algorithm by fixing  $x$  first, we call this the  *$x$ -first alternating algorithm*. We can also start the algorithm by fixing  $y^0$  first and the resulting variation is called the  *$y$ -first alternating algorithm*.

**Theorem 14.** *The best solution amongst the solutions produced by the  $x$ -first alternating algorithm with starting solutions  $(\mathbf{1}^m, \mathbf{1}^n)$  and  $(\mathbf{0}^m, \mathbf{0}^n)$  dominates  $2^{m+n-2} + 3(2^{n-1})$  solutions.*

*Proof.* Let  $(x^*, y^*)$  be the best solution obtained. When the starting solution is  $(\mathbf{1}^m, \mathbf{1}^n)$ ,  $f(x^*, y^*) \geq \max\{f(\mathbf{1}^m, \mathbf{1}^n), f(\mathbf{1}^m, \mathbf{0}^n)\}$ . Likewise, when the starting solution is  $(\mathbf{0}^m, \mathbf{0}^n)$ , we have  $f(x^*, y^*) \geq \max\{f(\mathbf{0}^m, \mathbf{0}^n), f(\mathbf{0}^m, \mathbf{1}^n)\}$ . Thus, by Theorem 12  $f(x^*, y^*) \geq \mathcal{A}(Q, c, d)$  and hence by Theorem 3,  $(x^*, y^*)$  dominates at least  $2^{m+n-2}$  solutions. To account for the remaining solutions that are dominated by  $(x^*, y^*)$  we proceed as follows.

Construct the set  $D_1$  as in the proof of Theorem 3. Recall that  $|D_1| = 2^{m+n-2}$ . Let  $D_2 = \{(\mathbf{1}^m, y), (\mathbf{0}^m, y) : y \in \mathbb{B}^n\}$ . Now,  $|D_2| = 2^{n+1}$ . For any  $y \in \mathbb{B}^n$ , we have  $P(\mathbf{0}^m, y) = P(\mathbf{1}^m, y)$  and  $P(\mathbf{0}^m, y) = \{(\mathbf{0}^m, y), (\mathbf{1}^m, y), (\mathbf{0}^m, \mathbf{1}^n - y), (\mathbf{1}^m, \mathbf{1}^n - y)\}$ . Thus,  $D_2$  can be partitioned into sets of the form  $P(\mathbf{0}^m, y^k)$  and there are  $\frac{1}{4}2^{n+1} = 2^{n-1}$  such sets. Exactly one element from each  $P(\mathbf{0}^m, y^k)$  is in  $D_1$ . Thus,  $|D_1 \cap D_2| = 2^{n-1}$  and hence  $|D_1 \cup D_2| = 2^{m+n-2} + 2^{n+1} - 2^{n-1} = 2^{n-1} (2^{m-1} + 3)$ .  $\square$

From Theorem 14 the dominance ratio of the  $x$ -first alternating algorithm is at least  $\frac{1}{4} + \frac{3}{2^{m+1}}$  when starting solutions are selected carefully and the algorithm is applied twice. Note that to achieve this dominance ratio, we simply need to perform only one iteration each, when the algorithm starts with  $(\mathbf{1}^m, \mathbf{1}^n)$  and  $(\mathbf{0}^m, \mathbf{0}^n)$ . Thus, we can achieve this dominance ratio in polynomial time. A similar result can be derived for  $y$ -first alternating algorithm and local search algorithm with neighborhood  $\mathbb{N}^\alpha$  for any  $\alpha \geq 0$ . Further, similar dominance ratio can be achieved by RxOy-algorithm or RyOx-algorithm applied twice, once starting with  $(\mathbf{1}^m, \mathbf{1}^n)$  and then starting with  $(\mathbf{0}^m, \mathbf{0}^n)$  and choosing the best solution.

The problem BBQP can be formulated as integer linear programming problem [26] as follows:

$$\text{ILP1: Maximize } \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} q_{ij} z_{ij} + \sum_{i \in \mathcal{M}} c_i x_i + \sum_{j \in \mathcal{N}} d_j y_j$$

$$\text{Subject to } z_{ij} - x_i \leq 0, i \in \mathcal{M}, j \in \mathcal{N}$$

$$z_{ij} - y_j \leq 0, i \in \mathcal{M}, j \in \mathcal{N}$$

$$z_{ij} - x_i - y_j \geq -1, (i, j) \in S,$$

$$x_i \in \{0, 1\} \text{ for } i \in \mathcal{M}, y_j \in \{0, 1\} \text{ for } j \in \mathcal{N}, z_{ij} \in \{0, 1\} \text{ for } (i, j) \in \mathcal{M} \times \mathcal{N},$$

where  $S = \{(i, j) : q_{ij} < 0\}$ . Let  $(z', x', y')$  be an optimal solution to the LP relaxation, where  $z'$  is an  $m \times n$  matrix with  $(i, j)$ th entry  $z'_{ij}$ , and  $x'$  is an  $m$ -vector with  $i$ th entry  $x'_i$  and  $y'$  is an  $n$ -vector with  $j$ th entry  $y'_j$ . Recall that for  $x' \in [0, 1]^m$  and  $y' \in [0, 1]^n$ ,  $\phi(x', y') = \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} q_{ij} x'_i y'_j + \sum_{i \in \mathcal{M}} c_i x'_i + \sum_{j \in \mathcal{N}} d_j y'_j$  and  $h(z', x', y')$  is the optimal objective function value of the linear programming relaxation of ILP1.

For any  $x \in [0, 1]^m$  and  $y \in [0, 1]^n$ , the solution  $(x^1, y^1)$  obtained by the RxOy (RyOx) algorithm satisfies  $\phi(x, y) \leq f(x^1, y^1) \leq h(z', x', y')$ . This follows from Theorem 9 and the property of LP relaxations. Alternative integer programming formulations of BBQP could provide different LP relaxation solutions and hence the resulting RxOy (RyOx) rounding solutions could be different. We give below a new integer programming formulation of BBQP by increasing the number of

variables.

$$\begin{aligned}
\text{ILP2: Maximize } & \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} q_{ij} \left( \frac{1}{4}u_{ij} + v_{ij} - \frac{1}{4}w_{ij} - \frac{1}{4}z_{ij} \right) + \sum_{i \in \mathcal{M}} c_i x_i + \sum_{j \in \mathcal{N}} d_j y_j \\
\text{Subject to } & u_{ij} + 2v_{ij} = x_i + y_j \text{ for } i \in \mathcal{M}, j \in \mathcal{N} \\
& u_{ij} + v_{ij} \leq 1, \text{ for } i \in \mathcal{M}, j \in \mathcal{N} \\
& -z_{ij} + w_{ij} = x_i - y_j \text{ for } i \in \mathcal{M}, j \in \mathcal{N} \\
& z_{ij} + w_{ij} \leq 1 \text{ for } i \in \mathcal{M}, j \in \mathcal{N} \\
& x_i \in \{0, 1\} \text{ for } i \in \mathcal{M}, y_j \in \{0, 1\} \text{ for } j \in \mathcal{N} \\
& u_{ij}, v_{ij}, w_{ij}, z_{ij} \in \{0, 1\} \text{ for } (i, j) \in \mathcal{M} \times \mathcal{N},
\end{aligned}$$

Experimental analysis of RxOy (RyOx) algorithms starting from the LP relaxations of ILP1 and ILP2 are discussed in the next section.

## 5 Computational results

Although the primary focus of the paper is on theoretical analysis of approximation algorithms, we have conducted preliminary experimental analysis with the RxOy and RyOx rounding algorithms to examine features that are not clear from theoretical analysis. The alternating algorithm and local search algorithms are thoroughly analyzed from an experimental analysis point of view in [19] and [?] and hence are not considered in our experimental study. The rounding algorithms being simple constructive heuristics, they are not expected to outperform more sophisticated algorithms that employ powerful neighborhoods within a metaheuristic framework. Nonetheless, it is interesting to explore the behavior of these algorithms in the light its theoretical properties and the potential for computing initial solutions for more advanced solution improvement algorithms.

The data set used in our experiments are smallsize instances of random problems, biclique instances, matrix factorization instances, and maxcut instances from [19]. The algorithms are implemented in C# and tested on a DELL PC with Windows 7 operating system, Intel i7 processor and 16GB of memory. All CPU times reported are in milliseconds and do not include input-output times. We used the solution obtained by the linear programming relaxation of ILP1 as the fractional vector  $(x, y)$  to initiate the rounding process.

Our first set of experiments were aimed to identify the percentage of problems where LP relaxation produced optimal solutions. This data is important since for such problems, rounding (heuristic algorithms) is irrelevant. The results of these experiments are summarized in Figure 1. The trend shows that as the problem size increases, the number of problems where the LP relaxation produced optimal solution decreases. This decrease is more rapid in some class of problems compared to random instances.

Let us now discuss the experimental results with the RxOy and RyOx algorithms. The details are summarized in Tables 1 to 5. In each table, the column “best” reports the best known objective function value of the problem. Most of these values are optimal. The column “LP obj” contains the upper bound obtained by the LP relaxation. The columns “xy” and “yx” respectively contains the objective function value of the solution produced by the RxOy algorithm and the RyOx algorithms. It may be noted that the solution produced by these algorithms are close to the best known solution values and the running time is very minimal. This makes these algorithms good candidates to be

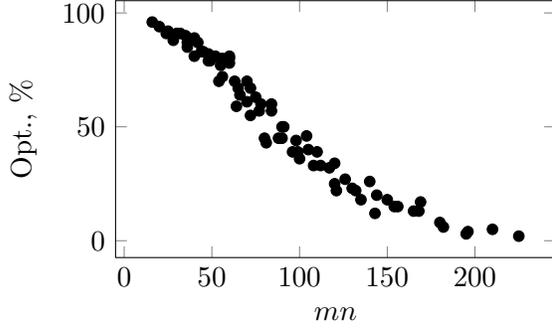


Figure 1: Random

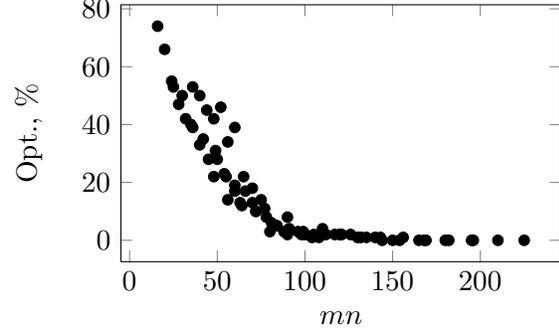


Figure 2: Matrix Factorization

used to generate starting solutions for more sophisticated algorithms. The table also provide insight into the value  $\mathcal{A}(Q, c, d)$  in comparison to the best known objective function value. The column “avg” reports  $\mathcal{A}(Q, c, d)$  and the column “Avg+” reports the value of the upper bound on  $\mathcal{A}(Q, c, d)$  provided by Theorem 12. Note that Avg+ is the objective function value of the solution reported in Corollary 13. These are certainly inferior solutions. The column “ $x_i y_j$ ” represents  $f(x, y)$  for the LP relaxation solution  $(x, y)$ . The experimental results conclusively demonstrate the power of RxOy and RyOx algorithms as a very fast stand alone heuristics or as an algorithm for generating starting solutions in more complex algorithms.

Instance	Objective							Time, ms		
	Best	LP obj.	$x_i y_j$	yx	xy	Avg+	Avg	LP	yx	xy
$20 \times 50$	13555	21179	437.5	12715	12208	2160	624.0	151	171	146
$25 \times 50$	13207	25405	759.0	12226	11016	2911	505.0	69	77	87
$30 \times 50$	15854	29713	634.0	13992	14824	3219	778.0	102	105	102
$35 \times 50$	14136	33471	-2146.0	12554	12099	835	-2028.5	137	143	131
$40 \times 50$	18778	39444	42.8	16562	15573	624	125.8	168	183	170
$45 \times 50$	22057	44760	-401.0	17688	19660	610	-440.0	212	236	200
$50 \times 50$	23801	50390	1576.0	21645	22178	5115	1576.0	234	272	233
Average	17341	34909	128.9	15340	15365	2211	162.9	154	169	153

Table 1: Random

Instance	Objective							Time, ms		
	Best	LP obj.	$x_i y_j$	yx	xy	Avg+	Avg	LP	yx	xy
20 × 50	18341	29999	-1911363.0	4933	0	0	-1911363.0	47	49	47
25 × 50	24937	38513	-3265058.5	2764	0	0	-3265058.5	66	90	65
30 × 50	27887	51726	-2981174.8	9050	15138	0	-2981174.8	97	95	95
35 × 50	32515	57302	-3782013.0	3270	10502	0	-3782013.0	135	134	134
40 × 50	33027	61966	-4562391.5	0	0	0	-4562391.5	133	131	133
45 × 50	37774	67923	-5789225.5	4420	0	0	-5789225.5	140	141	140
50 × 50	30124	78745	-5004969.5	5517	5833	0	-5004969.5	200	195	200
Average	29229	55168	-3899456.5	4279	4496	0	-3899456.5	117	119	1160

Table 2: Max Biclique

Instance	Objective							Time, ms		
	Best	LP obj.	$x_i y_j$	yx	xy	Avg+	Avg	LP	yx	xy
20 × 50	6983	10704	-315.3	5223	5481	0	-393.3	110	109	111
25 × 50	8275	13866	-678.5	7630	7350	0	-678.5	169	166	164
30 × 50	10227	18958	-140.0	8378	8685	0	-140.0	232	234	228
35 × 50	11897	20590	-543.8	11071	9008	0	-543.8	283	283	277
40 × 50	14459	23110	892.3	13216	13433	3569	892.3	506	524	498
45 × 50	13247	24393	-984.5	12496	9513	0	-984.5	430	431	428
50 × 50	15900	28875	356.3	14075	15357	1425	356.3	628	622	621
Average	11570	20071	-201.9	10298	9832	713	-213.1	337	338	333

Table 3: Max Induced Subgraph

Instance	Objective							Time, ms		
	Best	LP obj.	$x_i y_j$	yx	xy	Avg+	Avg	LP	yx	xy
20 × 50	9008	21266	-791.0	3838	4928	0	-780.0	93	83	89
25 × 50	10180	27546	-1205.0	4062	6410	0	-1352.0	123	130	120
30 × 50	13592	37676	-1172.0	5808	7560	0	-274.0	205	184	183
35 × 50	14024	40900	-1196.0	5816	8578	0	-1105.0	331	267	269
40 × 50	17610	45948	1133.0	9424	12082	3568	1784.0	334	290	297
45 × 50	15252	48492	-1209.0	6204	9790	0	-1956.0	441	402	402
50 × 50	19580	57408	723.0	12332	10364	1446	723.0	591	538	536
Average	14178	39891	-531.0	6783	8530	716	-422.9	303	271	271

Table 4: MaxCut

Instance	Objective							Time, ms		
	Best	LP obj.	$x_i y_j$	yx	xy	Avg+	Avg	LP	yx	xy
$20 \times 50$	114	249	-1.5	106	94	0	-1.5	37	32	31
$25 \times 50$	127	314	1.5	109	96	6	1.5	45	40	42
$30 \times 50$	148	373	-2.5	124	140	0	-2.5	65	61	61
$35 \times 50$	139	425	-13.0	106	122	0	-13.0	80	77	78
$40 \times 50$	210	512	12.0	182	194	48	12.0	148	73	71
$45 \times 50$	191	559	-3.5	165	179	0	-3.5	100	93	94
$50 \times 50$	217	637	11.5	181	191	46	11.5	120	119	119
Average	164	438	0.6	139	145	14	0.6	85	71	71

Table 5: Matrix Factorization

We have also conducted experiments with RxOy and RyOx rounding algorithms using the LP relaxation solution of ILP2. For random instances, the LP relaxation solution of ILP1 and ILP2 were different but for all other test instances, they produced same solutions most of the time. In some random instances considered, the rounding algorithms produced better solutions when started ILP2 LP relaxation solution in comparison to the solutions produced from ILP1 LP relaxation. However, in general we did not see significant performance difference and hence preferred ILP1 formulation, considering its smaller size.

Note that the RxOy and RyOx rounding algorithms can be initiated using any  $x^0 \in \mathbb{U}^m$  and  $y^0 \in \mathbb{U}^n$  and not necessarily a fractional solution for an LP relaxation. For  $i = 1, 2, \dots, m$  let  $\gamma_i = c_i + \sum_{j \in \mathcal{N}} q_{ij}$  and for  $j = 1, 2, \dots, n$  let  $\delta_j = d_j + \sum_{i \in \mathcal{M}} q_{ij}$ . Also,  $\text{ran}[a, b]$  represents a uniformly distributed random number in the interval  $[a, b]$ . Using these values we generate random vectors  $(x^0, y^0)$  as follows:

**Type 1 random vectors:** Choose

$$x_i^0 = \begin{cases} \text{ran}(0, 0.5] & \text{if } \gamma_i < 0, \\ \text{ran}(0.5, 1] & \text{otherwise,} \end{cases} \quad \text{and} \quad y_j^0 = \begin{cases} \text{ran}(0, 0.5] & \text{if } \delta_j < 0, \\ \text{ran}(0.5, 1] & \text{otherwise.} \end{cases}$$

**Type 2 random vectors:** These are generated using weighted random numbers with weight proportional to the corresponding  $\gamma_i$  or  $\delta_j$  values. Let  $\beta_1 = \max\{|\gamma_i| : i \in M\}$  and  $\beta_2 = \max\{|\delta_j| : j \in \mathcal{N}\}$ . Now choose

$$x_i^0 = \begin{cases} 0.5 - \frac{|\gamma_i|}{\beta_1} \text{ran}(0, 0.5] & \text{if } \gamma_i < 0, \\ \frac{\gamma_i}{\beta_1} \text{ran}(0.5, 1] & \text{otherwise,} \end{cases} \quad \text{and} \quad y_j^0 = \begin{cases} 0.5 - \frac{|\delta_j|}{\beta_2} \text{ran}(0, 0.5] & \text{if } \delta_j < 0, \\ \frac{\delta_j}{\beta_2} \text{ran}(0.5, 1] & \text{otherwise.} \end{cases}$$

The RxOy and RyOx algorithms using the above choices of starting solution  $(x^0, y^0)$  can also be used to construct starting solutions for advanced algorithms. The built-in randomness generates good solutions that can be embedded in metaheuristics with multiple starts. Systematic experimental analysis of such sophisticated algorithms is beyond the scope of this paper.

## 6 Conclusion

In this paper we studied approximation algorithms for BBQP which is a generalization of the well known BQP. Various approximation algorithms are analyzed using averaged value based measures and domination analysis. It is demonstrated that very powerful local search algorithms could get trapped at poor quality local minimum even if we allow exponential time in searching a very large scale neighborhood. Some of the proof techniques used are simple yet elegant and could be of use in domination analysis of heuristics for other related problems. Experimental results with two construction algorithms are also given. A natural question for further investigation is to close the gap between non-approximability bounds and lower bounds on domination ratio. Since BBQP is not as well studied as BQP, there are many other avenues for further investigation and we are currently investigating further properties of BBQP.

## References

- [1] R.K. Ahuja, O. Ergun, J.B. Orlin, and A.P. Punnen, Very Large Scale Neighborhood Search: Theory, Algorithms and Applications, *Approximation Algorithms and Metaheuristics*, T. Gonzalez (ed), CRC Press, 2007.
- [2] N. Alon, G. Gutin and M. Krivelevich, Algorithms with large domination ratio, *Journal on Algorithms* 50 (2004) 118-131.
- [3] N. Alon and A. Naor, Approximating the cut-norm via Grothendieck's inequality, *SIAM Journal of Computing* 35 (2006) 787-803.
- [4] C. Ambühl, M. Mastrolilli, and O. Svensson, Inapproximability results for maximum edge biclique, minimum linear arrangement, and sparsest cut, *SIAM Journal of Computing* 40 (2011) 567-596.
- [5] E. Angel and V. Zissimopoulos, On the quality of local search for the quadratic assignment problem, *Discrete Applied Mathematics*, 82 (1995) 15–25.
- [6] G. Bendall and F. Margot, Minimum number of below average triangles in a weighted complete graph. *Discrete Optimization* 3(2006) 206–219.
- [7] D. Berend, S. Skiena, Y. Twitto, Combinatorial dominance guarantees for problems with infeasible solutions, *ACM Transactions on Algorithms* 5 (2008) Article 8.
- [8] M. Demange, P. Grisoni, and V. Th. Paschos, Differential approximation algorithms for some combinatorial optimization problems, *Theoretical Computer Science* 209 (1998) 107–122.
- [9] V. G. Deneko and G. J. Woeginger, A study of exponential neighborhoods for the Travelling Salesman Problem and for the Quadratic Assignment Problem, *Mathematical programming* 87 (200) 519–542.
- [10] N. Gillis and F. Glineur. Low-rank matrix approximation with weights or missing data is NP-hard. *SIAM Journal of Matrix Analysis and Applications* 32 (2011) 1149–1165.
- [11] F Glover and AP Punnen The travelling salesman problem: new solvable cases and linkages with the development of approximation algorithms *Journal of the Operational Research Society* 48 (1997), 502–510
- [12] F. Glover, G.A. Kochenberger, B. Alidaee, Adaptive memory tabu search for binary quadratic programs, *Management Science* 44 (1998) 336–345.
- [13] F. Glover, T. Ye, A.P. Punnen, and G.A. Kochenberger, Private communication.
- [14] L. K. Grover, Local search and the local structure of NP-complete problems, *Operations Research Letters* 12 (1992) 235–243.
- [15] G. Gutin, A. Vainshtein and A. Yeo, Domination analysis of combinatorial optimization problems *Discrete Applied Mathematics* 129 (2003) 513–520.
- [16] G Gutin and A Yeo, Polynomial approximation algorithms for the TSP and the QAP with a factorial domination number *Discrete Applied Mathematics* 119 (2002) 107–116.

- [17] R. Hassin and S. Khuller,  $z$ -Approximations *Journal of Algorithms* 41 (2001) 429-442.
- [18] D. Kuhn and D. Osthus, Hamilton decompositions of regular expanders: a proof of Kelly's conjecture for large tournaments, Manuscript, School of Mathematics, University of Birmingham, UK, 2012.
- [19] D. Karapetyan and A.P. Punnen, Heuristic algorithms for the bipartite unconstrained 0-1 quadratic programming problem, (2012) <http://arxiv.org/abs/1210.3684>
- [20] A.E. Koller and S.D. Noble, Domination analysis of greedy heuristics for the frequency assignment problem, *Discrete Mathematics* 275 (2004) 331-338
- [21] H. Lu, J. Vaidya, V. Atluri, H. Shin, and L. Jiang, Weighted rank-one binary matrix factorization
- [22] T. R. Minina and V.T. Perekrest, On a method of approximating solutions of the travelling salesman problem, *Soviet Math. Doklady* 16 (1975) 26–30.
- [23] J.B. Orlin, A.P. Punnen, and A. Schulz, Integer programming: optimization and evaluation are equivalent, *Springer LNCS*, Proceedings of WADS (2009).
- [24] A.P. Punnen and S. Kabadi, Domination analysis of some heuristics for the asymmetric travelling salesman problem *Discrete Applied Mathematics* 119 (2002) 117-128.
- [25] A.P. Punnen, F. Margot and S.N. Kabadi, TSP heuristics domination analysis and complexity *Algorithmica* 35 (2003) 111-127.
- [26] A.P. Punnen, P. Sripratak, and D. Karapetyan. The bipartite unconstrained 0-1 quadratic programming problem: polynomially solvable cases. (2012) <http://arxiv.org/abs/1212.3736>.
- [27] P. Raghavendra and D. Steurer, Towards computing the Grothendieck constant. In Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '09). Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 525-534, 2009.
- [28] V.I. Rublinekii, Estimates of the accuracy of procedures in the traveling salesman problem, *Numerical Mathematics and Computer Technology* 4 (1973) 18-23 (in Russian).
- [29] V. Sarvanov and N. Doroshko, The approximate solution of the traveling salesman problem by a local algorithm that searches neighborhoods of exponential cardinality in quadratic time, *Software: Algorithms and Programs* 31 (1981) 8-11 (in Russian).
- [30] V. Sarvanov and N. Doroshko, The approximate solution of the traveling salesman problem by a local algorithm that searches neighborhoods of factorial cardinality in cubic time, *Software: Algorithms and Programs* 31 (1981) 11-13 (in Russian).
- [31] V.I. Sarvanov, The mean value of the functional in sampling problems, *Vestsi Akademii Navuk BSSR. Seryya Fizika-Matematychnykh Navuk* 139 (1978) 51-54.
- [32] B.-H. Shen, S. Ji, and J. Ye. Mining discrete patterns via binary matrix factorization. In *Proc. of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* 757–766, ACM, 2009.

- [33] P. Sripratak and A.P. Punnen, The bipartite boolean quadric polytope: facets and valid inequalities, Working paper, Department of Matheamtics, Simon Fraser university Surrey, 2013.
- [34] J. Tan. Inapproximability of maximum weighted edge biclique and its applications. In Proceedings of the 5th international conference on Theory and applications of models of computation (TAMC'08), Manindra Agrawal, Dingzhu Du, Zhenhua Duan, and Angsheng Li (Eds.). Springer-Verlag, Berlin, Heidelberg, 282-293, 2008.
- [35] A. Tanay, R. Sharan, and R. Shamir, Discovering statistically significant biclusters in gene expression data, *Bioinformatics* 18 (2002) Suppl. 1, S136-S144.
- [36] Y. Twitto, Dominance guarantees for above-average solutions, *Discrete Optimization* 5 (2008) 563-568.
- [37] V. Vazirani, *Approximation Algorithms*, Springer (2001).
- [38] V.G. Vizing, Values of the target functional in a priority problem that are majorized by the mean value, *Kibernetika* (1973) 76–78.
- [39] Y. Wang, Z. Lu, F. Glover, J-K Hao, Path relinking for unconstrained binary quadratic programming, *European Journal of Operational Research* 223 (2012) 595–604.
- [40] E. Zemel, Measuring the quality of approximate solutions to zero-one programming problems *Mathematics of Operations Research* 6 (1981) 319-332.