Quadratic Bottleneck Problems

Abraham P. Punnen, Ruonan Zhang

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

Received 2 September 2009; revised 12 December 2010; accepted 14 December 2010
DOI 10.1002/nav.20446
Published online 10 February 2011 in Wiley Online Library (wileyonlinelibrary.com).

Abstract: We study the quadratic bottleneck problem (QBP) which generalizes several well-studied optimization problems. A weak duality theorem is introduced along with a general purpose algorithm to solve QBP. An example is given which illustrates duality gap in the weak duality theorem. It is shown that the special case of QBP where feasible solutions are subsets of a finite set having the same cardinality is NP-hard. Likewise the quadratic bottleneck spanning tree problem (QBST) is shown to be NP-hard on a bipartite graph even if the cost function takes 0–1 values only. Two lower bounds for QBST are derived and compared. Efficient heuristic algorithms are presented for QBST along with computational results. When the cost function is decomposable, we show that QBP is solvable in polynomial time whenever an associated linear bottleneck problem can be solved in polynomial time. As a consequence, QBP with feasible solutions form spanning trees, s-t paths, matchings, etc., of a graph are solvable in polynomial time with a decomposable cost function. We also show that QBP can be formulated as a quadratic minsum problem and establish some asymptotic results. Naval Research Logistics 58: 153–164, 2011

Keywords: combinatorial optimization; bottleneck problems; bottleneck extrema; quadratic optimization

1. INTRODUCTION

Let $E = \{1, 2, \ldots, m\}$ be a finite set and $\mathfrak{F}$ be a family of subsets of E. The elements of $\mathfrak{F}$ are called feasible solutions. Let f be a real-valued function defined on E. Then the linear bottleneck problem (LBP) is to:

$$\underset{S \in \mathfrak{F}}{\text{Minimize}}\ \max\{f(x)\ :\ x \in S\}$$

It is assumed that $\mathfrak{F}$ is given in a compact form of size polynomial in m. A family of subsets of E is called a clutter if no member of the family is contained in another member of the family. Thus without loss of generality we assume that $\mathfrak{F}$ is a clutter. For the clutter $\mathfrak{F}$ on E, its blocking clutter (blocker) is the unique clutter $\mathfrak{B} = \{S \subseteq E : |S \cap T| \geq 1\ \forall T \in \mathfrak{F}$ and S is minimal$\}$. Edmonds and Fulkerson [12] established a strong duality relationship between a clutter-blocker pair for LBP. Since then, LBP has been studied systematically by various authors for specially structured family $\mathfrak{F}$ of feasible solutions [10, 13, 15, 19, 20, 25, 27, 28, 30–32, 34, 38].

In this paper, we study the Quadratic Bottleneck Problem (QBP). For any subset S of E, let $P[S]$ be the set of all subsets of S containing exactly 2 elements, i.e., $P[S] = \{T \subseteq S : |T| = 2\}$. Let $f(\{x, y\})$ be a function from $P[E]$ to $\Re^+ \cup \{0\}$. Then QBP is defined as follows:

$$\underset{S \in \mathfrak{F}}{\text{Minimize}}\ \max\{f(\{x, y\}) : \{x, y\} \in P[S]\}$$

Note that unlike LBP, costs for QBP are defined for pairs of elements of E. Let $\pi : P[E] \to N$ be a one to one correspondence where $N = \{1, 2, \ldots, r\}$ and $r = \binom{m}{2}$. For each $S \in \mathfrak{F}$, consider the subset S' of N defined by $S' = \{\pi(x, y) : \{x, y\} \in P[S]\}$. Let $\mathfrak{H} = \{S' : S \in \mathfrak{F}\}$. Thus $\mathfrak{H}$ is a family of subsets of N. Then the QBP can be viewed as a standard bottleneck problem on the system $(N, \mathfrak{H})$ and hence the duality theorem of Edmonds and Fulkerson [12] is applicable for an appropriate clutter-blocker pair defined for $(N, \mathfrak{H})$. However, the structure of this clutter-blocker pair in terms of the original system $(E, \mathfrak{F})$ is not clearly understood. Burkard [3] made this observation and stated that further understanding of the structure of this clutter-blocker pair (in terms of the original clutter-blocker pair) could lead to significant algorithmic developments.

The special case of QBP, where feasible solutions are perfect matchings of a bipartite graph, was studied by Steinberg [36] in the context of backboard wiring problem by formulating it as a quadratic bottleneck assignment problem (QBAP). Kellerer and Wirsching [26] used QBP to solve bandwidth minimization of matrices. The problem was

Correspondence to: A.P. Punnen (apunnen@sfu.ca)

further investigated by Burkard and Finoke [4]. Balanced optimization problems introduced by Martello et al. [29] can also be viewed as a special case of QBP. Recently, several authors considered the quadratic spanning tree problem [2,40]. Applications of the quadratic spanning tree model have a natural interpretation in terms of the QBP objective function as well. Similarly, the QBP counterpart of the quadratic knapsack problem [17] is a meaningful model to consider in situations where the cost of a solution is proportional to the largest cost of pairs of items in the solution rather than the average cost of pairs. However, QBP is not systematically studied in literature.

In this article, we study QBP from a theoretical and experimental point of view. We establish a weak duality theorem for QBP followed by a general purpose algorithm, called the quadratic threshold algorithm to solve it. The algorithm solves a sequence of quadratic feasibility problems, which can be viewed as the feasibility version of combinatorial optimization problems with conflict pair constraints [8,9,39]. We then consider two special cases. The first special case is when feasible solutions are restricted to all subsets of a finite set having the same cardinality. This problem is shown to be NP-hard. The second special case is when feasible solutions are spanning trees of a graph [quadratic bottleneck spanning trees (QBST)], which is also shown to be NP-hard even on bipartite graphs with 0-1 costs. Further, we rule out the existence of a polynomial time ϵ-approximation algorithm for the problem (modulo P=NP). Two lower bounds and two heuristic algorithms are given for QBST along with detailed computational results. When the cost function f is decomposable, we show that QBP is polynomially solvable whenever an associated linear bottleneck problem is solvable in polynomial time. Finally, it is observed that QBP can be formulated as a quadratic minsum problem and some asymptotic results are presented.

For any graph G, we sometimes use the notation $V(G)$ to represent its node set and $E(G)$ to represent its edge set. If S is a subgraph of G, to simplify the notation, we sometimes represent $E(S)$ simply by S, when the context raises no confusion.

2. WEAK DUALITY AND THE QUADRATIC THRESHOLD ALGORITHM

Let us first consider a weak duality result for QBP. For any $S \subseteq E, T \subseteq E$ denote $P[S,T] = \{\{x,y\} : x \in S, y \in T, x \neq y\}$.

THEOREM 1: For any clutter $\mathfrak{F}$ and its blocker $\mathfrak{B} = b(\mathfrak{F})$ on E

$$\max_{\substack{R,S\in\mathfrak{B}\\ R\neq S}} \min_{\{x,y\}\in P[R,S]} f(\{x,y\}) \leq \min_{T\in\mathfrak{F}} \max_{\{x,y\}\in P[T]} f(\{x,y\})$$

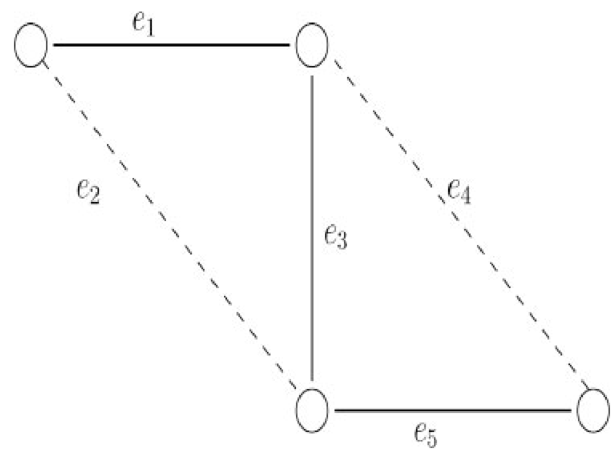

Figure 1. An example of duality gap.

PROOF: By definition, for any $T \in \mathfrak{F}$ and any $R, S \in \mathfrak{B}$ we have $T \cap R \neq \emptyset$ and $T \cap S \neq \emptyset$. Thus

$$\min_{\{x,y\}\in P[R,S]} f(\{x,y\}) \leq \max_{\{x,y\}\in P[T]} f(\{x,y\}).$$

Since T, R and S are arbitrary, we have

$$\max_{\substack{R,S\in\mathfrak{B}\\ R\neq S}} \min_{\{x,y\}\in P[R,S]} f(\{x,y\}) \leq \min_{T\in\mathfrak{F}} \max_{\{x,y\}\in P[T]} f(\{x,y\})$$

□

The weak duality theorem discussed above has duality gap as illustrated in the following example. Consider the graph G on four nodes given in Fig. 1. Choose $\mathfrak{F}$ as the collection of all spanning trees of G. Its blocker $\mathfrak{B}$ is the collection of all cuts in G.

Let $x, y \in \{e_1, e_2, e_3, e_4, e_5\}$ and the matrix

$$C = (c_{xy}) = \begin{bmatrix} 0 & 2 & 0 & 2 & 1 \\ 2 & 0 & 2 & 0 & 2 \\ 0 & 2 & 0 & 2 & 0 \\ 2 & 0 & 2 & 0 & 2 \\ 1 & 2 & 0 & 2 & 0 \end{bmatrix}$$

gives the cost of pairs of edges of G. Thus $f(\{x,y\}) = c_{xy}$. It may be verified that

$$\min_{T\in\mathfrak{F}} \max_{\{x,y\}\in P[T]} f(\{x,y\}) = 1$$

whereas

$$\max_{\substack{R,S\in\mathfrak{B}\\ R\neq S}} \min_{\{x,y\}\in P[R,S]} f(\{x,y\}) = 0$$

We now discuss an algorithm to solve QBP. Let $z_1 < z_2 < \cdots < z_p$ be an ascending arrangement of distinct costs $\{f(\{x, y\}) : \{x, y\} \in P[E]\}$. For any non-negative integer k, let $Q(k) = \{\{x, y\} \in P[E] : f(\{x, y\}) > z_k\}$. Thus $Q(p) = \emptyset$. Let $F_k = \{T \in \mathfrak{F} : P[T] \cap Q(k) = \emptyset\}$. In other words, F_k contains all solutions in F with QBP objective function value less than or equal to z_k. Note that $F_1 \subseteq F_2 \subseteq \cdots \subseteq F_p$ where $F_p = \mathfrak{F}$.

THEOREM 2: For $1 \leq k \leq p$, if k is the largest index such that $F_k = \emptyset$ then any $S \in F_{k+1}$ is an optimal solution to the QBP. Further if $F_q = \emptyset$ for some q then $F_k = \emptyset$ for all $k \leq q$.

The proof of the above theorem is straightforward. The quadratic feasibility problem QFP(k) associated with QBP can be described as follows: "Given $1 \leq k \leq p$, determine if F_k is empty." Equivalently $QFP(k)$ verifies that for a given $k, 1 \leq k \leq p$ and the family $\mathfrak{F}$, does there exist a feasible solution $S \in \mathfrak{F}$ which does not contain both x and y for any set $\{x, y\} \in Q(k)$? Note that such an S may contain either x or y or none, but not both.

Using Theorem 2, it is easy to see that QBP can be solved as a sequence of quadratic feasibility problems by performing binary search over $z_1 < z_2 < \cdots < z_p$. A formal description of our algorithm is given below.

Algorithm 1 Algorithm Quadratic Threshold

1. **Input:** A family $\mathfrak{F}$ (in compact form), the cost function f and an oracle $\alpha(.)$ which with input k verifies if $F_k = \emptyset$ and output an "yes" or "no" answer along with an $S \in F_k$ whenever $F_k \neq \emptyset$.
2. **Output:** An optimal solution to QBP.
3. Construct an ascending arrangement $z_1 < z_2 < \cdots < z_p$ of distinct costs $\{f(\{x, y\}) : \{x, y\} \in P[E]\}$.
4. $\ell = 1; u = p; k = \ell;$
5. **while** $u - \ell > 0$ **do**
6. $\quad k = \lfloor \frac{(l+u)}{2} \rfloor;$
7. $\quad$**if** $F_k = \emptyset$ **then** $\ell = k + 1$; **else** $u = k$
8. **end while**
9. Output any $T \in F_\ell$

THEOREM 3: QBP can be solved in polynomial time if and only if the the corresponding quadratic feasibility problem is polynomially solvable. Further, algorithm Quadratic Threshold computes an optimal solution to QBP in $O(\phi(m) \log m)$ time, where $O(\phi(m))$ is the complexity of QFP(k).

PROOF: Suppose that the quadratic feasibility problem can be solved in polynomial time. By Theorem 2, it can be verified that algorithm the Quadratic Threshold correctly solves QBP. To establish the complexity, note that the algorithm tests if $F_k = \emptyset$ by invoking QFP(k). Such a test is performed at most $O(\log(p))$ times. Since $p = O(m^2)$, $O(\log(p)) = O(\log(m^2)) = O(\log m)$, the complexity of the algorithm is $O(\phi(m) \log m)$. Thus if $\phi(m)$ is polynomial then QBP is solvable in polynomial time.

Conversely, suppose QBP is solvable in polynomial time. Let S^* be an optimal solution. Choose q such that $z_q = f(S^*)$. Then $F_k = \emptyset$ precisely when $k < q$. Thus the quadratic feasibility problem can be solved in polynomial time □

Although Theorem 3 appears to be promising, even for the case where the family of feasible solutions have a very simple structure, the quadratic feasibility problem (i.e., verifying if $F_k = \emptyset$) may be difficult. We illustrate this by considering the quadratic bottleneck problem with only cardinality restrictions. Suppose $\mathfrak{F}$ is the family of all subsets of E with cardinality at least q for some given q. We call the resulting QBP, the cardinality constrained QBP (CQBP). The corresponding quadratic feasibility problem is: "Given an instance of CQBP, represented by E and q, verify if there exist an $S \in \mathfrak{F}$ such that $g(S) \leq K$ for a given constant K." where $g(S) = \max\{f(\{x, y\}) : \{x, y\} \in P[S]\}$.

THEOREM 4: The quadratic feasibility problem of CQBP is NP-complete.

PROOF: We reduce the maximum independent set problem [18] to CQBP. Let G be a graph on which the maximum independent set problem is defined. Construct an instance of CQBP as follows. Choose the ground set E as the node set of G and $\mathfrak{F} = \{S : S \subseteq V(G), |S| \geq q\}$. Construct the cost $f(\{x, y\})$ for pairs of elements of E as

$$f(\{x, y\}) = \begin{cases} 1 & \text{if the nodes } x \text{ and } y \text{ are adjacent in } G \\ 0 & \text{otherwise} \end{cases}$$

Choose $K = 0$. Now G has an independent set of size greater than or equal to q if and only if there exists an $S \in \mathfrak{F}$ such that $g(S) \leq 0$. Since the maximum independent set problem is NP-complete [18], the result follows. □

3. POLYNOMIALLY SOLVABLE CASES

Although the problem CQBP appears to be the simplest non trivial QBP, theorem 4 shows that even this problem is hard. Despite this disappointment, it can be shown that there are nontrivial cases of QBP that can be solved in polynomial time.

One way to achieve polynomial solvability is by restricting the structure of $f(\{x, y\})$. Before considering specially structured $f(\{x, y\})$ let us briefly discuss four different variations of QBP. The QBP we discussed so far is called uniform symmetric QBP where no cost is considered for single elements of E. In addition to pairs of elements of E, if there is a cost associated for each element $x \in E$, then we have an instance of the nonuniform symmetric QBP. In this case we represent the cost of $x \in E$ as $f(\{x,x\})$ and the domain of the cost function as $P[E] \cup \{\{x,x\} : x \in E\}$. When the domain of the cost function f is $E \times E$ we have the nonuniform asymmetric QBP. In this case, to reflect the fact that (x, y) is an ordered pair, we write $f(x, y)$ in place of $f(\{x, y\})$. Finally, when the domain of the cost function is restricted to $E \times E \setminus \{(x,x) : x \in E\}$ we get the uniform asymmetric QBP. The following lemma shows that the four variations of QAP discussed above are equivalent in the sense that given any two variations, one can be reduced to another.

LEMMA 5: The four variations of QBP discussed above are equivalent.

PROOF: Consider a nonuniform asymmetric QBP with cost function f_1. Define the nonuniform symmetric cost function f_2 as $f_2(\{x, y\}) = \max\{f_1(x, y), f_1(y, x)\}$. It can be verified that an optimal solution to the nonuniform symmetric QBP with cost function f_2 is also an optimal to the asymmetric QBP with cost function f_1. Consider the uniform symmetric cost function f_3 defined as $f_3(\{x, y\}) = \max\{f_1(x, y), f_2(y, x), f_1(x, x), f_1(y, y)\}$ for $x \neq y$. It can be verified that an optimal solution to the uniform symmetric QBP with cost function f_3 is also an optimal to the nonuniform asymmetric QBP with cost function f_1. Reduction among other pairs of variations can be established by appropriate modifications of the above arguments and hence the details are omitted. □

Based on Lemma 5 without loss of generality one may assume that the cost functions are of uniform symmetric type. We concentrate on uniform cases primarily for notational clarity and continue to consider only uniform symmetric cases in all other sections. However, a distinction between the four versions of QBP is maintained in this section which is important for specially structured cost functions that we consider. In particular, maintaining asymmetric problems, without reducing them to the symmetric case is advantageous in some cases, as we illustrate later in this section.

A nonuniform asymmetric cost function f is said to be decomposable if there exist real numbers a_x and b_x for each $x \in E$ such that $f(x, y) = a_x + b_y$ for all $x \in E, y \in E$. A nonuniform symmetric cost function f is said to be decomposable if there exist a real number a_x for each $x \in E$ such that $f(\{x, y\}) = a_x + a_y$ for all $x, y \in E$.

THEOREM 6: A nonuniform asymmetric QBP with a decomposable cost function f is solvable in $O(m\zeta(m))$ time if an associated linear bottleneck problem can be solved in $O(\zeta(m))$ time.

PROOF: For any feasible solution S

$$\begin{aligned} g(S) &= \max\{f(x, y) : x \in S, y \in S\} \\ &= \max\{a_x + b_y : x \in S, y \in S\} \\ &= \max\{a_x : x \in S\} + \max\{b_y : y \in S\} \end{aligned}$$

Thus $g(S)$ decomposes into sum of two maximum functions. Then minimizing $g(S)$ over $\mathfrak{F}$ is a special case of the ξ-deviation problem [11]. Using the results of [11], minimization of $g(S)$ can be done by solving $O(m)$ linear bottleneck problems of the type

$$\underset{S\in\mathfrak{F}}{\text{Minimize}} \max\{b_x : x \in S\}.$$

Thus QBP can be solved in $O(m\zeta(m))$ time. □

The complexity of algorithms for decomposable instances may further be improved by exploiting the special structure, if any, for the set $\mathfrak{F}$. By a straightforward application of Theorem 6, the nonuniform asymmetric quadratic bottleneck spanning tree problem on a graph with n nodes and m edges under a decomposable cost function can be solved in $O(m^2)$ time since the linear bottleneck spanning tree problem can be solved in $O(m)$ time [32]. The algorithm for quadratic bottleneck spanning tree problem for this special case can be improved to run in $O(m \log n)$ time using dynamic tree data structure along the lines of [16, 35], where n is the number of nodes in the underlying graph. It may be noted that when $b_x = -a_x$, the the nonuniform asymmetric QBP with decomposable cost function reduces to the balanced optimization problem [6, 29].

To take advantage of theorem 6 we need a way to test if a given cost function is decomposable or not. We say that an $m \times m$ matrix $C = (c_{ij})$ is decomposable if and only if there exists constants $a_1, a_2, \ldots, a_m$ and $b_1, b_2, \ldots, b_m$ such that $c_{ij} = a_i + b_j$. Further, if C is symmetric then $a_i = b_i$ holds for all i. The following lemma allows us to test for decomposability of a cost function.

LEMMA 7: Let $C = c_{ij}$ be an $m \times m$ matrix with (i, j)th entry $f(i, j)$. Let $\hat{C} = \hat{c}_{ij}$ be defined as $\hat{c}_{ij} = c_{ij} - (a_i + b_j)$, where $a_i = c_{im} - \frac{c_{mm}}{2}$, $b_j = c_{mj} - \frac{c_{mm}}{2}$. Then C is decomposable if and only if $\hat{c}_{ij} = 0$ for $i, j = 1, \ldots, m$.

PROOF: Suppose C is decomposable, then $\exists \alpha_i, \beta_i, i = 1, \ldots, m$ such that $c_{ij} = \alpha_i + \beta_j$. Now

$$\begin{aligned} \hat{c}_{ij} = c_{ij} - (a_i + b_j) &= c_{ij} - \left(c_{im} - \frac{c_{mm}}{2} + c_{mj} - \frac{c_{mm}}{2}\right) \\ &= c_{ij} - c_{im} - c_{mj} + c_{mm} \\ &= \alpha_i + \beta_j - \alpha_i - \beta_m - \alpha_m - \beta_j + \alpha_m + \beta_m \\ &= 0. \end{aligned}$$

Conversely, suppose $\hat{c}_{ij} = c_{ij} - (a_i + b_j) = 0$, then clearly $c_{ij} = a_i + b_j$ holds and hence C is decomposable. □

Conditions similar to lemma 7 have been studied in the context of traveling salesman problem by various authors [14,22,24].

Note that f is decomposable if and only if its associated cost matrix $C = (c_{ij})$ defined by $f(i,j) = c_{ij}$ is decomposable and hence decomposability of f can be tested in $O(m^2)$ time. Based on Lemma 5, one might wonder why asymmetric cost functions need to be considered at all. To answer this, consider the matrices C and D given by

$$C = \begin{bmatrix} 3 & 8 & 3 & 4 & 7 \\ 2 & 7 & 2 & 3 & 6 \\ 8 & 13 & 8 & 9 & 12 \\ 6 & 11 & 6 & 7 & 10 \\ 5 & 10 & 5 & 6 & 9 \end{bmatrix} \text{ and } D = \begin{bmatrix} 3 & 8 & 8 & 6 & 7 \\ 8 & 7 & 13 & 11 & 10 \\ 8 & 13 & 8 & 9 & 12 \\ 6 & 11 & 9 & 7 & 10 \\ 7 & 10 & 12 & 10 & 9 \end{bmatrix}$$

Now $c_{ij} = a_i + b_j$ with $(a_1, a_2, a_3, a_4, a_5) = (2, 1, 7, 5, 4)$ and $(b_1, b_2, b_3, b_4, b_5) = (1, 6, 1, 2, 5)$ and hence C is decomposable but Lemma 7 guarantees that D is not decomposable. The matrix D is obtained from C by using the transformation described in Lemma 5. Let h be the nonuniform asymmetric cost function associated with C and f be the nonuniform symmetric cost function associated with D. Recall that the nonuniform cost functions include costs for single elements and the cost of element i is denoted as $h(i,i)$ ($f(i,i)$). Thus $h(i,i) = c_{ii}$ and $f(\{i,i\}) = d_{ii}$). By Lemma 5, QBP with cost functions h and f are equivalent. But if we are simply presented the QBP with cost function f, we do not have an obvious easy way to solve it, without probably using the quadratic threshold algorithm, but when presented with cost function h, based on Theorem 6 it can be solved as a sequence of linear bottleneck problems. Note that an LBP is much simpler compared to QBP. Thus, there are instances where it is advantageous to maintain the asymmetry of the cost function, without converting it into a symmetric problem.

Consider an asymmetric(symmetric) cost function f and the associated cost matrix D. Suppose f is not decomposable. Let L be a lower bound and U be an upper bound on the optimal objective function value of QBP. Consider the linear inequality system

$$\text{(LI)} \quad \begin{array}{lll} a_i + b_j < L, & \text{if } \max\{d_{ij}, d_{ji}\} < L & \text{(r1)} \\ a_i + b_j > U, & \text{if } \max\{d_{ij}, d_{ji}\} > U & \text{(r2)} \\ a_i + b_j = d_{ij} & \text{if } L \le d_{ji} \le d_{ij}\} \le U & \text{(r3)} \\ a_i + b_j \le d_{ji} & \text{if } L \le d_{ij} < d_{ji}\} \le U & \text{(r4)} \end{array}$$

where $a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_m$ are variables.

Suppose $a^0 = (a_1^0, a_2^0, \ldots, a_m^0)$, $b^0 = (b_1^0, b_2^0, \ldots, b_m^0)$ be a feasible solution to (LI). Let $c_{ij} = a_i^0 + b_j^0$, and h be the cost function associated with $C = c_{ij}$.

LEMMA 8: The QBP with cost functions f and h are equivalent.

PROOF: Note that the QBP with cost function f and h have the same solution set. For any feasible solution S, let $Z^1(S) = \max\{f(i,j) : i, j \in S\}$ and $Z^2(S) = \max\{h(i,j) : i, j \in S\}$. For any $i, j \in S$ clearly $Z^1(S) \ge \max\{d_{ij}, d_{ji}\}$. Thus by construction, for any feasible solution S satisfying $L \le Z^1(S) \le U$, we have $Z^1(S) = Z^2(S)$. The result now follows from the fact that L and U are respectively lower and upper bounds on the optimal objective function value of the QBP. □

Since h is decomposable the QBP is polynomially solvable whenever the associated linear bottleneck problem can be solved in polynomial time. In the example of the 5×5 matrix D given earlier, if we simply take the trivial lower bound $L = 3$ and trivial upper bound $U = 13$ the resulting (LI) is feasible with solution $a^0 = (2, 1, 7, 5, 4)$, $b^0 = (1, 6, 1, 2, 5)$. This verifies that we can get an equivalent decomposable problem from D. The difference between the lower bound L and upper bound U can affect the number of solution to the system (LI). When L and U are closer, there are less number of equality constraints in (r3) and hence the feasible region of (LI) is likely to be larger. So if we have heuristics to obtain high quality lower and upper bounds, the QBP is more likely to be solved in polynomial time, by finding an equivalent decomposable cost function.

THEOREM 9: A nonuniform symmetric QBP with a decomposable cost function f can be formulated as a linear bottleneck problem. Further, a uniform symmetric QBP with a decomposable cost function can be formulated as a 2-sum optimization problem.

PROOF: Let us first consider the nonuniform case. For any feasible solution S for the QBP, from the proof of theorem 6, we have

$$\begin{aligned} g(S) &= \max\{a_x : x \in S\} + \max\{a_y : y \in S\} \\ &= 2\max\{a_x : x \in S\}. \end{aligned}$$

Thus minimizing $g(S)$ is equivalent to minimizing $\max\{a_x : x \in S\}$, which is an LBP.

When the cost function is uniform, the restriction $x \neq y$ is retained in the definition of $P[E]$ and $f(\{x,y\})$. Thus, in this case, minimizing $g(S)$ is equivalent to finding a solution S where the sum of the largest and second largest (counting multiplicity) value of a_x for $x \in S$ is minimized. This is precisely the 2-sum optimization problem [21,33]. □

Note that a 2-sum optimization problem can be solved efficiently whenever an associated linear minsum problem can be solved efficiently [21, 33]. In the case of spanning trees, the 2-sum problem can be solved by solving just one minimum spanning tree problem [21]. It is also possible to solve this special 2-sum spanning tree problem in $O(m)$ time. We omit the details of this scheme, which can be obtained by suitably modifying the $O(m)$ algorithm for the bottleneck spanning tree problem [5] and exploiting the properties described in [21].

4. THE QUADRATIC BOTTLENECK SPANNING TREE PROBLEM

When E is the edge set of a graph G and $\mathfrak{F}$ is the collection of all spanning trees of G, QBP reduces to the quadratic bottleneck spanning tree problem (QBST). Although the bottleneck spanning tree problem can be solved in linear time [5], QBST seems more difficult. The recognition version of QBST, denoted by RQBST, can be stated as follows: "Given a graph G with cost function f for pairs of edges of G and a constant K, does there exist a spanning tree T in G such that $g(T) \leq K$?" where $g(T) = \max\{f(\{x, y\}) : \{x, y\} \in P[T]\}$.

THEOREM 10: RQBST is NP-complete on bipartite graphs even if f takes 0–1 values only.

PROOF: RQBST is clearly in NP. We now reduce the Hamiltonian path problem (HPP) on a directed graph to RQBST. The HPP can be stated as follows: "Given a directed graph D, does there exist a Hamiltonian path in D?" From an instance of HPP, we now construct an instance of RQBST. Let $V(D) = \{1, 2, \ldots, n\}$ be the node set of D and $E(D)$ be its arc set. From D, let us construct an undirected graph G on $2n$ nodes $\{1, 2, \ldots 2n\}$. For each arc (i, j) of D create an edge $(i, n + j)$ in G. Also introduce edges $(i, n + i)$ for $i = 1, 2, \ldots n$. For $i = 1, 2, \ldots n$, let $A(i) = \{(i, n + j) : (i, n + j) \in E(G), i \neq j\}$ and $A(n + i) = \{(j, n + i) : (j, n + i) \in E(G), i \neq j\}$. Note that the definition of $A(i)$ and $A(n + i)$ excludes the edge $(i, n + i)$ from it for all $i = 1, 2, \ldots, n$. Let f be the cost function for pairs of edges of G where

$$f(\{x, y\}) = \begin{cases} 1 & \text{if } \{x, y\} \in P[A(i)] \text{ for } i = 1, 2, \ldots 2n \\ 0 & \text{otherwise.} \end{cases}$$

Thus $f(\{x, y\}) = 0$ if and only if $x \in A(i), y \in A(j), i \neq j$. Choose $K = 0$. Note that edges in G incident on node i (other than the edge $(i, n + i)$) represent arcs going out of node i in D for $1 \leq i \leq n$. Similarly, edges in G incident on node $n + j$ (other than the edge $(j, n + j)$) represent arcs coming into node j in D, for $1 \leq j \leq n$. Suppose that G contains a spanning tree T such that $g(T) \leq 0$. Such a tree can choose at most one edge from $A(i)$ for $i = 1, 2, \ldots, 2n$. It can be verified that this is possible only if T is a Hamiltonian path in G and from this path, a Hamiltonian path of D can easily be recovered. Similarly, if D contains a Hamiltonian path, we can construct a spanning tree T in G from it by splitting each node i of the path using the edge $(i, n + i)$. Thus $g(T) = 0$. The proof follows from the NP-completeness of the Hamiltonian path problem on a directed graph [18]. □

COROLLARY 11: For any $\epsilon > 1$ computing an ϵ-optimal solution for $QBST$ is NP-hard.

PROOF: The proof of this corollary follows from the same construction as in the proof above using cost function

$$f(\{x, y\}) = \begin{cases} 1 + \epsilon & \text{if } \{x, y\} \in P[A(i)] \text{ for } i = 1, 2, \ldots 2n \\ 1 & \text{otherwise.} \end{cases}$$

It can be verified that an ϵ-optimal solution and an optimal solution to the QBST instance constructed cannot use any edge pair with cost more than 1 when D is Hamiltonian. If D is not Hamiltonian, they must use an edge pair of weight $1 + \epsilon$. Thus the QBST instance have an ϵ optimal solution with objective function value 1 if and only if D contains a Hamiltonian path and hence the proof follows. □

4.1. Algorithms for QBST

In this section, we specialize our general quadratic threshold algorithm to solve QBST and modify it to obtain an efficient heuristic algorithm. For simplicity of notation, we assume that a tree T is represented as its edge set. The crucial step in the quadratic threshold algorithm is solving the quadratic feasibility problem (QFP). For any real number K, let $Q(K) = \{\{x, y\} : f(\{x, y\}) > K\}$. In the context of QBST, we can rephrase the QFP as follows: "Does there exist a spanning tree T of G such that $P[T] \cap Q(K) = \emptyset$?" In other words, we want a spanning tree T of G that uses at most one edge out of any pair of edges $\{x, y\}$ in $Q(K)$. In view of Theorems 3 and 10, this problem is also NP-hard. The NP-hardness of this feasibility problem was independently established in [8, 9, 39] in the context of minimum spanning tree problem with conflict pairs.

The quadratic feasibility problem for QBST can be formulated as a quadratic spanning tree problem (QST):

$$\begin{aligned} &\text{Minimize } \textstyle\sum_{\{x,y\} \in P[T]} h(\{x, y\}) \\ &\text{Subject to} \\ &\qquad T \in \mathfrak{F}, \end{aligned}$$

where

$$h(\{x, y\}) = \begin{cases} 1 & \text{if } \{x, y\} \in Q(K) \\ 0 & \text{otherwise} \end{cases}$$

The quadratic feasibility problem has an yes answer if and only if QST has an optimal objective function value zero. Thus any algorithm for QST [2] can be used to solve the quadratic feasibility problem. Let us now consider another formulation of the problem.

Note that any spanning tree of G can be represented by a 0–1 vector $x = (x_1, x_2, \dots, x_m)$, called the incidence vector, where m is the number of edges of G and $x_e = 1$ if and only if $e \in T$. Let $\mathcal{F}$ be the convex hull of all incidence vectors of spanning trees of G. Consider the minimum spanning tree problem with conflict pairs (MSTC) [8, 9, 39]

MSTC Minimize $\sum_{e \in T} c_e$
Subject to

$$X \in \mathcal{F},$$
$$x_e + x_f \le 1 \text{ for all } \{e, f\} \in Q(K)$$
$$x_e, x_f \in \{0, 1\}$$

where $c_e = 0$ for all e. Note that MSTC has a feasible solution if and only if the quadratic feasibility problem has an "yes" answer. MSTC has been studied recently by Darmann et al. [8, 9] and Zhang et al. [39]. Various heuristic algorithms are developed in [39] to solve MSTC based on tabu search, tabu thresholding among other approaches. Thus, these algorithms can be employed to solve our quadratic feasibility problem within the quadratic threshold algorithm. If we use an exact algorithm to solve MSTC, the quadratic threshold algorithm computes an optimal solution to QBST. If a heuristic algorithm is used to solve MSTC, it provides a heuristic solution. Let us now have a closer look at this heuristic algorithm.

When an MSTC heuristic is used to solve the quadratic feasibility problem, and the answer is "yes," we make a correct decision in the quadratic threshold algorithm. But if the answer is "no," it is not guaranteed that the answer to the quadratic feasibility problem is indeed "no" and it is possible that the heuristic may have produced a "suboptimal" solution leading to a "no" answer. Since we are interested in just a feasible solution, we can try to run the MSTC heuristic again using a randomly generated cost vector c_e for $e \in E(G)$ with the hope that it may take the algorithm out of the current infeasible spanning tree for MSTC. We call this a shake operation. The shake operation can be repeated for a fixed number of iterations and if a "no" answer emerges consistently, we can conclude with high probability that the answer is indeed "no," especially when our MSTC heuristic is powerful. The performance of the heuristic algorithm (and the quadratic threshold algorithm) can be improved by computing good lower and upper bounds on the optimal objective function value and thereby reducing the search range.

For any two subsets R and S of E, let $P[R, S] = \{\{x, y\} : x \in R, y \in S, x \neq y\}$. A quick lower bound for QBST can be obtained as follows. Any spanning tree must have at least one edge incident on each node. For any node i, let $N(i)$ be the set of edges incident on i. For any two distinct nodes i and j, $\min\{f(\{x, y\}) : \{x, y\} \in P[N(i), N(j)]\}$ is a lower bound for the optimal objective function value of QBST. Thus

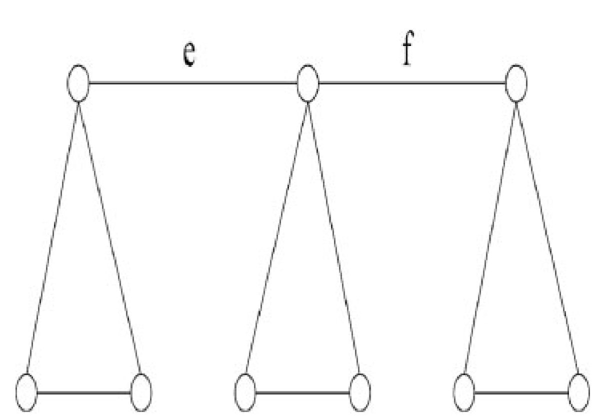

Figure 2. Comparison of lower bounds. The number e and f are edge labels.

$$\max_{i,j \in V(G), i \neq j} \min\{f(\{x, y\}) : \{x, y\} \in P[N(i), N(j)]\} \quad (1)$$

is a lower bound for the optimal objective function value of QBST. Note that for the family of spanning trees of the graph G the blocker is the family of cuts in G. Let $\mathfrak{D}$ be the collection of all cuts in G. Then by weak duality theorem,

$$\max_{\substack{R,S \in \mathfrak{D} \\ R \neq S}} \min\{f(\{x, y\}) : \{x, y\} \in P[R, S]\} \quad (2)$$

where $P[R, S] = \{\{x, y\} : x \in R, y \in S, x \neq y\}$ and R and S are two distinct cuts (represented as edge sets), gives another lower bound for the optimal objective function value of QBST. Let L_1 be the lower bound given in Eq. (1) and L_2 be the lower bound given Eq. (2). Since $[N(i), V(G) \setminus N(i)]$ is a cut, it can be verified that the L_2 is no worse than L_1. However it is easy to compute L_1 and hence we used it in our computational experiments.

The lower bound L_1 could be arbitrarily bad compared to the L_2 bound, in worst case. Consider graph in Fig. 2 formed by three triangles joined together by two bridges.

Set the cost of the edge pair $\{e, f\}$ to w and cost of all other pairs to be zero. Then the L_1 bound is zero whereas L_2 bound is w. Nevertheless, L_1 does not perform so bad in practice, as suggested by our experimental results.

We can speed up our main heuristic algorithm by constructing good upper bounds as well. For this we employ a simple heuristic: first generate a random tree T and $U_1 = g(T)$ is an obvious upper bound. To get a possibly improved upper bound U_2, we use binary search on distinct costs $\{z_1, z_2, \dots, z_t\}$ of the edge-pairs $\{x, y\}$ such that $f(\{x, y\}) \le U_1$. Initially set $\ell = 1$, $u = t$ and these values are updated appropriately in each iteration. Choose

$k = \lfloor \frac{(\ell+u)}{2} \rfloor$ and construct $Q(k)$ as the set of edge pairs $\{x, y\}$ where $f(\{x, y\}) > z_k$. Then from the original graph we delete all the edges associated with $Q(k)$ and denote $\tilde{E}$ the set of the remaining edges and $\tilde{G}$ as the resulting graph. If $\tilde{G}$ is connected then z_k is clearly an upper bound and hence we update u as k. If $\tilde{G}$ is disconnected then z_k may or may not be an upper bound. We proceed as if z_k is not an upper bound update ℓ as $k + 1$. This procedure is repeated until $\ell = u$ and we have the upper bound $U_2 = z_l$. This heuristic is summarized in Algorithm 2.

Algorithm 2 The upper bound algorithm

1. **Input:** A graph G and cost function f for pairs of arcs in G.
2. **Output:** An upper bound of the objective function value of QBST.
3. Randomly generate a spanning tree T, let $U_1 = \max_{\{x,y\} \in P[T]} f(\{x, y\})$;
4. Construct an ascending arrangement $z_1 < z_2 < \cdots < z_t$ of distinct costs $\{f(\{x, y\}) : \{x, y\} \in P[E(G)], f(\{x, y\}) \leq U_1\}$; $\ell = 1$; $u = t$;
5. **while** $u - \ell > 0$ **do**
6. $\quad k = \lfloor \frac{(l+u)}{2} \rfloor$;
7. $\quad Q(k) = \{\{x, y\} : \{x, y\} \in P[E(G)], f(\{x, y\}) > z_k\}$
8. $\quad \tilde{E} = \{q : q \notin \{x, y\}$ for any $\{x, y\} \in Q(k)\}$
9. $\quad \tilde{G}$ be the spanning subgraph of G with edge set $\tilde{E}$;
10. $\quad$ **if** $\tilde{G}$ is connected **then** $u = k$ **else** $\ell = k + 1$;
11. **end while**
12. Output $U_2 = z_\ell$

Since connectivity of a graph can be tested in linear time and sorting of edge-pair costs can be done in $O(m^2 \log n)$ the upper bound algorithm runs in $O(m^2 \log n)$ time. We now present our main heuristic algorithm (Algorithm 3), called quadratic threshold heuristic, to solve QBST.

Algorithm 3 The Quadratic Threshold Heuristic

1. **Input:** A graph G and cost function f for pairs of edges in G.
2. **Output:** A spanning tree T of G which is a heuristic solution for QBST.
3. Compute a lower bound L for the optimal objective function value. (Probably using equation (1));
4. Compute an upper bound U for the optimal objective function value. (Probably obtained from the upper bound algorithm);
5. Construct an ascending arrangement $z_1 < z_2 < \cdots < z_p$ of distinct costs $\{f(\{x, y\}) : \{x, y\} \in P[E(G)]\}$;
6. Choose ℓ and u such that $z_\ell = L,\ z_u = U$;
7. **while** $u - \ell > 0$ **do**
8. $\quad k = \lfloor \frac{(l+u)}{2} \rfloor$; count =0;
9. $\quad Q(k) = \{\{x, y\} : \{x, y\} \in P[E(G)], f(\{x, y\}) > z_k\}$
10. $\quad$ **repeat**
11. $\quad\quad$ **if** count = 0 **then**
12. $\quad\quad\quad d(e) = 0$ for all $e \in E$
13. $\quad\quad$ **else**
14. $\quad\quad\quad$ generate random costs $d(e)$ for $e \in E(G)$;
15. $\quad\quad$ **end if**
16. $\quad\quad$ Apply a MSTC heuristic on the minimum spanning tree problem with set $Q(k)$ of conflict pairs and edge costs $d(e)$. Let T^* be the output tree and $P[T^*]$ be the set of unordered pairs of edges of T^*;
17. $\quad\quad$ count = count + 1;
18. $\quad$ **until** $P[T^*] \cap Q(k) = \emptyset$ or count > max-count
19. $\quad$ **if** $P[T^*] \cap Q(k) = \emptyset$ **then** $u = k$; Tree $= T^*$; **else** $\ell = k + 1$;
20. **end while**
21. Output Tree

Note that in Step 16 of the algorithm, the MSTC heuristic may or may not find a feasible solution. If it failed to compute a feasible solution, we "heuristically" conclude that no feasible solution exist with the current threshold value. This imprecise step makes our Algorithm 3, a heuristic. If this step is done by using an exact algorithm for MSTC, then Algorithm 3 becomes an exact algorithm for QBST.

5. COMPUTATIONAL RESULTS FOR QBST

The quadratic threshold heuristic was coded in C++ and tested on a Dell PC with 3.40 GHz processor and 2.0 GB memory running on Windows XP operation system. We used the public domain compiler Dev C++ for all compilation work. No standard benchmark test problems are available for QBST. Thus we have generated random test instances as follows: 17 random connected graphs with the number of nodes ranging from 10 to 50 and the number of edges ranging from 20 to 500 are generated. Note that for instances with 500 edges, there are 124,750 variables (pairs of edges) and hence our problems are of reasonably large size. For each graph generated, we have 10 sets of costs, resulting in 10 different problems on the same graph. Costs $f(\{x, y\})$ are random integers in the interval $[0, r]$ where $r = \binom{m}{2}$ and m is the number of edges in the graph. In addition to these instances, we also generated special test problems where the optimal objective function value is known a priori. To construct these instances, we again used our 17 graphs generated for the problem set discussed earlier. For each graph G an arbitrary spanning tree T is selected and assigned costs for pairs of its edges $\{x, y\}$ a random integer in the interval $[0,n]$. All other edge pairs have cost a random integer in $[n + 1, r]$. Here n is the number of nodes in G. By construction, T is an optimal solution to the resulting instance of QBST.

The MSTC heuristic we used in the algorithm is a tabu search heuristic developed in [39]. The number of iterations in the tabu search is set to $\min\{100, \frac{m}{2}\}$ and the length of the tabu list (tabu size) is selected as 7. (We have compared the tabu search algorithm with tabu size 5, 7, and 10. It turns out that the solution is better when the tabu size is larger. With tabu size 7, the quality of the solution obtained was similar to that with tabu size 10, but the latter takes less time. Therefore we set the tabu size at 7.) The maximum number of "shake" operations, i.e. max-count, is set to 5, but to save the computational time, if the number of violated conflict pairs in MSTC returned by the tabu search is larger (say, greater than 10), then we assume it is not likely to have any feasible solutions and hence the rest of the shake operations were skipped.

Table 1 shows computational results for the general instances. For each problem, there are $\binom{m}{2}$ cost elements and we use the index rather than actual costs to record the lower bound L. The initial upper bound U is selected as the value returned by the upper bound algorithm and the objective function value "heu-obj" is selected as that returned by the Quadratic Threshold Heuristic. For example, $L = 87$ means the lower bound is z_{87} in the ordered list of distinct costs. The CPU time was measured in seconds corrected to three decimal digits. Also the number of times a lower bound was updated ("l-update") and the number of times an upper bound was updated ("u-update") are recorded. The column "shake" shows how many upper bound updates are caused by the shake operations. The "min," "max," and "average" of these values are calculated based on the 10 problems for each graph topology we considered.

To evaluate the quality of the lower bound, the initial upper bound and the heuristic solution, we define the relative bounds as $r_L = \frac{L}{p}\%$, $r_U = \frac{U}{p}\%$, and relative objective function value as $r_{\text{obj}} = \frac{\text{heu}-\text{obj}}{p}\%$, where p is the number of distinct costs. In each binary search iteration of the algorithm, we used tabu search together with shake operations to solve the MSTC on Step 16. The ratio $\frac{u-\text{update}}{\ell-\text{update}}$ provides us some insight into the effectiveness of our MSTC heuristic on Step 16. Further, the ratio $\frac{\text{shake}}{u-\text{update}}$ measures in percentage the effectiveness of shake operations. We calculated these ratios for the test problems by using the average values of L, U, heu-obj, u-update, ℓ-update and shake from Table 1 and the results are summarized in Table 2.

From Tables 1 and 2, it can be observed that, in general, our heuristics run in a reasonable amount of CPU time with acceptable solution quality. The lower bound L tends to get worse as the size of the problems get larger and it is affected significantly by the number of edge pairs. The quality of the initial upper bound U appears not very good, and it turns worse as n and m increases. Tabu search heuristic for

Table 1. General problems.

n	m	L			U			Heu-obj			CPU time			l-update			u-update			Shake		
		Min	Max	Av.	Min	Max	Av.	Min	Max	Av.	Min	Max	Av.	Min	Max	Av.	Min	Max	Av.	Min	Max	Av.
10	20	38	87	51.9	109	119	114.2	90	101	97.3	0.110	0.250	0.181	2	4	3.1	2	4	2.80	0	0	0
	30	25	40	34	238	261	253.4	173	210	193	0.297	0.844	0.517	3	6	4.5	2	5	3.3	0	1	0.3
	40	25	48	35	448	478	466.2	293	323	308.8	0.547	0.922	0.736	3	6	4.1	3	6	4.7	0	3	0.9
	45	22	68	33.8	597	621	606.1	366	406	391.2	0.625	1.156	0.838	3	6	4.6	3	6	4.5	0	2	0.6
30	60	599	946	711	1070	1129	1099.3	1023	1083	1056.4	3.094	5.859	4.505	4	6	5.2	1	5	3.2	0	1	0.7
	80	523	959	705.4	1918	2000	1966.6	1799	1859	1826.6	6.078	8.672	7.466	4	6	4.9	4	7	5.6	0	2	0.7
	100	525	1684	926.5	3031	3120	3063.8	2774	2843	2805.6	5.422	8.328	7.080	4	8	6	4	7	5.2	0	4	2.2
	200	456	660	554.3	12,004	12,154	12,092.6	9883	10,285	10,096.9	50.782	80.517	65.881	5	10	7.1	5	9	6.3	1	8	3.7
	300	420	1156	626.6	28,021	28,194	28,090.7	23,310	23,749	23,468.8	208.097	1254.110	645.713	5	11	8.3	3	10	6.3	2	6	3.9
	435	227	354	297.5	40,232	40,375	40,276.8	31,435	32,504	31,974.2	902.699	5270.080	2693.204	5	12	8.6	3	11	6.7	1	7	3.9
50	100	1723	2670	2055	3053	3137	3082	2987	3049	3011.5	18.407	37.125	29.600	4	9	6.1	1	6	3.8	0	4	1.7
	150	1940	4419	2519.1	6928	7061	7001.7	6614	6711	6669.8	74.735	121.064	97.867	4	10	8.6	2	9	3.4	1	8	2.6
	200	1859	4002	2646.7	12,030	12,180	12,131.2	11,173	11,440	11,335.4	126.033	185.675	143.282	4	9	7.7	4	9	5.4	2	5	3.6
	250	1490	5431	2430.1	19,258	19,479	19,378.6	17,863	18,073	17,988.2	224.347	385.755	290.841	6	12	9	2	8	4.9	0	6	3.1
	300	659	1152	884.6	10,954	10,993	10,975.6	10,072	10,302	10,173.5	306.926	1125.080	646.438	5	10	7.6	3	9	5.8	2	6	3.6
	450	1222	2565	1725.4	43,146	43,402	43,245.6	38,772	39,298	39,104.8	947.325	5614.320	2796.142	8	11	9.9	4	8	5.3	2	7	3.4
	500	1178	2553	1665.1	53,552	53,785	53,699.2	48,358	49,004	48,640.3	1197.140	9551.810	5052.977	4	11	8.4	4	12	7.2	3	6	4.3

Table 2. Relative bounds and obj on general problems.

n	m	$r_L(\%)$	$r_U(\%)$	$r_{\text{obj}}(\%)$	$\frac{u\text{−update}}{l\text{−update}}$	$\frac{\text{shake}}{u\text{−update}}\%$
10	20	43.3	95.2	81.1	0.9	0.0
10	30	12.5	93.2	71.0	0.7	9.1
10	40	7.2	95.5	63.3	1.1	19.1
10	45	5.4	96.7	62.4	1.0	13.3
30	60	63.8	98.7	94.8	0.6	21.9
30	80	35.5	98.8	91.8	1.1	12.5
30	100	29.8	98.6	90.3	0.9	42.3
30	200	4.5	98.8	82.5	0.9	58.7
30	300	2.2	99.2	82.9	0.8	61.9
30	435	0.7	99.3	78.8	0.8	58.2
50	100	66.1	99.2	96.9	0.6	44.7
50	150	35.7	99.2	94.5	0.4	76.5
50	200	21.6	99.1	92.6	0.7	66.7
50	250	12.5	99.5	92.4	0.5	63.3
50	300	8.0	99.4	92.1	0.8	62.1
50	450	4.0	99.5	90.0	0.5	64.2
50	500	3.1	99.5	90.2	0.9	59.7

MSTC combined with "shake" operations performed consistently and the ratio $\frac{u\text{−update}}{\ell\text{−update}}$ averaged around 0.8. The scheme worked relatively well for larger problems, since the quality of the heuristic objective function value appears to become better as the number of edge pairs is increased. From $\frac{\text{shake}}{u\text{−update}}\%$ it can be seen that the shake operation works fairly well and it contributed to more than 60% of all the u-updates for large problems.

The results on the special problems are summarized in Table 3. "opt-gap" represents the gap between the objective function value of the heuristic solution and that of the optimal solution.

It is interesting to note that our heuristic algorithm identified an optimal solution for all special problems generated with known optimal solution. This indicates the power of the tabu search heuristic in solving MSTC and the power of the "shake" operations in search diversifications. But it is possible that the special problems we constructed may be "too special" that they do not capture the real complexity of QBST. Nevertheless, the test results on these problems are very impressive.

6. LINKAGES WITH LBP

As mentioned in the introduction section, a QBP can be viewed as a linear bottleneck problem with a higher dimensional ground set and an altered family of feasible solutions. This relationship can sometimes be exploited to derive corresponding results for QBP. We illustrate this with two examples.

Note that any linear bottleneck problem can formulated as a minsum problem using exponentially large costs [23]. As an analogous result, we now observe that any QBP can be formulated as a quadratic minsum problem (QSP). For each feasible solution $S \in \mathfrak{F}$, let $g(S) = \max\{f(\{x,y\}) : \{x,y\} \in P[S]\}$. Let $z^1 < z^2 < \cdots < z^p$ be an ascending arrangement of distinct costs from $\{f(\{x,y\}) : \{x,y\} \in P[E]\}$ and $F^r = \{S \in \mathfrak{F} : g(S) = z^r\}$ for $r = 1, 2, \ldots, p$. Also, let $U_r = \bigcup_{i=1}^r F^i = \{S \in \mathfrak{F} : g(S) \le z^r\}$ for $r = 1, \ldots, p$.

Let h be another real valued cost function defined on $P[E]$.

THEOREM 12: If $h(\{x,y\})$ satisfies

$$\min\left\{\sum_{\{x,y\}\in P[S]} h(\{x,y\}) : S \in F^r\right\} > \min\left\{\sum_{\{x,y\}\in P[S]} h(\{x,y\}) : S \in U_{r-1}\right\}$$

for $2 \le r \le p$, then every optimal solution to QSP with cost function h is also an optimal solution to QBP with cost function f.

PROOF: If k is the smallest index such that F^k is nonempty, then it is obvious that any $S \in F^k$ is an optimal solution to QBP with the optimal objective function value z^k. Now assume that S' is an optimal solution to QSP with cost function h and $S' \in F^q$ for some q, then we have

Table 3. Special problems.

n	m	Opt-gap (%)	CPU time Min	Max	Av.
10	20	0	0.031	0.156	0.067
10	30	0	0.015	0.172	0.040
10	40	0	0.031	0.203	0.072
10	45	0	0.046	0.250	0.102
30	60	0	0.078	0.125	0.102
30	80	0	0.188	0.797	0.295
30	100	0	0.391	1.016	0.478
30	200	0	5.234	6.328	5.634
30	300	0	25.233	28.311	26.835
30	435	0	109.813	119.388	112.391
50	100	0	0.390	0.437	0.409
50	150	0	1.719	2.078	1.825
50	200	0	5.125	6.016	5.375
50	250	0	12.235	13.453	12.505
50	300	0	25.328	31.891	27.389
50	450	0	125.892	130.220	127.395
50	500	0	191.940	214.393	198.801

$$\sum_{\{x,y\}\in P[S']} h(\{x,y\}) = \min\left\{\sum_{\{x,y\}\in P[S]} h(\{x,y\}) : S \in F^q\right\}$$
$$> \min\left\{\sum_{\{x,y\}\in P[S]} h(\{x,y\}) : S \in U_{q-1}\right\}.$$

Thus U_{q-1} must be empty and therefore S' is an optimal solution to QBP. □

The cost function h satisfying the condition of the above theorem can be constructed easily by modifying the results for the bottleneck traveling salesman problem discussed in [23] and we omit details. By Theorem 12, QBP can be solved by solving a single QSP. However, to satisfy the condition of the theorem, $h(\{x,y\})$ often grows exponentially in the problem size m, and hence it is difficult to use the theorem directly with computational advantage.

Before concluding this section, let us consider an interesting asymptotic theorem of Albrecher [1] proved for bottleneck problems. We give below the adaptation of this theorem for QBP and the proof follows directly from [1] in view of the transformation discussed in Section 1 where a QBP can be viewed as an LBP with a higher dimensional ground set. Burkard and Fincke [4] obtained the first such asymptotic result for the quadratic bottleneck assignment problem which later was improved by Albrecher [1]. Another interesting work in this direction is by Szpankowski [37].

THEOREM 13: Let $I_n = (E^n, F^n, f^n)$ be an instance of a QBP such that $S \in F^n$ implies $|S| = \alpha_n$ for $n \in \aleph$. Let $f^n(\{x,y\})$ for all $\{x,y\} \in P[E^n]$ be identically distributed random variables in $[0, M]$ and $f^n(\{x,y\})$ for $\{x,y\} \in P[S]$ be independent for every fixed feasible solution $S \in F^n, n \in \aleph$. Let $\gamma(n)$ be a positive function such that

$$\ln|S| + \alpha_n^2 \ln \mathbb{P} \times \left(f^n(\{x,y\}) \le \frac{M}{1+\gamma(n)}\right) \text{ diverges to } -\infty \text{ as } n \to \infty. \tag{3}$$

Then,

$$\mathbb{P}\left(\frac{g^n(S^w) - g^n(S^*)}{g^n(S^*)} \le \gamma(n)\right) = 1 - o(1), \tag{4}$$

where S^* is an optimal solution, S^w is the worst solution for the instance I_n, and $g^n(S) = \max\{f^n(\{x,y\}) : \{x,y\} \in P[S]\}$. Furthermore, if the series

$$\sum_{i=1}^{\infty} |F^n| \left(\mathbb{P}(f^n(\{x,y\}) \le \frac{M}{1+\gamma(n)})\right)^{\alpha_n^2} \tag{5}$$

converges then $\frac{g^n(S^w)-g^(S^*)}{g^n(S^*)} \le \gamma(n)$ holds almost surely.

Theorem 13 can be used to obtain specialized results for various special cases of QBP. This is illustrated for the case of QBST below.

Consider an instance $ST(n)$ of the QBST defined on the graph G^n with cost function f^n where $f^n(\{x,y\})$s are uniformly distributed random numbers in the interval [0, 1]. Let T^* be an optimal solution to $ST(n)$ and T^w is a worst solution. Specializing Theorem 13 to QBST, we get the following bound.

THEOREM 14: $\frac{g^n(T^w)}{g^n(T^*)} \le n^{\frac{n}{(n-1)^2}}$ almost surely. Equivalently, $\mathbb{P}\left(\frac{g^n(T^w)}{g^n(T^*)} > n^{\frac{n}{(n-1)^2}} \text{ infinitely often}\right)$ is zero.

PROOF: Let F^n be the family of all spanning trees of G^n on n nodes. By Cayley's theorem [7] $|F^n| \le n^{n-2}$. Each spanning tree contains $n-1$ arcs. Since $f^n(\{x,y\})$s are uniformly and independently distributed random numbers in the interval [0, 1], $\mathbb{P}(f^n(\{x,y\}) \le \kappa) = \kappa$. Note that

$$\sum_{n=1}^{\infty} |F^n| \left(\mathbb{P}(f^n(\{x,y\}) \le \frac{1}{n^{\frac{n}{(n-1)^2}}})\right)^{(n-1)^2} \le \sum_{n=1}^{\infty} n^{n-2}\frac{1}{n^n} \tag{6}$$

$$= \sum_{n=1}^{\infty} \frac{1}{n^2} \text{ which coverges to } \frac{\pi}{6}. \tag{7}$$

The result now follows from Theorem 13 by choosing $\gamma(n) = n^{\frac{n}{(n-1)^2}} - 1$, $\alpha_n = n-1$ and $M = 1$. □

7. CONCLUSION

In this article, we considered the general quadratic bottleneck problem (QBP) and developed an algorithmic framework that can be used to devise exact and heuristic algorithms for the problem. The problem is shown to be NP-hard even for very simple structure of the family of feasible solutions. A weak duality theorem is introduced and special polynomially solvable cases are identified. We then studied the special case of QBST form a computational point of view. Heuristic algorithms are provided and results of computational experiments are reported illustrating the efficacy of our algorithms. Finally, it is shown that QBP can be solved a quadratic minsum problem and some asymptotic results discussed. We are currently investigating various special cases of QBP from an algorithmic point of view.

ACKNOWLEDGMENTS

This work was supported by an NSERC discovery grant awarded to Abraham Punnen. We are thankful to the referees for their comments which improved the presentation of the article. A preliminary version of this article was presented at the annual CORS meeting in London, Ontario in 2007. This work was supported by an NSERC discovery grant awarded to Abraham P. Punnen.

REFERENCES

[1] H. Albrecher, A note on the asymptotic behaviour of bottleneck problems, Oper Res Lett 33 (2005), 183–186.
[2] A. Assad and W. Xu, The quadratic minimum spanning tree problem, Naval Res Logist 39 (1992), 399–417.
[3] R.E. Burkard, Quadratische bottleneckprobleme, Oper Res Verfahren 18 (1974), 26–41.
[4] R.E. Burkard and U. Fincke, On random quadratic bottleneck assignment problems, Math Program 23 (1982), 227–232.
[5] P.M. Camerini, The min-max spanning tree problem and some extensions, Inform Process Lett 7 (1978), 10–14.
[6] P.M. Camerini, F. Maffioli, S. Martello, and P. Toth, Most and least uniform spanning trees, Discrete Appl Math 15 (1986), 181–197.
[7] A. Cayley, A theorem on trees, Quart J Math 23 (1889), 376–378.
[8] A. Darmann, U. Pferschy, and J. Schauer, Minimal spanning trees with conflict graphs, Optim Online (2009).
[9] A. Darmann, U. Pferschy, J. Schauer, and G.J. Woeginger, Path, trees and matchings under disjunctive constraints, Optim Online (2009).
[10] U. Derigs, On three basic methods for solving bottleneck transportation problems, Naval Res Logist Quart 29 (1982), 505–515.
[11] C.W. Duin and A. Volgenant, Minimum deviation and balanced optimization: A unified approach, Oper Res Lett 10 (1991), 43–48.
[12] J. Edmonds and D.R. Fulkerson, Bottleneck extrema, J Combin Theory 8 (1970), 299–306.
[13] E. Fernandez, R. Garfinkel, and R. Arbiol, Mosaicking of aerial photographic maps via seams defined by bottleneck shortest paths, Oper Res 46 (1998), 293–304.
[14] E.Y. Gabovich, Constant discrete programming problems on subsititution sets, Kibernetika 5 (1976), 128–134, (in Russian).
[15] H.N. Gabow and R.E. Tarjan, Algorithms for two bottleneck optimization problems, J Algorithms 9 (1988), 411–417.
[16] Z. Galil and B. Schieber, On finding most uniform spanning trees, Discrete Appl Math 20 (1988), 173–175.
[17] G. Gallo, P.L. Hammer, and B. Simeone, Quadratic knapsack problems, Math Program Study 12 (1980), 132–149.
[18] M.R. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness, W. H. Freeman & Co., New York, NY, USA, 1979.
[19] R.S. Garfinkel and M.R. Rao, The bottleneck transportation problem, Naval Res Logist Quarterly 18 (1971), 465–472.
[20] L. Georgiadis, Arborescence optimization problems solvable by Edmonds' algorithm, Theoret Comput Sci 301 (2003), 427–437.
[21] S.K. Gupta and A.P. Punnen, k-sum optimization problems, Oper Res Lett 1990 (9), 121–126.
[22] S.N. Kabadi, The polynomially solvable cases of the TSP. The traveling salesman problem and its variations, G. Gutin and A.P. Punnen (Editors), Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002, pp. 489–493.
[23] S.N. Kabadi and A.P. Punnen, The bottleneck TSP. The traveling salesman problem and its variations, G. Gutin and A.P. Punnen (Editors), Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002, pp. 697–734.
[24] S.N. Kabadi and A.P. Punnen, Weighted graphs with all Hamiltonian cycles of the same length, Discrete Math 271 (2003), 129–139.
[25] H. Katagiri, M. Sakawa, and H. Ishii, Fuzzy random bottleneck spanning tree problems using possibility and necessity measures, Eur J Oper Res 152 (2004), 88–95.
[26] H. Kellerer and G. Wirsching, Bottleneck quadratic assignment problems and the bandwidth problem, Asia-Pacific J Oper Res 15 (1998), 169–177.
[27] J. Krarup and P.M. Pruzan, Reducibility of minimax to minisum 0-1 programming problems, Eur J Oper Res 6 (1981), 125–132.
[28] S.V. Listrovio and V.I. Khrin, Parallel algorithm to find maximum capacity paths, Cybernet Syst Anal 34 (1998), 261–268.
[29] S. Martello, W. Pulleyblank, P. Toth, and D. de Werra, Balanced optimization problems, Oper Res Lett 3 (1984), 275–278.
[30] U. Pferschy, Solution methods and computational investigations for the linear bottleneck assignment problem, Computing 59 (1997), 237–258.
[31] A.P. Punnen and K.P.K. Nair, Improved complexity bound for the maximum cardinality bottleneck bipartite matching problem, Discrete Appl Math 55 (1994), 91–93.
[32] A.P. Punnen, A fast algorithm for a class of bottleneck problems, Computing 56 (1996), 397–401.
[33] A.P. Punnen and Y.P. Aneja, On k-sum optimization, Oper Res Lett 18 (1996), 233–236.
[34] A.P. Punnen and K.P.K. Nair, An improved algorithm for the constrained bottleneck spanning tree problem, INFORMS J Comput 8 (1996), 41–44.
[35] A.P. Punnen and K.P.K. Nair, An $O(m \log n)$ algorithm for the max + sum spanning tree problem, Eur J Oper Res 89 (1996), 423–426.
[36] L. Steinberg, The backboard wiring problem: A placement algorithm, SIAM Rev 3 (1961), 37–50.
[37] W. Szpankowski, Combinatorial optimization problems for which almost every algorithm is asymptotically optimal! Optimization 33 (1995), 359–367.
[38] V. Vassilevska, R. Williams, and R. Yuster, All-pairs bottleneck paths for general graphs in truly sub-cubic time, Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing, New York, NY, USA, 2007, pp. 585–589.
[39] R. Zhang, S.N. Kabadi, and A.P. Punnen, The minimum spanning tree problem with conflict constraints and its variations, Discrete Optim doi:10.1016/j.disopt.2010.08.001.
[40] G. Zhou and M. Gen, Genetic algorithm approach on multicriteria minimum spanning tree problem, Eur J Oper Res 114 (1999), 141–152.