

## On the Quadratic Shortest Path Problem

Borzou Rostami, Federico Malucelli, Davide Frey, Christoph Buchheim

► **To cite this version:**

Borzou Rostami, Federico Malucelli, Davide Frey, Christoph Buchheim. On the Quadratic Shortest Path Problem. 14th International Symposium on Experimental Algorithms, Jun 2015, Paris, France. 14th International Symposium on Experimental Algorithms, 2015, 14th International Symposium on Experimental Algorithms. <10.1007/978-3-319-20086-6\_29>. <hal-01251438>

**HAL Id: hal-01251438**

**<https://hal.inria.fr/hal-01251438>**

Submitted on 6 Jan 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# On the Quadratic Shortest Path Problem

Borzou Rostami<sup>1</sup>, Federico Malucelli<sup>2</sup>, Davide Frey<sup>3</sup>, and  
Christoph Buchheim<sup>1</sup>

<sup>1</sup> Fakultät für Mathematik, TU Dortmund, Germany

<sup>2</sup> Department of Electronics, Information, and Bioengineering, Politecnico di Milano,  
Milan, Italy

<sup>3</sup> INRIA-Rennes Bretagne Atlantique, Rennes, France

**Abstract.** Finding the shortest path in a directed graph is one of the most important combinatorial optimization problems, having applications in a wide range of fields. In its basic version, however, the problem fails to represent situations in which the value of the objective function is determined not only by the choice of each single arc, but also by the combined presence of pairs of arcs in the solution. In this paper we model these situations as a Quadratic Shortest Path Problem, which calls for the minimization of a quadratic objective function subject to shortest-path constraints. We prove strong NP-hardness of the problem and analyze polynomially solvable special cases, obtained by restricting the distance of arc pairs in the graph that appear jointly in a quadratic monomial of the objective function. Based on this special case and problem structure, we devise fast lower bounding procedures for the general problem and show computationally that they clearly outperform other approaches proposed in the literature in terms of its strength.

**Keywords:** Shortest Path Problem, Quadratic 0–1 optimization, Lower bounds

## 1 Introduction

The Shortest Path Problem (SPP) is among the best studied combinatorial optimization problems on graphs. It arises frequently in practice in a variety of settings and often appears as a subproblem in algorithms for other combinatorial optimization problems. In a directed network with arbitrary given lengths, the SPP is the problem of finding a directed path from an origin node  $s$  to a target node  $t$  with shortest total length. Many classical algorithms such as Dijkstra’s labeling algorithm [7] and Bellman-Ford’s successive approximation algorithm [2] have been developed to solve the problem.

The basic SPP fails to model situations in which the value of a linear objective function is not the only interesting parameter in the choice of the optimal solution. Such problems include situations in which the choice of the shortest path is constrained by parameters such as the variance of the cost of the path, or cases in which the objective function takes into account not only the cost

of each selected arc but also the cost of the interactions among the arcs in the solution. We call such a problem Quadratic Shortest Path Problem (QSPP).

The first variant of the SPP studied in the literature that is directly related to QSPP is probably that of Variance Constrained Shortest Path [13]. The problem seeks to locate the path with the minimum expected cost subject to the constraint that the variance of the cost is less than a specified threshold. The problem arises for example in the transportation of hazardous materials. In such cases a path must be short but it must also be subject to a constraint that the variance of the risk associated with the route is less than a specified threshold. More generally, this problem may arise in all situations in which the costs associated with each arc consist of stochastic variables. Possible approaches to solving the Variance Constrained Shortest Path problem involve a relaxation in which the quadratic variance constraint is incorporated into the objective function, thus yielding a QSPP problem. In this case, the quadratic part of the objective function is determined by the covariance matrix of the coefficient's probability distributions. In [12] the authors develop a multi-objective model to minimize both the expected travel time of a path and its variance. Then they solve the multi-objective optimization problem by combining the linear and quadratic objective functions into a single quadratic shortest path problem.

A different type of applications arises from research on network protocols. In [10], the authors study different restoration schemes for self-healing ATM networks. In particular, the authors examine line and end-to-end restoration schemes. In the former, link failures are addressed by routing traffic around the failed link, in the latter, instead, traffic is rerouted by computing an alternative path between source and target. Within their analysis, the authors point out the need to solve a QSPP to address rerouting in the latter scheme. Nevertheless, they do not provide details about the algorithm used to obtain a QSPP solution.

Recently, Amaldi et al. [1] introduced new combinatorial optimization problems called *reload cost paths, tours, and flows* which have several applications in transportation networks, energy distribution networks, and telecommunication networks. In the reload cost problems, one is given a graph whose every edge is assigned a color and there is a reload cost when passing through a node on two edges that have different colors. Therefore, the reload cost path problem is a special case of the QSPP in which the objective function takes into account only the reload cost of consecutive arcs with different colors. The authors proved that the reload cost path problem is polynomially solvable.

All problems described above involve variants of the shortest-path problem in which the cost associated with each arc is integrated by a contribution associated with the presence of pairs of arcs in the solution. Such a contribution can be expressed by a quadratic objective function on binary variables associated with each arc, and leads to the definition of a QSPP. To best of our knowledge, there is no previous research dealing directly with solution methods nor complexity studies of the QSPP. Buchheim and Traversi [4] proposed a generic framework for solving binary quadratic programming problems by computing quadratic global underestimators of the objective function that are separable but not necessarily

convex. In their computational experiments, they solve some special classes of quadratic 0 – 1 problems including the QSPP.

In this paper we analyze the complexity of the QSPP and study different special cases of the problem which can be solved in polynomial time. We then develop efficient lower bounding schemes which build a classical SPP or a new special QSPP from the original problem in order to obtain lower bounds. It turns out that the new bounds outperform all lower bounding schemes proposed in the literature so far [4].

## 2 Problem formulation and complexity

Given a directed graph  $G(V, A)$ , a source node  $s \in V$ , a target node  $t \in V$ , a cost function  $c : A \rightarrow \mathbb{R}^+$ , which maps every arc to a non-negative cost, and a cost function  $q : A \times A \rightarrow \mathbb{R}^+$  that maps every pair of arcs to a non-negative real cost, we denote by  $\delta^-(i) = \{j \in V \mid (j, i) \in A\}$  and  $\delta^+(i) = \{j \in V \mid (i, j) \in A\}$  the set of predecessor and successor nodes for any given  $i \in V$ . Defining a binary variable  $x_{ij}$  indicating the presence of arc  $(i, j)$  on the optimal path, the QSPP is represented as:

$$\begin{aligned} \text{QSPP: } z^* = \min & \sum_{(i,j),(k,l) \in A} q_{ijkl} x_{ij} x_{kl} + \sum_{(i,j) \in A} c_{ij} x_{ij} \\ \text{s.t. } & x \in X_{st}, x \text{ binary.} \end{aligned} \quad (1)$$

Here the feasible region,  $X_{st}$ , is exactly the same as that associated with the standard shortest-path problem, i.e.,

$$X_{st} = \left\{ 0 \leq x \leq 1 : \sum_{j \in \delta^+(i)} x_{ij} + \sum_{j \in \delta^-(i)} x_{ji} = b(i) \quad \forall i \in V \right\}.$$

Note that  $b(i) = 1$  for  $i = s$ ,  $b(i) = -1$  for  $i = t$ , and  $b(i) = 0$  for  $i \in V \setminus \{s, t\}$ .

**Theorem 1.** *QSPP is strongly NP-hard.*

*Proof.* Let us consider the general form of the Quadratic Assignment Problem (QAP) on a complete bipartite graph  $G = (U, V, E)$  with nodes  $U \cup V$ , undirected arcs  $E$ , a linear cost  $c$ , and a quadratic cost  $q$ . We may assume that nodes in  $U$  and  $V$  are both numbered  $1, \dots, m$ . We show that this generic instance of the QAP can be reduced to a corresponding instance of QSPP in polynomial time. To this end, we define an QSPP instance on a graph  $\tilde{G} = (\tilde{V}, \tilde{A})$  and map each feasible QAP assignment onto a feasible path in  $\tilde{G}$ , where  $\tilde{V}$  and  $\tilde{A}$  are defined as follows:

$$\tilde{V} = (U \times V) \cup \{s, t\}, \quad \text{and} \quad \tilde{A} = A_s \cup A_+ \cup A_t,$$

where

$$\begin{aligned} A_s &= \{(s, (1, i)) : i \in V\}, \quad A_t = \{((m, i), t) : i \in V\}, \quad \text{and} \\ A_+ &= \{((i, j), (i+1, k)) : i \in U \setminus \{m\}, j, k \in V, j \neq k\}. \end{aligned}$$

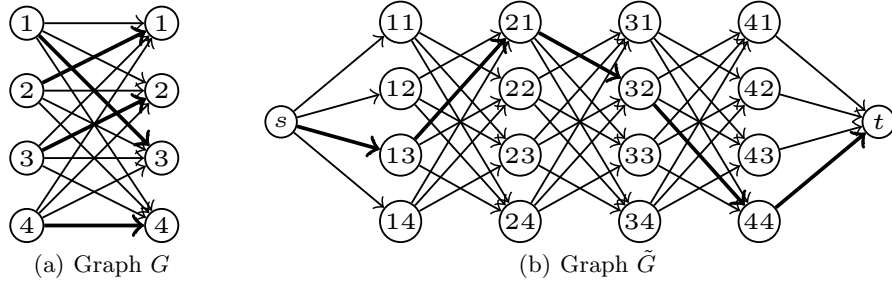


Fig. 1: Graph  $G$  and  $\tilde{G}$ . Bold lines in Graph  $G$  and  $\tilde{G}$  illustrate a feasible assignment for the QAP and its corresponding unique feasible path for QSPP, respectively.

Each node  $(i, j) \in U \times V$  corresponds to an edge in the original QAP instance, we will use the notation  $u((i, j)) := i$  and  $v((i, j)) := j$  in the following.

Figure 1 shows the graphs  $G$  and  $\tilde{G}$  with  $m = 4$ . With reference to this figure,  $u((i, j))$  represents the column of node  $(i, j)$  when the graph  $\tilde{G}$  is arranged on a grid as shown. Moreover, it represents the index of the first of the two QAP nodes corresponding to  $(i, j)$  in the bipartite graph on the left. Analogously,  $v((i, j))$  represents the row in the grid and the index of the second QAP node in the bipartite graph.

The graph structure resulting from the above transformation has a number of nodes equal to  $m^2 + 2$  and a number of arcs equal to  $m^3 - 2m^2 + 3m$ , which makes the reduction polynomial.

Moreover, this construction maps each feasible assignment  $\pi: U \rightarrow V$  in  $G$  to a unique feasible path in  $\tilde{G}$  as follows: the first arc of the path is  $(s, (1, \pi(1)))$ , the next arcs are  $((i, \pi(i)), (i+1, \pi(i+1)))$  for  $i = 1, \dots, m-1$ , and the final arc is  $((m, \pi(m)), t)$ . By construction and since  $\pi(i) \neq \pi(i+1)$  for all  $i = 1, \dots, m-1$ , all arcs in this path exist in  $\tilde{G}$ . Vice versa, every path in  $\tilde{G}$  uniquely determines a function  $\pi: U \rightarrow V$  by setting  $\pi(u(w)) = v(w)$  for all  $w \in U \times V$  belonging to the path. However, this function is not necessarily a feasible QAP assignment, as different nodes of  $U$  may be mapped to the same node of  $V$ . This problem is easily addressed by appropriately generating the cost matrix as we show next.

The linear cost vector is defined in Equation (2). The cost for any arc pointing to node  $e$  is given by the cost of the arc from  $u(e)$  to  $v(e)$  in the QAP.

$$\tilde{c}_{fe} = \begin{cases} c_{u(e)v(e)} & e \neq t \\ 0 & e = t. \end{cases} \quad (2)$$

The assignment of quadratic costs to pairs of arcs in  $\tilde{G}$  is defined according to Equation (3). In general, the cost  $\tilde{q}_{f_e h_w}$  corresponding to the pair  $(f, e), (h, w) \in \tilde{A}$  is equal to the cost  $q_{u(e)v(e)u(w)v(w)}$  in the original problem. However, Equation (3) includes an additional constraint to prevent the creation of paths corre-

sponding to infeasible QAP solutions, where two distinct nodes in  $U$  are assigned to the same node in  $V$ .

$$\tilde{q}_{f_{ehw}} = \begin{cases} q_{u(e)v(e)u(w)v(w)} & e \neq t \wedge w \neq t \wedge v(e) \neq v(w) \\ 0 & e = t \vee w = t \\ \infty & \text{otherwise.} \end{cases} \quad (3)$$

The last case in Equation (3) thus makes sure that any optimal solution of QSPP in graph  $\tilde{G}$  defines a feasible assignment  $\pi$  in graph  $G$ , so that there is a one-to-one correspondence between the feasible assignments in  $G$  and the directed paths in  $\tilde{G}$  with finite weight, as explained above. It is easy to verify that by construction the cost remains the same under this transformation.

As the QAP problem is strongly NP-hard [11] and the numbers defined in the transformation all have polynomial values (infinite costs can be replaced by an appropriate polynomial value  $M$ ), the result follows.  $\square$

### 3 The adjacent quadratic shortest path problem

In this section, we consider special cases of the QSPP where the quadratic part of the cost function has a *local* structure, meaning that each pair of variables appearing jointly in a quadratic term in the objective function corresponds to a pair of arcs lying close to each other. We start with the Adjacent QSPP (AQSP), where interaction costs of all non-adjacent pair of arcs are assumed to be zero. Therefore, only the quadratic terms of the form  $x_{ij}x_{kl}$  with  $j = k$  and  $i \neq l$  or with  $j \neq k$  and  $i = l$  have nonzero objective function coefficients. The AQSP can be viewed as a generalization of the Reload Cost path introduced by Amaldi et al. [1].

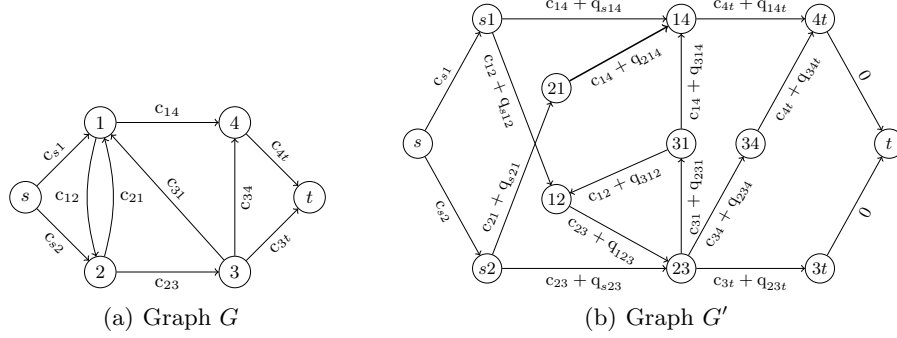
In order to solve the AQSP, we propose a polynomial-time algorithm based on a transformation that reduces the original problem on graph  $G = (V, A)$  to the classical shortest path problem in an auxiliary directed graph  $G' = (V', A')$ . For this, we may assume w.l.o.g. that there is no direct arc from  $s$  to  $t$  in  $G$ . Now define

$$\begin{aligned} V' &= \{\langle s, s \rangle\} \cup \{\langle i, j \rangle : (i, j) \in A\} \cup \{\langle t, t \rangle\}, \\ A' &= \{(\langle i, j \rangle, \langle j, k \rangle) : \langle i, j \rangle, \langle j, k \rangle \in V'\}, \end{aligned}$$

where  $\langle s, s \rangle$  and  $\langle t, t \rangle$  represent nodes  $s$  and  $t$ , respectively, while all the other nodes in  $G'$  correspond to the arcs in the original graph  $G$ . Next, we associate each arc  $(\langle i, j \rangle, \langle j, k \rangle) \in A'$  with a weight  $w$  defined as:

$$w(i, j, k) = \begin{cases} c_{jk} + q_{ijjk} & \langle i, j \rangle \neq \langle s, s \rangle \wedge \langle j, k \rangle \neq \langle t, t \rangle \\ c_{jk} & \langle i, j \rangle = \langle s, s \rangle \\ 0 & \langle j, k \rangle = \langle t, t \rangle \end{cases}$$

Since  $G'$  contains  $|A| + 2$  nodes and  $\delta^+(s) + \delta^-(t) + \sum_{i \neq s, t} (\delta^-(i)\delta^+(i))$  arcs, it can be constructed in polynomial time. In Figure 2 we present an example of a graph  $G$  and the corresponding auxiliary graph  $G'$ .

Fig. 2: Graph  $G = (V, A)$  and its auxiliary graph  $G' = (V', A')$ .

Let  $c(P) = \sum_{(i,j) \in P} c_{ij} + \sum_{(i,j),(j,k) \in P} q_{ijjk}$  be the cost of any  $s - t$  path  $P$  in  $G$ , and  $w(P') = \sum_{e \in P'} w_e$  be the cost of any  $\langle s, s \rangle - \langle t, t \rangle$  path  $P'$  in  $G'$ . The following lemma is a straightforward result implied by the construction of  $G'$ .

**Lemma 1.** *For any  $s - t$  path  $P$  in  $G$  there exists an  $\langle s, s \rangle - \langle t, t \rangle$  path  $P'$  in  $G'$  with  $c(P) = w(P')$ , and vice versa.*

*Proof.* For a given  $s - t$  path  $P \subseteq A$  in  $G$ , the path  $P' \subseteq A'$  is defined as follows: an arc  $\langle (i, j), (j, k) \rangle$  belongs to  $P'$  if and only if  $(i, j), (j, k) \in P \cup \{(s, s), (t, t)\}$ . The path  $P$  can be computed from  $P'$  accordingly.  $\square$

This immediately implies the following

**Theorem 2.** *An optimal solution for AQSP in graph  $G$  can be obtained by solving a classical shortest path over  $G'$ .*

**Corollary 1.** *For any given source node  $s$  and target node  $t$ , the AQSP on graph  $G$  can be solved in  $O(\min\{|A|^2, |V|^3\} + |A| \log |A|)$  time.*

*Proof.* Using Dijkstra's algorithm, the running time is  $O(|A'| + |V'| \log |V'|)$ , where  $|A'|$  can be both restricted by  $|A|^2$ , as each edge in  $G'$  corresponds to a pair of edges in  $G$ , and by  $|V|^3$ , as it is defined by three nodes in  $G$ .  $\square$

If the vertex degrees in  $G$  are bounded by  $\Delta$ , a bound of  $O(\Delta^2 |V| + |A| \log |A|)$  on the running time can be obtained.

These results hold for the case of a fixed source  $s$  and target  $t$ . Let us now consider the single-source AQSP which finds the minimum AQSP from a given source  $s$  to each vertex  $v \in V$ . To solve the problem we again consider the graph  $G'$ , but since  $t$  is not specified, we do not add node  $\langle t, t \rangle$ , nodes  $\langle k, t \rangle \forall k$ , and the arcs incident to these nodes. Then we use Dijkstra's algorithm to find the shortest path  $P_{\langle s, s \rangle \langle i, j \rangle}^*$  from the source node  $\langle s, s \rangle$  to all the other nodes  $\langle i, j \rangle$  of  $G'$ . For any target node  $t \in V$ , the solution of AQSP can then be obtained by computing

$$\min\{w(P_{\langle s, s \rangle \langle i, t \rangle}^*) : \langle i, t \rangle \in A'\}. \quad (4)$$

The total running time for solving the single-source AQSP is thus again given by  $O((\min\{|A|^2, |V|^3\} + |A| \log |A|))$ , since the additional total running time needed to solve (4) for all  $t \in V$  is  $O(|A'|)$  and thus dominated by the running time of the first phase.

Motivated by the results of Theorem 2, we can generalize the Adjacent QSP to an  $r$ -Adjacent QSP by defining the concept of  $r$ -adjacency.

**Definition 1.** *Given a fixed positive integer  $r$ , the graph  $G = (V, A)$  and two arcs  $(i, j)$  and  $(k, l)$  in  $A$ , we say that  $(i, j)$  and  $(k, l)$  are  $r$ -adjacent in  $G$  if there exists a directed path of length at most  $r$  containing both arcs.*

We can now define the  $r$ -Adjacent QSP ( $r$ -AQSP) as a more general case of the AQSP where objective function coefficients of the quadratic terms  $x_{ij}x_{kl}$  of non- $r$ -adjacent arcs  $(i, j), (k, l) \in A$  are assumed to be zero. With this definition, the AQSP agrees with the 2-Adjacent QSP.

Therefore, for any fixed positive integer number  $r \geq 2$ , we can apply the aforementioned graph construction to transform an  $r$ -AQSP to an  $(r - 1)$ -AQSP, where the 1-AQSP is equivalent to the classical shortest path problem. For fixed  $r$ , this leads to a polynomial time algorithm for the  $r$ -AQSP. However, the running time increases exponentially with  $r$ . Clearly, for large enough  $r$ , the  $r$ -AQSP agrees with the general QSP and is thus NP-hard by Theorem 1.

## 4 Lower bounding schemes

In this section, we propose lower bounding schemes for the general case of QSP based on a simple observation on the structure of the problem combined with the polynomial solvability of the AQSP. The methods are based on the Gilmore-Lawler (GL) procedure. The GL procedure is one of the most popular approaches to find a lower bound for the QAP proposed by Gilmore [8] and Lawler [9] and has been adapted to many other quadratic 0-1 problems in the meantime [5].

For each arc  $e = (i, j) \in A$ , potentially in the solution, we consider the minimum interaction cost of  $e$  in a path from  $s$  to  $t$ . In other words, we compute the shortest among the paths from  $s$  to  $t$  which contain arc  $e$ , using the  $ij$ -th column of the quadratic cost matrix as the cost vector. Let  $P_e$  be such a subproblem for a given arc  $e \in A$ :

$$P_e : \quad z_e = \min \left\{ \sum_{f \in A} q_{ef} x_f : x \in X_{st}, x_e = 1 \right\} \quad \forall e \in A. \quad (5)$$

The value  $z_e$  is the best quadratic contribution to the QSP objective function where arc  $e$  is in the solution. One possible way to solve  $P_e$  is to consider it as a minimum cost flow problem with two origins  $s$  and  $j$  and two destinations  $i$  and  $t$  in a network without arc  $e$ . Thus a solution to  $P_e$  can be found by solving a minimum-cost-flow problem with two units of cost to be transferred between two sources  $s$  and  $j$  and two destinations  $i$  and  $t$  in a graph  $\bar{G} = (V, A \setminus \{e\})$ . However, this represents a relaxation of  $P_e$ : in particular, it admits solutions



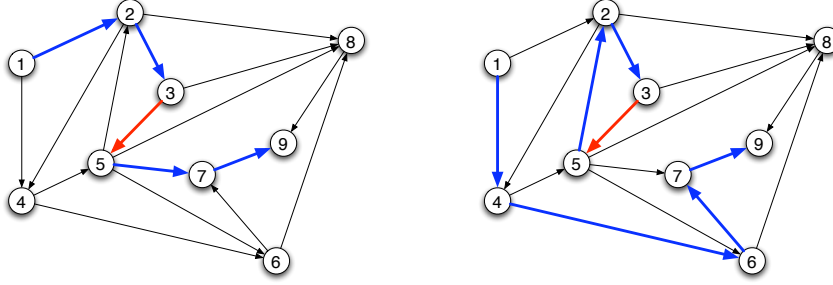


Fig. 3: Possible solutions to  $P_e$  when  $e = (3, 5)$ ,  $s = 1$  and  $t = 9$ .

that consist of the union of a path from  $s$  to  $t$  that does not contain arc  $e$  and a cycle containing  $e$ . The resulting solution will then have either of the two forms depicted in Figure 3.

To avoid the situations presented in Figure 3, one can modify the shortest path algorithms to include any given fixed arc  $e = (i, j)$ . The main idea is to compute the shortest path from  $s$  to  $i$ , add arc  $e$  to the path, and compute the shortest path from  $j$  to  $t$ . In addition we set to infinity the weights of all the arcs incident to  $t$  when computing the path from  $s$  to  $i$ . This prevents node  $t$  from being included in this path.

Once  $z_e$  has been computed for each  $e \in A$ , the GL bound is given by the solution to the following shortest path problem:

$$LB_{GLT} = \min \left\{ \sum_{e \in A} (z_e + c_e) x_e : x \in X_{st} \right\}.$$

The popularity of the GL approach for computing lower bounds stems from its low computational cost. However, for some quadratic 0–1 problems the obtained bounds deteriorate quickly as the size of the problem increases [6]. In the following subsections we propose two novel approaches to improve the GL lower bound for the QSPP.

#### 4.1 A generalized Gilmore-Lawler type bound

We consider a generalization of the GL (GGL) procedure which considers the minimum interaction cost not only of one arc but of two consecutive arcs. More precisely, for each two consecutive arcs  $e = (i, j), f = (j, k) \in A$ , potentially in the solution, we consider a subproblem  $P_{ef}$  to compute the shortest among the paths from  $s$  to  $t$  which contains these two arcs, i.e.,

$$P_{ef} : z_{ef} = \min \left\{ \sum_{h \in A} \hat{q}_{ef}^h x_h : x \in X_{st}, x_e = x_f = 1 \right\} \quad \forall e, f \in S_{2A},$$

where  $S_{2A}$  is the set of all 2-adjacent arcs in  $G$ , and  $\hat{q}$  is defined as follows:

$$\hat{q}_{ef}^h = \begin{cases} \frac{1}{2}(q_{eh} + q_{fh}) & i \neq s, k \neq t \\ q_{eh} + \frac{1}{2}q_{fh} & i = s, k \neq t \\ \frac{1}{2}q_{eh} + q_{fh} & i \neq s, k = t. \end{cases}$$

Similar to problem  $P_e$ , the solution to  $P_{ef}$  can be easily found by either solving a minimum-cost-flow problem or applying a modified version of the shortest path algorithms. Then the GGL bound is defined to be the solution of the following AQSPP:

$$LB_{GGL} = \min \left\{ \sum_{e \in A} c_e x_e + \sum_{e, f \in S_{2A}} z_{ef} x_e x_f : x \in X_{st} \right\}.$$

By the results of Section 3, the value of  $LB_{GGL}$  can be computed in polynomial time. It is now easy to show

**Theorem 3.**  $LB_{GGL}$  is a lower bound for QSPP; that is  $LB_{GGL} \leq z^*$ .

*Proof.* Let  $P$  be any  $s - t$  path in  $G$ , consisting of edges  $e_1, \dots, e_k$ . Then the cost of  $P$  is

$$c(P) = \sum_{i=1}^k c_{e_i} + \sum_{i,j=1}^k q_{e_i e_j} = \sum_{i=1}^k c_{e_i} + \sum_{i=1}^{k-1} \sum_{j=1}^k \hat{q}_{e_i e_{i+1}}^{e_j} \geq \sum_{i=1}^k c_{e_i} + \sum_{i=1}^{k-1} z_{e_i e_{i+1}}.$$

By definition, the latter expression is bounded from below by  $LB_{GGL}$ .  $\square$

Note that this approach can be easily generalized by using the  $r$ -Adjacent QSPP in order to obtain lower bounds. Clearly, as  $r$  is increased, the resulting bound will converge towards the optimal solution. However, the running time for computing the bound grows exponentially in  $r$ . Parameter  $r$  can thus be used to balance running time and quality of the bound.

## 4.2 An iterated Gilmore-Lawler type bound

Next, we present Iterated GL (IGL), an iterative bounding procedure inspired by the one proposed in [6] for the QAP. We start by defining a new cost matrix using the reduced costs associated with the dual problem of  $P_e$ .

$$\bar{q}_{ef} = q_{ef} + (\lambda_e)_k - (\lambda_e)_l - (\mu_e)_f \quad \forall f = (k, l) \in A \quad (6)$$

where  $\lambda_e$  is the optimal dual-solution vector associated with  $X_{st}$ , and  $\mu_e$  is the one associated with constraint  $x \leq 1$ . Using this matrix, and (5), we reformulate the QSPP by shifting some of the quadratic costs to the linear part.

$$\begin{aligned} \text{RQSPP: } \bar{z}^* = \min & \quad \sum_{e, f \in A} \bar{q}_{ef} x_e x_f + \sum_{e \in A} (c_e + z_e) x_e \\ \text{s.t. } & \quad x \in X_{st}, x \text{ binary.} \end{aligned} \quad (7)$$

The use of the reduced costs as the quadratic-cost matrix balances the increased linear costs making *RQSPP* equivalent to *QSPP* as shown by the following theorem. The proof is omitted due to space restrictions.

**Theorem 4.** *Problems QSPP and RQSPP are equivalent.* □

The theorem allows us to iterate the procedure by applying (6) to the reformulated problem and by repeating the reformulation. This results in a sequence of equivalent QSPP instances  $(Q_0, Q_1, \dots, Q_k$  with  $Q_0 = \text{QSPP}$ ), each characterized by a stronger impact of linear costs than the previous ones, and thus providing a better bound. Note that the GL bound is obtained by considering only the linear portion of the objective function in the first iteration.

## 5 Computational results

In this section, we present our computational experiments to evaluate the strength of the lower bounds for the QSPP presented in Section 4. We compare the results of the GLT, GGL, and IGL procedures with three other methods considered in [4]: the first is the root bound calculated by Cplex 12.4 when applied to the problem formulation (1). The other approaches, called QCR and OSU, are general approaches for solving quadratic 0-1 programming problems. The QCR (quadratic convex relaxation) method reformulates quadratic 0-1 programming with linear constraints into an equivalent 0-1 program with a convex quadratic objective function, where the reformulation is chosen such that the resulting lower bound is maximized. For this, an appropriate semidefinite program is solved [3]. The OSU (optimal separable underestimators) approach computes quadratic global underestimators of the objective function that are separable but not necessarily convex [4]. To evaluate and compare all methods, we use the random instances with  $|V| = 100, 121, 144, 169, 196, 225$  on grid graphs generated in [4]. The linear and quadratic costs are generated uniformly at random in  $\{1, \dots, 10\}$ . Given a pair of arcs  $(i, j)$  and  $(k, l)$ , their associated quadratic costs is equal to  $q = q_{ijkl} + q_{klij}$ . Since in each subproblem of our lower bounding schemes, each of these two values are processed separately, we consider a redistribution of the quadratic cost  $q_{ijkl} = q_{klij} = q/2$ —for IGL, we redistribute the costs at each iteration. Table 1 presents the results. The first two columns give the problem sizes and the optimal objective values. Columns three to eight give the lower bound values obtained by Cplex, QCR, OSU, GLT, GGL, and IGL respectively. The last five columns of the table present the percentage gap closed by QCR, OSU, GLT, GGL, and IGL over Cplex with respect to the optimum. The formula we used to compute the relative gap closed by a lower bound LB over the lower bound of Cplex (LBc) is  $100 \times (LB - LBc)/(OPT - LBc)$ .

The results show that Cplex provides by far the worst lower bounds. The GLT lower bound is better than the OUS bound, but both are outperformed by QCR, GGL, and IGL. GGL and IGL provide very similar bounds and clearly outperform QCR. Moreover, our purely combinatorial approach allows us to compute the GLT, GGL, and IGL bounds quickly, while the QCR bound requires

Table 1: Lower bound comparison for QSPP

Instance		Lower bound						Impv. vs. Cplex (%)				
$n$	Opt.	Cplex	QCR	OSU	GLT	GGL	IGL	QCR	OSU	GLT	GGL	IGL
100	621	200	489	357	434	528	511	68.8	37.2	55.5	77.9	73.8
100	635	211	501	323	419	511	512	68.3	26.4	49.1	70.7	70.9
100	636	217	498	367	449	532	530	56.4	35.7	55.3	75.1	74.7
100	661	209	491	359	447	537	534	62.3	33.1	52.6	72.5	71.9
100	665	233	504	367	453	549	545	62.7	31.1	50.9	73.1	73.2
<b>Ave.</b>								63.7	32.7	52.7	73.9	72.7
121	813	253	609	420	531	658	663	63.5	29.8	49.6	72.3	73.2
121	788	251	593	417	518	630	631	63.6	30.9	49.7	70.5	70.7
121	795	225	592	384	530	643	645	64.3	27.8	53.5	73.4	73.6
121	782	236	619	402	518	629	648	70.1	30.4	51.6	71.9	75.4
121	767	228	582	404	536	650	644	65.6	32.6	57.1	78.2	77.1
<b>Ave.</b>								65.4	30.3	52.3	73.2	74.0
144	959	271	714	479	623	767	775	64.3	30.2	51.1	72.1	73.2
144	963	282	707	524	627	768	764	62.4	35.3	50.6	71.3	70.7
144	900	259	687	491	592	730	735	66.7	36.1	51.9	73.4	74.2
144	960	236	698	481	625	758	766	63.8	33.8	53.7	72.1	73.2
144	976	289	701	479	632	773	772	59.9	27.6	49.9	70.4	70.3
<b>Ave.</b>								63.4	32.6	51.4	71.9	72.3
169	1159	335	805	586	730	899	891	57.0	30.4	47.9	68.4	67.4
169	1178	333	821	590	759	940	920	57.7	30.4	50.4	71.8	69.4
169	1164	325	822	558	733	883	876	59.2	27.7	48.6	66.5	65.6
169	1110	301	805	568	729	887	875	62.2	33.0	52.9	72.4	70.9
169	1115	322	842	567	737	918	897	65.5	30.8	52.3	75.1	72.5
<b>Ave.</b>								60.3	30.5	50.4	70.1	69.2
196	1363	364	959	680	841	1055	1064	59.5	31.6	47.7	69.1	70.1
196	1367	357	963	669	859	1058	1056	60.0	30.8	49.7	69.4	69.2
196	1320	334	934	651	820	1040	1009	60.8	32.1	50.0	72.6	69.4
196	1347	348	982	661	862	1058	1062	63.4	31.3	51.4	71.1	71.4
196	1344	354	949	704	868	1070	1043	60.1	35.3	51.9	72.3	69.5
<b>Ave.</b>								60.8	32.2	50.1	70.9	69.9
225	1551	367	1094	729	965	1199	1200	61.4	30.5	50.5	70.2	70.3
225	1588	412	1099	806	987	1223	1211	58.4	33.5	48.8	68.9	67.9
225	1561	419	1067	762	937	1169	1168	56.7	30.0	45.3	65.6	65.5
225	1569	386	1061	744	938	1173	1146	57.1	30.2	46.6	66.5	64.2
225	1582	389	1084	791	978	1223	1203	58.2	33.6	49.3	69.9	68.2
<b>Ave.</b>								58.4	31.6	48.1	68.2	67.2

solving a semidefinite program, which is often time-consuming in practice even if theoretically possible in polynomial time. Moreover, allowing a longer running time for our GGL approach, we could also improve our bounds by using the 3-Adjacent QSPP.

## 6 Conclusion

In this paper, we have investigated the quadratic variant of the shortest path problem. We have analyzed its complexity and studied polynomially solvable cases of the problem obtained by allowing only products of adjacent arcs in the objective function. We have proposed efficient procedures to compute strong

lower bounds that are based on the well-known Gilmore-Lawler approach combined with the polynomial solvability of the SPP and AQSP. Our future research will concentrate on combining the GGL procedure with some different reformulation techniques to improve the lower bounds, and an integration of these lower bounds into a branch-and-bound scheme.

## 7 Acknowledgments

The first author has been supported by the German Research Foundation (DFG) under grant BU 2313/2.

## References

1. Amaldi, E., Galbiati, G., Maffioli, F.: On minimum reload cost paths, tours, and flows. *Networks* 57(3), 254–260 (2011)
2. Bellman, R.: On a Routing Problem. *Quarterly of Applied Mathematics* 16, 87–90 (1958)
3. Billionnet, A., Elloumi, S., Plateau, M.C.: Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method. *Discrete Applied Mathematics* 157(6), 1185–1197 (2009)
4. Buchheim, C., Traversi, E.: Quadratic 0–1 optimization using separable underestimators. Tech. rep., Optimization Online (2015)
5. Caprara, A.: Constrained 0-1 quadratic programming: Basic approaches and extensions. *European Journal of Operational Research* 187(3), 1494 – 1503 (2008)
6. Carraraesi, P., Malucelli, F.: A new lower bound for the quadratic assignment problem. *Operations Research* 40(1-Supplement-1), S22–S27 (1992)
7. Dijkstra, E.W.: A note on two problems in connexion with graphs. *Numerische Mathematik* 1(1), 269–271 (1959)
8. Gilmore, P.C.: Optimal and suboptimal algorithms for the quadratic assignment problem. *Journal of the Society for Industrial & Applied Mathematics* 10(2), 305–313 (1962)
9. Lawler, E.L.: The quadratic assignment problem. *Management science* 9(4), 586–599 (1963)
10. Murakami, K., Kim, H.S.: Comparative study on restoration schemes of survivable atm networks. In: *INFOCOM’97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE. vol. 1*, pp. 345–352. IEEE (1997)
11. Sahni, S., Gonzalez, T.: P-complete approximation problems. *J. ACM* 23(3), 555–565 (Jul 1976)
12. Sen, S., Pillai, R., Joshi, S., Rathi, A.K.: A mean-variance model for route guidance in advanced traveler information systems. *Transportation Science* 35(1), 37–49 (2001)
13. Sivakumar, R.A., Batta, R.: The variance-constrained shortest path problem. *Transportation Science* 28(4), 309–316 (1994)