Approximating minimum quadratic assignment problems

Refael Hassin[*] Asaf Levin[†] Maxim Sviridenko[‡]

Abstract

We consider the well-known minimum quadratic assignment problem. In this problem we are given two $n \times n$ nonnegative symmetric matrices $A = (a_{ij})$ and $B = (b_{ij})$. The objective is to compute a permutation π of $V = \{1, \ldots, n\}$ so that $\sum_{\substack{i,j \in V \\ i \neq j}} a_{\pi(i),\pi(j)} b_{i,j}$ is minimized.

We assume that A is a 0/1 incidence matrix of a graph, and that B satisfies the triangle inequality. We analyze the approximability of this class of problems by providing polynomial bounded approximation for some special cases, and inapproximability results for other cases.

1 Introduction

In the MINIMUM QUADRATIC ASSIGNMENT PROBLEM two $n \times n$ nonnegative symmetric matrices $A = (a_{ij})$ and $B = (b_{ij})$ are given and the objective is to compute a permutation π of $V = \{1, \ldots, n\}$ so that $\sum_{\substack{i,j \in V \\ i \neq j}} a_{\pi(i),\pi(j)} b_{i,j}$ is minimized. The problem is one of the most important problem of combinatorial optimization. It generalizes many fundamental theoretical problems such as the TRAVELING SALESMAN PROBLEM, GRAPH BISECTION, MINIMUM WEIGHT PERFECT MATCHING, MINIMUM k-CLIQUE, LINEAR ARRANGEMENT, and many others. It also generalizes many practical problems that arise in various areas such as modeling of backboard wiring [20], campus and hospital layout [6, 8], scheduling [12] and many others [7, 17].

The MINIMUM QUADRATIC ASSIGNMENT PROBLEM (MQA) is a notoriously difficult problem both from practical and theoretical viewpoints. Practically, only instances with $n \approx 30$ are computationally tractable [2]. Theoretically, Sahni and Gonzalez [19] show that no constant factor approximation exists for the problem unless $P = NP$. In fact, Queyranne [18] showed that approximating the MQA within a polynomial factor in polynomial time implies P=NP even for the case when $a_{ij} = 1$ for every $i \neq j$, $a_{ii} = 0$ for every $i \in V$, and the weights in G_B correspond to a line metric.

In this paper we consider a special case, the MINIMUM METRIC QUADRATIC ASSIGNMENT PROBLEM (METRIC MQA), in which the weights in B satisfy the triangle inequality, $b_{i,j} \leq b_{i,k} + b_{k,j}$, for all $i, j, k \in V$ and A is a 0/1 incidence matrix of a graph. We use G_A to denote the graph corresponding to A and G_B to denote the complete weighted graph corresponding to the metric B. Thus, the problem is to compute in G_B a subgraph isomorphic to G_A of minimum total weight. We will denote by OPT the cost of an optimal solution to the MQA problem. An algorithm for a minimization problem is called *a ρ-approximation algorithm* if it always delivers in polynomial time a feasible solution whose cost is at most ρ times OPT.

Several interesting special cases of METRIC MQA can be solved in polynomial time, others are known to have polynomial algorithms that guarantee a solution withing a constant or a logarithmic

[*]Department of Statistics and Operations Research, Tel-Aviv University, Tel-Aviv 69978, Israel, hassin@post.tau.ac.il
[†]Department of Statistics, The Hebrew University, Jerusalem 91905, Israel, levinas@mscc.huji.ac.il
[‡]IBM T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598. sviri@us.ibm.com

factor from optimal. The approximability of other interesting cases is still open. In this paper we obtain new results on the approximability of METRIC MQA, thus narrowing the gap between the known cases that can and cannot be approximated.

Known results for the special cases. The METRIC k-TRAVELING SALESMAN PROBLEM is a special case of MQA, for which there is a known 2-approximation [11], and a 1.5-approximation when $k = n$ [5]. Similarly, the HAMILTONIAN PATH PROBLEM is a special case of MQA, for which a similar bound is known [16].

The case when G_B corresponds to a metric on n integer points $\{1, \ldots, n\}$ and G_A is a complete graph on n vertices is known as the LINEAR ARRANGEMENT PROBLEM and admits a $O(\sqrt{\log n} \log \log n)$-approximation [9].

When G_A consists of p vertex disjoint paths, (cycles, cliques) there are constant factor approximations under restrictions: For p fixed see [14, 15], and for equal-sized sets see [13].

The case when G_A is a matching corresponds to the MINIMUM MATCHING PROBLEM which is polynomially solvable.

The MAXIMUM METRIC QUADRATIC ASSIGNMENT PROBLEM seems to be a much easier problem since it admits a $\frac{1}{4}$-approximation algorithm [3]. Another case that admits good approximation is so-called DENSE QUADRATIC ASSIGNMENT PROBLEM. This subclass of problems has a polynomial time approximation scheme [4].

Our Results. First we consider the case when G_A is a spanning tree. Note that in this case the topology of the tree G_A is pre-specified and therefore the MQA on trees is different from the MINIMUM SPANNING TREE PROBLEM. We prove that there is no $O(n^\alpha)$-approximation algorithm for any $\alpha < 1$ for this special case, unless $P = NP$. On the positive side we show that if G_A is a *spider*, i.e. a tree with at most one vertex of degree ≥ 3, then there exists a constant factor approximation algorithm. For the case in which the maximum degree Δ of a vertex in the tree G_A is bounded, we present a $(\Delta \log n)$-approximation algorithm.

Finally, we consider the problem in the case when G_A is a special case of 3-regular Hamiltonian graph and the case when G_A is a double tour (see Section 4 for the exact definitions). We obtain a 3-approximation for the first problem and 2.25-approximation for the second one.

Techniques and ideas. The most nontrivial result of this paper is a constant factor approximation algorithm for the special case when G_A is a spider. Although this case looks quite specialized, it contains the Minimum Metric Hamiltonian Path Problem as a special case when the spider is just a path. We prove that by guessing the root and partitioning the vertex set into classes by their distance to the root, we could find a collection of spanning trees connecting each of the vertex sets to the root so that their total weight is at most constant factor of optimal value of the problem. This theorem provides us an auxiliary optimization problem that is easy to solve, and its output has cost approximating the cost of the optimal spider. The proof looks at each leg of the optimal solution (mapping of G_A into G_B) and uses a non-trivial charging technique to prove that within one leg we could find subtrees that span vertices of one class only and have bounded total cost. Given the collection of spanning trees we transform it into a tour and after that into a spider by a well known spanning tree doubling and short-cutting techniques.

The algorithm for the general bounded degree graphs consists of recursively finding the approximate Hamiltonian path and ordering vertices of the current subtree according to that path. After that we map subtrees in G_A into the path such that the subtree with more vertices is mapped closer to the beginning of the Hamiltonian path. Finally, we connect each child of the current root vertex by direct edges and repeat the algorithm with each subtree rooted at a child node.

2 Non-approximability of the MQA on trees

It is trivial to compute an $O(n)$-approximation for the MQA when G_A is a tree. If the tree is a spanning tree then by the triangle inequality, any feasible solution is an n-approximation. Otherwise, compute a 2-approximated k-MST where k is the number of vertices as the required tree (using Garg's [11] 2-approximation algorithm), and compute any feasible solution using these vertices. Again the bound follows from the triangle inequality. We now prove that this is essentially the best possible bound.

Theorem 1 *Unless $P = NP$, there is no a polynomial time n^{α}-approximation algorithm for the* MQA *when G_A is a tree, for any $\alpha < 1$.*

Proof: Similar to [18], we use a reduction from 3-PARTITION: Given $3k$ integers $s(1), \ldots, s(3k)$ such that $\sum_a s(a) = kR$, the goal is to decide whether $\{1, \ldots, 3k\}$ can be partitioned into k disjoint subsets $S_1, \ldots, S_k$ with $|S_h| = 3$ and $\sum_{a \in S_h} s(a) = R$ for $h = 1 \ldots, k$. The 3-PARTITION problem is known to be NP-complete in the strong sense (see problem [SP15] in [10]).

Consider a positive constant $\alpha < 1$, and suppose that there exists an algorithm $\mathcal{A}$ that guarantees a n^{α}-approximation to the MQA on trees. Let $P = 2\left(3k^2R\right)^l$, where $\frac{l}{l+1} > \alpha$.

Suppose that an instance I of 3-PARTITION is given. We define an instance of the MQA where G_A corresponds to the tree T with $1 + 3k^2RP$ vertices: a root vertex r, connected to $3k$ subtrees where the a-th subtree is a star with $3ks(a)P$ vertices. The graph G_B consists of k disjoint cliques, each with $3kRP$ vertices and with zero weight edges, plus one additional vertex v_r which does not belong to any of these cliques. The other edges of G_B have unit weight. Note that since 3-PARTITION is NP-hard in the strong sense, the resulting instance of the MQA has a polynomial size.

Suppose that I has a 3-partition $S_1, \ldots, S_k$. We can map the stars of T according to this partition to the cliques of G_B. The only unit length edges used by this solution are those connecting v_r to the centers of the stars. Therefore the value of this solution is $3k$.

Suppose now that I has no 3-partition. We claim that in this case the optimal solution to our MQA instance has value strictly greater than $3kP$. Consider feasible solution for such MQA instance. This solution defines an assignment of star centers into cliques and v_r in the graph G_B. Consider the clique with maximum number of centers assigned to it. Let N be the total size of the stars with centers assigned to that clique and v_r. Since there is no 3-partition, it follows that $N > (R+1)3kP$. Therefore, the MQA solution uses at least $3kP$ unit weight edges that connect star centers to their leaves, and its cost is strictly greater than $3kP$.

Note that in the graph we constructed, the number of vertices is $N = 1+3k^2RP = 1+\left(\frac{P}{2}\right)^{1+\frac{1}{l}} < P^{\frac{l+1}{l}}$, and therefore $P > N^{\frac{l}{l+1}} > N^{\alpha}$. Thus, a guaranteed error ratio less than N^{α} means that if a 3-partition exists in I then it must be found by the algorithm $\mathcal{A}$. ■

3 Approximating MQA on spiders

Definition 2 *A spider graph consists of a root vertex and a collection of subtrees that are paths. If the paths have equal lengths then the spider is uniform. These paths are called legs, and the size of a leg is the number of vertices in the leg excluding the root.*

We note that the proof of Theorem 1 does not apply to spiders. The MQA on spiders is obviously NP-hard even with just two subtrees since this is exactly the Hamiltonian Path Problem.

Theorem 3 *There is a polynomial 3-approximation algorithm for* MQA *on uniform spiders.*

Proof: Suppose that the spider T consists of a root r and l path subtrees of k vertices each. Then the algorithm by Altinkemer and Gavish [1] for the CAPACITATED MINIMUM SPANNING TREE PROBLEM with capacity k has performance guarantee 3 for our problem since every subtree returned by their algorithm is just a path. ∎

We now consider the MQA on general non-uniform spiders. We assume that the weights in the matrix B are positive integers (except zeros on the diagonal). Let $q \geq 1$ be a constant to be chosen later. We assume that we know the root vertex r of the tree whose degree is at least three in a fixed optimal solution OPT whose cost is also denoted by OPT. This assumption can be justified by testing all possibilities for choosing r and applying the following algorithm for each possibility.

We partition the vertices in $V \setminus \{r\}$ according to their distances from r in the following way. Let V_i be the set of vertices whose distance from r belongs to the interval $[q^{i-1}, q^i)$, that is $V_i = \{j \in V \setminus \{r\} : q^{i-1} \leq b_{rj} < q^i\}$. For a vertex j, we say that j is a *class i vertex* if $j \in V_i$. Let l be the maximum index for which $V_i \neq \emptyset$.

For each i, we compute an approximate minimum tour C_i on the set of vertices $V_i \cup \{r\}$ by first computing the minimum spanning tree T_i over $V_i \cup \{r\}$ in G_B, doubling T_i, converting the new graph into a Eulerian tour and finally short-cutting the Eulerian tour to get the Hamiltonian tour on $V_i \cup \{r\}$. We next create a Hamiltonian path P over V in which the indices of classes of the vertices along the path are monotone non-decreasing sequence. We do so by first placing r, then V_1, and so on until we place V_l at the end of the path. For each i, the order of V_i along this tour is exactly the order in C_i, starting at arbitrary chosen vertex in C_i. Let $v_1 = r, v_2, \ldots, v_n$ be the permutation of the vertices along the Hamiltonian path P.

Assume that the input spider G_A has legs of size $n_1 \leq n_2 \leq \cdots \leq n_t$. We return the spider whose root vertex is r, and its edge set is $E_1 \cup E_2$ where $E_1 = \{(r, v_k) : k = 1 + \sum_{j=1}^{i-1} n_j, i = 1, 2, \ldots, t\}$ and $E_2 = \{(v_i, v_{i+1}) : i = 2, 3, \ldots, n-1\} \setminus \{(v_{k-1}, v_k) : k = 1 + \sum_{j=1}^{i} n_j, i = 1, 2, \ldots, t\}$. I.e., we start to allocate the vertices along the order of P to different legs starting from the shortest leg that is allocated the vertices from the classes with smallest index.

Before we start to estimate the weight of the approximate solution we prove the following technical lemma.

Lemma 4 *We are given a set of positive numbers $a_1 \leq a_2 \leq \cdots \leq a_n$. Let $Q_1, \ldots, Q_t$ be a partition of the set $\{1, \ldots, n\}$ such that $|Q_i| = n_i$, $n_1 \leq n_2 \leq \cdots \leq n_t$, and $Q_i = \{j : j = 1 + \sum_{s=1}^{i-1} n_s, \ldots, \sum_{s=1}^{i} n_s\}$. Let $P_1, \ldots, P_t$ be arbitrary partition of the set $\{1, \ldots, n\}$ such that $|P_i| = n_i$. Then*

$$\sum_{i=1}^{t} \max_{j \in Q_i} a_j \leq \sum_{i=1}^{t} \max_{j \in P_i} a_j .$$

Proof: The proof is a straightforward application of induction on the number t of the sets in the partition. ∎

It is clear that the algorithm returns a feasible solution in polynomial time. It remains to analyze its performance guarantee. We bound separately the cost of E_1 and the cost of E_2. We first bound the cost of E_1.

Lemma 5 $\sum_{(r,i) \in E_1} b_{ri} \leq q \cdot \mathrm{OPT}$.

Proof: Consider the i-th leg in a fixed optimal solution. Assume that it consists of the vertices $r, u_1^i, u_2^i, \ldots, u_{n_i}^i$ in this order. Then, the total cost of this leg is $b_{ru_1^i} + \sum_{j=1}^{n_i - 1} b_{u_j^i u_{j+1}^i} \geq \max_{1 \leq j \leq n_i} b_{ru_j^i}$

by triangle inequality. We sum this inequality for all the legs, and conclude that

$$\mathrm{OPT} = \sum_{i=1}^{t}\left[b_{ru_1^i} + \sum_{j=1}^{n_i-1} b_{u_j^i,u_{j+1}^i}\right] \geq \sum_{i=1}^{t} \max_{1\leq j\leq n_i} b_{ru_j^i} \geq \sum_{i=1}^{t} \max_{1\leq j\leq n_i} b'_{ru_j^i}$$

where for a vertex $v \in V_i$ we define $b'_{rv} = q^{i-1}$.

On the other hand, we note that along the Hamiltonian path P the vertices are ordered according to the value of b'_{rv}. Let $j(1) < j(2) < \cdots < j(t)$ be the indices such that $(r, v_{j(i)}) \in E_1$ for $i = 1, 2, \ldots, t$. By the definition of the path P we have $j(s) = 1 + \sum_{i=1}^{s-1} n_i$. Therefore, $b'_{rv_{j(i)}} \leq \min_{k=0,\ldots,n_i-1} b'_{rv_{j(i)+k}} \leq \max_{k=0,\ldots,n_i-1} b'_{rv_{j(i)+k}}$. Applying Lemma 4 we obtain

$$\sum_{(r,i)\in E_1} b_{ri} \leq q \cdot \sum_{i=1}^{t} b'_{rv(j(i))} \leq q \cdot \sum_{i=1}^{t} \max_{k=0,\ldots,n_i-1} b'_{rv_{j(i)+k}} \leq q \cdot \sum_{i=1}^{t} \max_{1\leq j\leq n_i} b'_{ru_j^i} \leq q \cdot \mathrm{OPT}.$$

■

Next we bound the cost of E_2 by bounding the cost of path P. We do it by proving the existence of the collections of trees defined on $V_i \cup \{r\}$ with bounded total cost.

Lemma 6 *There exists a collection of trees T_i defined on the sets $V_i \cup \{r\}$ with total cost bounded above by $\frac{3q}{q-1}$OPT.*

Proof: Given an optimal solution OPT and the set of vertices $V_i \cup \{r\}$, we construct tree T_i as follows. Consider a leg $L_U = (r = u_1, \ldots, u_k)$ of OPT with vertex set U such that $U \cap V_i \neq \emptyset$. We scan L_U from u_1 to u_k and consider each vertex $u \in L_U \cap V_i$. For each such vertex we act as follows:

- Suppose that either all vertices between u and the previous vertex $v \in V_i \cup \{r\}$ belong to V_{i-1} or they all belong to V_{i+1}. In such a case we add the edge $e = (v, u)$ into the tree T_i and define $charge(e)$ to be the length of the $v - u$ path in L_U.

- Suppose that there is a vertex $w \in L_U$ such that all vertices between w and u belong to V_{i-1} and $w \notin V_i \cup V_{i-1}$. In this case we connect u directly to the root r, i.e. we add the edge $e = (r, u)$ to the tree T_i, and define $charge(e)$ to be the length of the $w - u$ path in L_U. The first edge (w, w') of this path will be called the *witness* of the edge $e = (r, u)$. Since $w \notin V_i \cup V_{i-1}$ and $u \in V_i$ we have by triangle inequality that

  $$charge(e) \geq b_{wu} \geq b_{ru} - b_{rw} \geq b_{ru}(1 - 1/q)$$

  if $w \in V_s$ for $s \leq i - 2$ and

  $$charge(e) \geq b_{wu} \geq b_{ww'} \geq b_{rw} - b_{rw'} \geq b_{rw}(1 - 1/q) \geq b_{ru}(1 - 1/q)$$

  if $w \in V_s$ for $s \geq i + 1$.

- Analogously, suppose that there is a vertex $w \in L_U$ such that all vertices between w and u belong to V_{i+1} and $w \notin V_i \cup V_{i+1}$. We connect u directly to the root r and define $charge(e)$ to be the length of the $w - u$ path in L_U. The first edge (w, w') of this path is the witness of the edge $e = (r, u)$. The lower bound for $charge(e)$ is computed similarly:

  $$charge(e) \geq b_{wu} \geq b_{rw} - b_{ru} \geq b_{rw}(1 - 1/q) \geq b_{ru}(1 - 1/q)$$

if $w \in V_s$ for $s \geq i+2$ and

$$charge(e) \geq b_{wu} \geq b_{ww'} \geq b_{rw'} - b_{rw} \geq b_{rw'}(1-1/q) \geq b_{ru}(1-1/q)$$

if $w \in V_s$ for $s \leq i-1$.

The above inequalities imply that the total length of all edges in trees T_i is upper bounded by $\frac{q}{q-1}\sum_i \sum_{e \in T_i} charge(e)$.

To complete the proof we prove that $\sum_i \sum_{e \in T_i} charge(e) \leq 3\mathrm{OPT}$. Indeed, if $(u_s, u_{s+1}) \in L_U$ and both vertices belong to the same set V_i then the edge (u_s, u_{s+1}) may contribute only to $charge(e)$ for $e \in T_{i-1} \cup T_i \cup T_{i+1}$. If u_s and u_{s+1} belong to different sets V_i then the edge (u_s, u_{s+1}) could be a witness for at most one edge (r, u). Also if $u_s \in V_i$ and $u_{s+1} \in V_j$ and $|i-j| = 1$ then the edge (u_s, u_{s+1}) contributes once to the $charge(e)$ for some edge in T_i and once for some in T_j. ■

It follows that by finding a minimum spanning trees on each set $V_i \cup \{r\}$, and then doubling and short-cutting these trees, we get the path P with total cost bounded above by $\frac{6q}{q-1}\mathrm{OPT}$. By Lemma 5 we conclude the following theorem:

Theorem 7 $w(E_1) + w(E_2) \leq \left(\frac{6q}{q-1} + q\right)\mathrm{OPT}$.

Choosing $q = 1 + \sqrt{6}$ we obtain a $(7 + 2\sqrt{6})$-approximation algorithm (note $7 + 2\sqrt{6} \approx 11.9$) for the MQA on non-uniform spiders.

4 Other types of graphs G_A

4.1 Bounded degree trees

Suppose that the tree G_A has a maximum degree of at most Δ, where Δ is some fixed constant. For this case we present an $O(\Delta \log n)$-approximation algorithm. Note that when Δ is a constant (e.g., $\Delta = 3$ if G_A is a binary tree), this result gives a logarithmic approximation factor.

The algorithm first approximates a Hamiltonian path in G_B, and denotes the order of the vertices along this path as $v_1, v_2, \ldots, v_n$. The cost of this Hamiltonian path is at most twice the cost of a minimum cost spanning tree, and hence at most 2OPT. We root G_A in an arbitrary vertex $root$, and map $root$ to v_1, i.e., one of the endpoints of the Hamiltonian path.

Next, we recursively map the vertices of the tree G_A (starting from $root$). We assume that the current vertex is v that is mapped to v_r, and the subtree rooted at v contains n_v vertices, is mapped to a consecutive set of n_v vertices along the Hamiltonian path starting at v_r. I.e., the subtree rooted at v is mapped to the sub-path $v_r, v_{r+1}, \ldots, v_{r+n_v-1}$. We start this recursive procedure by mapping $root$ to v_1 and the subtree rooted at $root$ to the entire Hamiltonian path $v_1, v_2, \ldots, v_n$. Assume that in the current recursion call we process vertex v that is mapped to v_r. Consider the number of vertices in the subtrees hanged at each of the children of v. Assume that v has $\delta \leq \Delta$ children, where the i-th child denoted as c_i has n_i vertices in its subtree (so $\sum_{i=1}^{\delta} n_i = n_v - 1$). W.l.o.g. we assume that $n_1 \leq n_2 \leq \cdots \leq n_\delta$. We map c_i to $v_{j(i)}$ where $j(i) = r + 1 + \sum_{k=1}^{i-1} n_k$. We will allocate recursively the vertices of the subtree rooted at c_i to the vertex set $\{v_{j(i)}, v_{j(i)+1}, \ldots, v_{j(i+1)-1}\}$. This completes the definition of the solution.

The edges connecting v and its children in G_A are associated with v. The cost of the edges associated with v is at most δ times the cost of the subpath $v_r, v_{r+1}, \ldots, v_{j(l)}$, and we call this subpath *the evidence subpath of* v. We *charge* the edges of the evidence subpath of v for the edges connecting v and its children. Therefore, we conclude that if we can prove a bound B on the number of times each edge is charged, then the total cost of the resulting solution is at most ΔB times the cost of the Hamiltonian path.

First, note that each time an edge $e = (v_i, v_{i+1})$ is charged e belongs to an evidence subpath of some vertex v_i^e, and the vertices v_i^e and v_j^e (for $i \neq j$) belong to a common path in G_A from $root$ to a leaf. Next, we consider the number of vertices in the subtree rooted at v_i^e, and denote it by n_i^e. We argue that $n_{i+1}^e \leq \left(1 - \frac{1}{\Delta}\right) \cdot n_i^e$. Since $n_l \geq n_i$ for all i, the number of edges of the evidence subpath of v_i^e is at most $\left(1 - \frac{1}{\Delta}\right) \cdot n_i^e$. Note that v_{i+1}^e is a descendant of one of the children of v_i^e that is an inner vertex of the evidence subpath of v_i^e because otherwise the edges associated with v_{i+1}^e do not belong to the evidence subpath of v_i^e. Therefore, we conclude that $n_{i+1}^e \leq \left(1 - \frac{1}{\Delta}\right) \cdot n_i^e$. Since, for all i $1 \leq n_i^e \leq n$, we conclude that the number of times that an edge is charged is at most $B \leq O(\log_{\frac{\Delta-1}{\Delta}} n)$. Therefore, we conclude the following theorem (for constant values of Δ):

Theorem 8 *There is an $O(\log n)$-approximation algorithm for the* BOUNDED-DEGREE TREE MQA.

4.2 Hamiltonian 3-regular graphs

The approximability of GENERAL HAMILTONIAN 3-REGULAR-MQA is currently open. We describe in the sequel an approximable special case.

Given a graph with an even number of vertices, a *wheel* is a Hamiltonian tour say $\{(v_i, v_{i+1}) : i = 1, \ldots, n\}$ (indices are modulo n) and the edges $\{(v_i, v_{i+\frac{n}{2}}) : i = 1, \ldots, \frac{n}{2}\}$.

We note that a shortest (or approximate) tour does not guarantee any bound for WHEEL-MQA. To see this, consider points $p_1, \ldots p_{2n}$ ordered by their indices and uniformly scattered along a unit cycle. Of course, the cycle is a shortest tour. Its weight in the WHEEL-MQA is its length plus n times its diameter, i.e, $2\pi + n$. However, there is a much better solution that visits consecutively $p_1, p_3, \ldots, p_{2n-1}$ and then $p_2, p_4, \ldots, p_{2n}$. Its weight is approximately three times the length of the cycle, i.e., 6π.

Theorem 9 *There is a polynomial 3-approximation algorithm for* WHEEL-MQA.

Proof: Compute a minimum weight perfect matching $M = \{(a_1, b_1), \ldots, (a_{\frac{n}{2}}, b_{\frac{n}{2}})\}$ on G_B. Construct a 1.5 approximation tour T for the TSP on the graph with vertices $\{a_1, \ldots, a_{\frac{n}{2}}\}$. By the triangle inequality, $w(T) \leq 1.5w(T^*)$, where $w(T^*)$ is the length of an optimal tour over V. W.l.o.g., assume that $T = \{a_1, \ldots, a_{\frac{n}{2}}\}$. Let T_A be the tour $(a_1, a_2, \ldots, a_{\frac{n}{2}}, b_1, \ldots, b_{\frac{n}{2}}, a_1)$. (See Figure 1.) Return the union of T_A and M.

By the triangle inequality $w(b_i, b_{i+1}) \leq w(a_i, a_{i+1}) + w(a_i, b_i) + w(a_{i+1}, b_{i+1})$ for all $i = 1, \ldots, n/2 - 1$, $w(a_{\frac{n}{2}}, b_1) \leq w(a_{\frac{n}{2}}, a_1) + w(a_1, b_1)$ and $w(a_1, b_{\frac{n}{2}}) \leq w(a_{\frac{n}{2}}, a_1) + w(a_{\frac{n}{2}}, b_{\frac{n}{2}})$. Therefore,

$$\begin{aligned}
w(T_A) &= \sum_{i=1}^{\frac{n}{2}-1} [w(a_i, a_{i+1}) + w(b_i, b_{i+1})] + w(a_{\frac{n}{2}}, b_1) + w(a_1, b_{\frac{n}{2}}) \\
&\leq \sum_{i=1}^{\frac{n}{2}-1} [2w(a_i, a_{i+1}) + w(a_i, b_i) + w(a_{i+1}, b_{i+1})] + 2w(a_{\frac{n}{2}}, a_1) + w(a_1, b_1) + w(a_{\frac{n}{2}}, b_{\frac{n}{2}}) \\
&\leq 2w(T) + 2w(M).
\end{aligned}$$

Therefore, $apx = w(T_A) + w(M) \leq 3[w(M) + w(T^*)]$, whereas, $opt = w(T_{opt}) + w(M_{opt}) \geq w(T^*) + w(M)$, where the last inequality holds because of the triangle inequality. ■

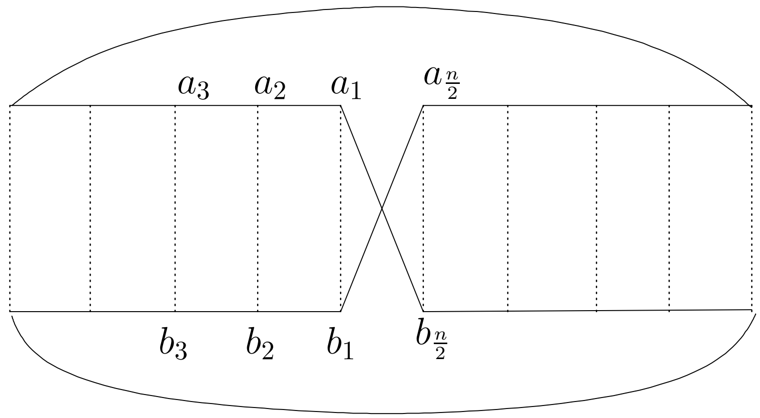

Figure 1: The tour T_A

4.3 Double tours

A *double tour* consists of the edges of a tour, say $\{(v_i, v_{i+1}) : i = 1, \dots, n\}$ (indices are modulo n) and their *shortcuts* $\{(v_i, v_{i+2}) : i = 1, \dots, n\}$.

Theorem 10 *A 1.5-approximation for* METRIC TSP *is a 2.25-approximation for the corresponding* DOUBLE TOUR-MQA *instance.*

Proof: By triangle inequality, the total length of the shortcuts is at most twice the length of the approximated tour. Therefore, the total length of the solution is at most 4.5 times that of a shortest tour. The result follows since any feasible solution has length of at least twice the shortest tour. This last claim holds because the optimal solution consists of a disjoint union of two Hamiltonian cycles. This is so for odd values of n as the set of shortcut edges is the edge set of a Hamiltonian cycle $(1, 3, 5, \dots, n = 0, 2, \dots, n-1, 1)$, and for even values of n this is so because the following are the two cycles: $1, 3, 5, \dots, n-1, n, n-2, n-4, \dots, 4, 2, 1$ and $2, 3, 4, \dots, n-2, n-1, 1, n, 2$. ■

References

[1] K. Altinkemer and B. Gavish, "Heuristics with constant error guarantees for the design of tree networks," Management Science, **34**, 331-341, 1988.

[2] K. Anstreicher, Recent advances in the solution of quadratic assignment problems, ISMP, 2003 (Copenhagen). Math. Program. 97 (2003), no. 1-2, Ser. B, 27–42.

[3] E. Arkin, R. Hassin and M. Sviridenko, Approximating the Maximum Quadratic Assignment Problem, Information Processing Letters v.77 (2001), pp. 13-16 .

[4] S. Arora, A. Frieze and H. Kaplan, A new rounding procedure for the assignment problem with applications to dense graph arrangement problems, Math. Program. 92 (2002), no. 1, Ser. A, 1–36.

[5] N. Christofides, " Worst-case analysis of a new heuristic for the traveling salesman problem", *Technical report* 338, *Graduate School of Industrial Administration, CMU,* 1976.

[6] J. Dickey and J. Hopkins, Campus building arrangement using TOPAZ, Transportation Science **6** (1972), 59-68.

[7] H. Eiselt and G. Laporte, A combinatorial optimization problem arising in dartboard design, Journal of Operational Research Society **42** (1991), 113-118.

[8] A. Elshafei, Hospital layout as a quadratic assignment problem, Operations Research Quarterly **28** (1977), 167-179.

[9] U. Feige and J. Lee, An improved approximation ratio for the minimum linear arrangement problem, manuscript, 2005.

[10] M.R. Garey and D.S. Johnson, "Computers and intractability a guide to the theory of NP-completeness", W.H. Freeman and company, New-York, 1979.

[11] N. Garg, "Saving an epsilon: a 2-approximation for the k-MST problem in graphs," STOC 2005, 396-402.

[12] A. Geoffrion and G. Graves, Scheduling parallel production lines with changeover costs: Practical applications of a quadratic assignment/LP approach, Operations Research **24** (1976), 596-610.

[13] M.X. Goemans and D.P. Williamson, "A general approximation technique for constrained forest problems", *SIAM Journal on Computing* **24** (1995), 296-317.

[14] N. Guttmann-Beck and R. Hassin, "Approximation algorithms for minimum tree partition", *Disc. Applied Math.* **87** (1998), 117-137.

[15] N. Guttmann-Beck and R. Hassin, "Approximation algorithms for min-sum p-clustering", *Disc. Applied Math.* **89** (1998), 125-142.

[16] J. A. Hoogeveen, "Analysis of Christofides' heuristic: Some paths are more difficult than cycles," *Operations Research Letters*, **10** (1991), pages 291-295.

[17] G. Laporte and H. Mercure, Balancing hydraulic turbine runners: A quadratic assignment problem, European Journal of Operations Research **35** (1988), 378-381.

[18] M. Queyranne, "Performance ratio of polynomial heuristics for triangle inequality quadratic assignment problems", *Operations Research Letters*, **4** (1986), 231-234.

[19] S. Sahni and T. Gonzalez, "P-complete approximation problems", *J. ACM* **23** (1976) 555-565.

[20] L. Steinberg, The backboard wiring problem: a placement algorithm. SIAM Rev. 3 1961 37–50.