Deterministic 7/8-approximation for the Metric Maximum TSP*

Łukasz Kowalik and Marcin Mucha
Institute of Informatics, University of Warsaw, Poland.
{kowalik,mucha}@mimuw.edu.pl

Abstract

We present the first 7/8-approximation algorithm for the maximum traveling salesman problem with triangle inequality. Our algorithm is deterministic. This improves over both the randomized algorithm of Hassin and Rubinstein [2] with expected approximation ratio of $7/8-O(n^{-1/2})$ and the deterministic $(7/8-O(n^{-1/3}))$-approximation algorithm of Chen and Nagoya [1].

In the new algorithm, we extend the approach of processing local configurations using so-called loose-ends, which we introduced in [4].

1 Introduction

The Traveling Salesman Problem and its variants are among the most intensively researched problems in computer science and arise in a variety of applications. In its classical version, given a set of vertices V and a symmetric weight function $w : V^2 \rightarrow \mathbb{R}_{\geq 0}$ satisfying the triangle inequality one has to find a Hamiltonian cycle of minimum weight.

There are several variants of TSP, e.g. one can look for a Hamiltonian cycle of minimum or maximum weight (MAX-TSP), the weight function can be symmetric or asymmetric, it can satisfy the triangle inequality or not, etc.

In this paper, we are concerned with the MAX-TSP variant, where the weight function is symmetric and satisfies the triangle inequality. This variant is often called *the metric MAX-TSP*.

MAX-TSP (not necessarily metric) was first considered by Serdyukov in [5], where he gives a $\frac{3}{4}$-approximation. Next, a $\frac{5}{6}$-approximation algorithm for the metric case was given by Kostochka and Serdyukov [3]. Hassin and Rubinstein [2] used these two algorithms

*This research is partially supported by a grant from the Polish Ministry of Science and Higher Education, project N206 005 32/0807. A preliminary version of this paper appeared in Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques (APPROX and RANDOM 2008), LNCS 5171, pp. 132-145, 2008.

together with new ideas to achieve a randomized approximation algorithm with expected approximation ratio of $(\frac{7}{8} - O(n^{-1/2}))$. This algorithm has later been derandomized by Chen and Nagoya [1], at a cost of a slightly worse approximation factor of $(\frac{7}{8} - O(n^{-1/3}))$.

In this paper, we give a deterministic $\frac{7}{8}$-approximation algorithm for metric MAX-TSP. Our algorithm builds on the ideas of Serdyukov and Kostochka, but is completely different from that of Hassin and Rubinstein. We apply techniques similar to those used earlier in [4] for the directed version of MAX-TSP with triangle inequality.

1.1 Closer look at previous results

Classic undirected MAX-TSP algorithm of Serdyukov [5] starts by constructing two sets of edges of the input graph G: a maximum weight cycle cover $\mathcal{C}$ and a maximum weight matching M, and then removing a single edge from each cycle of $\mathcal{C}$ and adding it to M. It can be shown that we can avoid creating cycles in M, so in the end we get two sets of paths: $\mathcal{C}'$ and M'. These sets can be extended to Hamiltonian cycles arbitrarily. Since we started with a maximum weight cycle cover and a maximum weight matching, we have $w(\mathcal{C}') + w(M') \geq w(\mathcal{C}) + w(M) \geq \frac{3}{2}\mathrm{OPT}$. It follows that the better of the two cycles has weight at least $\frac{3}{4}\mathrm{OPT}$. Here, we used two standard inequalities: $w(\mathcal{C}) \geq \mathrm{OPT}$ and $w(M) \geq \frac{1}{2}\mathrm{OPT}$. The latter only holds for graphs with even number of vertices. The case of odd number of vertices needs separate treatment.

Serdyukov's algorithm works for any undirected graph, with weight function not necessarily satisfying the triangle inequality. However, if this inequality is satisfied, we can get a much better algorithm. Kostochka and Serdykov observed the following useful fact (see e.g. [2] for a proof).

Lemma 1.1 (Kostochka, Serdyukov [3])**.** *Let $G = (V, E)$ be a weighted complete graph with a weight function $w : E \to \mathbb{R}_{\geq 0}$ satisfying the triangle inequality. Let $\mathcal{C}$ be a cycle cover in G and let $Q = \{e_1, \ldots, e_{|\mathcal{C}|}\}$ be a set of edges with exactly one edge from each cycle of $\mathcal{C}$. Then the collection of paths $\mathcal{C} \setminus Q$ can be extended in polynomial time to a Hamiltonian cycle H with*

$$w(H) \geq w(\mathcal{C}) - \sum_{i=1}^{|\mathcal{C}|} w(e_i)/2.$$

Kostochka and Serdyukov [3] propose an algorithm which starts by finding a maximum weight cycle cover $\mathcal{C}$ and then applies the above lemma with Q consisting of the lightest edges of cycles in $\mathcal{C}$. Since all cycles have length at least 3, the weight of the removed edges amounts to at most $\frac{1}{3}w(\mathcal{C})$, so we regain at least $\frac{1}{6}w(\mathcal{C})$, which leads to $\frac{5}{6}$-approximation. (Note that if it happens that all the cycles in $\mathcal{C}$ have length at least 4 we get $\frac{7}{8}$-approximation).

2 Our approach

Similarly to Serdykov's algorithm (as well as that of Hassin and Rubinstein), our algorithm starts by constructing a maximum weight cycle cover $\mathcal{C}$ and maximum weight matching M. In our reasoning we need the inequality $w(M) \geq \frac{1}{2}\mathrm{OPT}$, which holds only for graphs with even number of vertices. In what follows we only consider such graphs and in Section 5 we show that the odd case reduces to the even case in polynomial time.

In all previous algorithms edges are moved from the cycle cover $\mathcal{C}$ to the matching M. We do not follow this approach. Instead, we remove some edges from $\mathcal{C}$ and add some edges to M. The edges added to M are not necessarily the edges removed from $\mathcal{C}$. In fact, they might not even be cycle edges in $\mathcal{C}$. All we need to guarantee is that their total weight is sufficiently large compared to the weight loss in $\mathcal{C}$.

Here is how it works. Let $\min(C_i)$ be the lightest edge of a cycle $C_i \in \mathcal{C}$. Since removing a single edge from each C_i and then joining the resulting paths using Lemma 1.1 results in the weight loss equal to half the weight of the removed edges, it should be clear that we should remove $\min(C_i)$ from each C_i. The weight loss is then $\sum_i w(\min(C_i))/2$.

We are going to describe an iterative process of adding edges to a collection of paths P, initially equal to M. Edges will be added in *phases*, each phase corresponds to a single cycle $C_i \in \mathcal{C}$. After finishing the phase corresponding to C_i we will call C_i *processed*. The edges added in the phase corresponding to C_i will usually, but not necessarily belong to C_i or at least connect vertices of C_i. Their total weight will also be directly related to $w(C_i)$ and $w(\min(C_i))$. Let $(\alpha, \beta) \star C_i = \alpha w(C_i) + \beta w(\min(C_i))$. The following Lemma shows why this is a useful definition:

Lemma 2.1. *If during processing the cycles in $\mathcal{C}$, we can add edges of total weight at least $\sum_{C_i \in \mathcal{C}} (\alpha, 1/2) \star C_i$ to M, then we get a $(3/4 + \alpha/2)$-approximation algorithm.*

Proof. Let H_1 be the Hamiltonian cycle obtained from $\mathcal{C}$ by using Lemma 1.1, and let H_2 be the cycle obtained from M by processing all cycles of $\mathcal{C}$ and patching the resulting collection of paths into a Hamiltonian cycle. Then

$$w(H_1) + w(H_2) \geq \left[w(\mathcal{C}) - \sum_i w(\min(C_i))/2 \right] +$$

$$+ \left[w(M) + \alpha w(\mathcal{C}) + \sum_i w(\min(C_i))/2 \right] \geq (3/2 + \alpha)\mathrm{OPT},$$

so the heavier of the two cycles is a $(3/4 + \alpha/2)$-approximation. □

In the remainder of the paper, we show that this can be done for $\alpha = 1/4$, yielding a 7/8-approximation.

2.1 Skeleton of the algorithm

A graph P is sub-Hamiltonian if it is a family of disjoint paths or a Hamiltonian cycle (i.e. it can be extended to a Hamiltonian cycle). Let P be a family of disjoint paths. We say that set of edges S is *allowed* w.r.t. P, if S is disjoint from P and the edge sum of P and S is sub-Hamiltonian. We call an edge e *allowed* w.r.t P if $\{e\}$ is allowed w.r.t. P. If an edge is not allowed, we call it *forbidden*.

In the algorithm presented below, we maintain a sub-Hamiltonian graph P satisfying the following invariant.

Invariant 1. For any vertex v, if $\deg_P(v) = 2$ then the cycle v belongs to has been already processed.

Consider a phase of our algorithm and let C be the cycle that is still unprocessed. In this situation a set S of edges will be called a *support* of C if S is allowed w.r.t. P, and after adding S to P (and thus making C processed) Invariant 1 is satisfied.

The following is the skeleton of the algorithm, that we will develop in the remainder of the paper.

Algorithm 2.1 Main Algorithm

```
Let M be a heaviest matching and 𝒞 a heaviest cycle cover in G.
Let H_1 be the Hamiltonian cycle obtained from 𝒞 by using Lemma 1.1.
P := M
Mark all cycles in 𝒞 as unprocessed.
for each unprocessed cycle C in 𝒞 do
    Find S, a support of C of large weight.
    P := P ∪ S
    Mark C as processed.
Arbitrarily patch P to a Hamiltonian cycle H_2.
Return the heavier of H_1 and H_2.
```

2.2 Loose-ends

When considering a cycle C_i, we are going to extend P by adding some edges connecting the vertices of C_i. Ideally we would like to add $n_i/2$ new edges, where n_i is the length of C_i. However, this is not always possible, because some of the cycles have odd length and $n_i/2$ is not an integer. Instead we are going to use the idea of loose-ends introduced in [4].

A *loose-end* is a vertex v, for which $\deg_P(v) = 1$ even though the cycle it belongs to is already processed. A vertex v of cycle $C \in \mathcal{C}$ becomes a loose-end if no edge adjacent to v is added to P when C is processed. This vertex can be connected with some other vertex at a later stage and cease being a loose-end.

Consider two odd-length cycles C_1 and C_2, say both of length 5. When we process C_1, we can only add 2 edges to M, and some vertex $v \in C_1$ is not an endpoint of any of these edges, so it becomes a loose-end. Later, when we process C_2, we can add 3 edges to M, by connecting one of C_2's vertices with v. Using the triangle inequality, we can guarantee that this edge has large weight. So in this case we get a little less weight from C_1 and a little more weight from C_2. It is important to process cycles in order that guarantees that the weight lost when processing the earlier cycles (the ones that give loose-ends) is dominated by the weight gained when processing the later cycles (the ones that use loose-ends). We will show that the algorithm can determine this order.

Let S be a support of C in some phase of the algorithm. We will say that S is a k-support if after adding it to P (and thus processing cycle C) the number of loose-ends increases by at least k (k could be negative here).

In the following section we describe in detail how the cycles are processed in our algorithm. For even-length cycles we construct heavy 0-supports, and for odd-length cycles we construct both (-1)-supports and $(+1)$-supports.

When constructing (-1)-supports, we need to assume that at least one loose-end is available. Unfortunately, just one loose-end may be insufficient to guarantee the existence of a (-1)-support. This could happen if the loose-end u is connected to C, the cycle being processed, by a path in P. In that case, adding an edge between u and a vertex of C to P may create a cycle in P. This is acceptable only if that cycle is Hamiltonian (in particular, C would have to be the last cycle processed). Luckily, it turns out that two loose-ends are always sufficient to avoid creating such short cycles. Thus, when describing a (-1)-support for each odd cycle we will consider two situations: when there are two loose-ends, and when there is exactly one loose-end but the algorithm is in the last (i.e. $|\mathcal{C}|$-th) phase.

3 Processing cycles

In this section we consider an arbitrary phase of the algorithm and we describe supports of unprocessed cycles. The construction of a support of such a cycle C may depend on the number of loose-ends and the way the collection P of paths constructed so far interacts with C, in particular on which edges of C are forbidden etc.

The following observations will be used in many of our proofs.

Observation 1. Let C be an unprocessed cycle and let $M \subset E(C)$ be a matching. Let $\tilde{C}$ be any cycle in $P \cup M$. Then if $\tilde{C}$ contains an allowed edge of M, it contains at least two allowed edges. Also, if $\tilde{C}$ contains a forbidden edge of M, it contains exactly one edge of M. □

Observation 2. In any phase of the algorithm and for any unprocessed cycle C, forbidden edges with both endpoints in C form a matching. □

Consider an unprocessed cycle C. A set of edges S will be called a *semi-support* of C when $P \cup S$ contains vertices of degree at most 2, and after adding S to P (and thus making C processed) Invariant 1 is satisfied. If after adding S to P the number of loose-ends increases by k we will also call S a k-semi-support (k may be negative).

Note that the only difference between a semi-support and a support is that after adding a semi-support to P we may get a non-Hamiltonian cycle in P. The following lemma, similar to the Kostochka-Serdyukov technique, will be used to convert a semi-support M to a support S without losing much weight. The weight loss in this process depends on

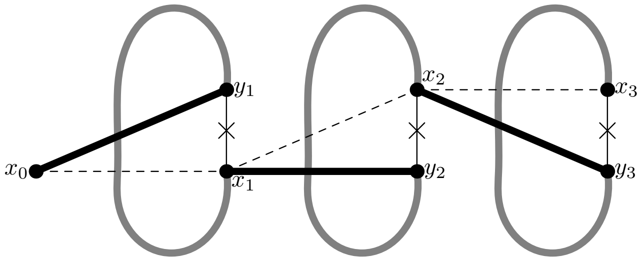

Figure 1: Breaking the cycles in the proof of Lemma 3.1. Dashed edges are lighter than the corresponding solid edges. Crossed-out edges are the edges removed from the cycles.

how the weight of M is distributed between allowed and forbidden edges, on the weight of allowed edges of M that belong to cycles in $P \cup M$, etc.

Lemma 3.1. *Consider any phase of the algorithm and let C be an unprocessed cycle. Let M be a k-semi-support of C. Assume there is a vertex $x_0 \notin V(M)$, such that x_0 is a loose-end or $x_0 \in V(C)$. Moreover, assume $P \cup M$ contains cycles (possibly of length 2) $C_1, \ldots, C_q$. For each i, $1 \leq i \leq q$, let e_i be any edge in $M \cap C_i$. Let $Q = \{e_1, \ldots, e_q\}$ and let $D = \bigcup_i C_i$. Finally, let us partition edges in M into two sets: F containing forbidden edges, and A containing allowed edges.*

Then one can find S, a k-support of C, such that

(i) $w(S) \geq w(M \setminus Q) + \frac{1}{2}w(Q)$,

(ii) $w(S) \geq w(A \setminus D) + \frac{3}{4}w(A \cap D) + \frac{1}{2}w(F)$.

Proof. Denote the ends of e_1 by x_1 and y_1 in such a way that x_0y_1 is heavier than x_0x_1. Note that $w(x_0y_1) = \max\{w(x_0x_1), w(x_0y_1)\} \geq \frac{1}{2}(w(x_0x_1) + w(x_0y_1)) \geq \frac{1}{2}w(e_1)$, where the last step follows from the triangle inequality. Moreover, by replacing e_1 by x_0y_1 we break the cycle C_1 and x_1 becomes a loose-end. We can proceed in this way for all cycles, i.e., for every $i = 1, \ldots, q$ the ends of e_i are labelled x_i and y_i so that

$$w(x_{i-1}y_i) \geq \tfrac{1}{2}w(e_i). \tag{1}$$

Let $S = M \setminus \{e_i \mid i = 1, \ldots, q\} \cup \{x_{i-1}y_i \mid i = 1, \ldots, q\}$. Clearly, $P \cup S$ does not contain cycles hence it is sub-Hamiltonian. Also, observe that there are only 2 vertices, namely x_0 and x_q whose degrees differ in graphs $P \cup M$ and $P \cup S$. Since $\deg_{P \cup S} x_0 = 2$ and $\deg_{P \cup S} x_q = 1$, after adding S to P (and thus processing C) Invariant 1 is still satisfied, and so S is a support. Also note that x_0 is a loose-end in $P \cup M$ and it is not a loose-end in $P \cup S$, while x_q is not a loose-end in $P \cup M$ and it is a loose-end in $P \cup S$. It follows that S is a k-support.

Now let us bound the weight of S. By (1), $w(S) \geq w(M \setminus Q) + \frac{1}{2}w(Q)$, which is claim (i). To prove (ii), in each cycle C_i we choose the lightest edge e_i in $M \cap C_i$ and we assume

Q consists of these edges. Notice that $F \subseteq Q$ (by Observation 1) and also $A \setminus D \subseteq M \setminus Q$, so by (i) we have,

$$w(S) \geq w(M \setminus Q) + \frac{1}{2}w(Q) \geq w(A \setminus D) + w((A \cap D) \setminus Q) + \tfrac{1}{2}w(A \cap Q) + \tfrac{1}{2}w(F). \quad (2)$$

By Observation 1, and since Q consists of the lightest edges in cycles, $w((A \cap D) \setminus Q) \geq \frac{1}{2}w(A \cap D)$. Then $w((A \cap D) \setminus Q) + \frac{1}{2}w(A \cap Q) = w((A \cap D) \setminus Q) + \frac{1}{2}w((A \cap D) \cap Q) = \frac{1}{2}w((A \cap D) \setminus Q) + \frac{1}{2}w(A \cap D) \geq \frac{3}{4}w(A \cap D)$. By plugging it into (2) we get (ii). □

3.1 Even cycles

Lemma 3.2. *Let C be an unprocessed 4-cycle and assume that there is at least one loose-end. Then there is a 0-support of C of weight $\geq (\frac{1}{4}, \frac{1}{2}) \star C$.*

Proof. We consider two cases:

Case 1 $E(C)$ has at most one forbidden edge. We partition $E(C)$ into two matchings, M_1 and M_2. W.l.o.g. assume M_1 does not contain forbidden edges. Let S_1 and S_2 be the supports corresponding to M_1 and M_2 by Lemma 3.1 and let S be the heavier of them. Following the notation from Lemma 3.1, define A_1, A_2 (F_1, F_2) as the sets of allowed (resp. forbidden) edges of M_1, M_2. Let D_1, D_2 be the sets of edges of $E(C)$ that belong to cycles in $P \cup M_1$ or $P \cup M_2$ respectively. Also let $A = A_1 \cup A_2$, $F = F_1 \cup F_2$ and $D = D_1 \cup D_2$.

Notice that by inequality (ii) of Lemma 3.1 applied to M_i, $i = 1, 2$ we get $w(S_i) \geq w(A_i \setminus D_i) + \frac{3}{4}w(A_i \cap D_i) + \frac{1}{2}w(F_i)$. Summing up the two inequalities yields

$$w(S) \geq \tfrac{1}{2}(w(S_1) + w(S_2)) \geq \tfrac{1}{2}w(A \setminus D) + \tfrac{3}{8}w(A \cap D) + \tfrac{1}{4}w(F). \quad (3)$$

Let us first assume that $P \cup M_1$ contains a cycle $\tilde{C}$. By Observation 1 both allowed edges of M_1 are in $\tilde{C}$. So either both chords of C are forbidden or both edges of M_2 are. Since we assumed that $E(C)$ has at most one forbidden edge, it is the chords of C that are forbidden. It now follows from Observation 2 that both edges of M_2 are allowed, so $A = C$. From (3) we get $w(S) \geq \frac{3}{8}w(A) = \frac{3}{8}w(C) \geq (\frac{1}{4}, \frac{1}{2}) \star C$.

Hence, we may assume that $P \cup M_1$ contains no cycle. It follows that $D_1 = \emptyset$, so $|A \setminus D| \geq 2$. From (3) we get $w(S) \geq \frac{1}{2}w(A \setminus D) + \frac{3}{8}w(A \cap D) + \frac{1}{4}w(F) \geq \frac{1}{4}(w(A \setminus D) + w(A \cap D) + w(F)) + \frac{1}{4}w(A \setminus D) \geq \frac{1}{4}w(C) + \frac{1}{4}w(A \setminus D) \geq (\frac{1}{4}, \frac{1}{2}) \star C$, where the last inequality follows from $|A \setminus D| \geq 2$.

Case 2 $E(C)$ has two forbidden edges. Denote the vertices of C by $v_1, \ldots, v_4$ in the order they appear on C and assume w.l.o.g. that v_1v_2 and v_3v_4 are forbidden. Let u be a loose-end. Consider four edge sets $S_1 = \{uv_1, v_2v_3\}$, $S_2 = \{uv_2, v_1v_4\}$, $S_3 = \{uv_4, v_2v_3\}$, and $S_4 = \{uv_3, v_1v_4\}$. Note that these sets are allowed since for any i, edges of S_i belong to a single path in $P \cup S_i$ (ending in v_4, v_3, v_1 and v_2 respectively). It follows that all S_i are supports and we choose S, the heaviest of them. Then $w(S) \geq \frac{1}{4}\sum_{i=1}^{4} w(S_i) \geq \frac{1}{4}[2w(v_2v_3) + 2w(v_1v_4) + (w(uv_1) + w(uv_2)) + (w(uv_3) + w(uv_4))] \geq \frac{1}{4}[2w(v_2v_3) + 2w(v_1v_4) + w(v_1v_2) + w(v_3v_4)]$, where the last step follows from triangle inequality. Hence $w(S) \geq \frac{1}{4}w(C) + \frac{1}{4}[w(v_2v_3) + w(v_1v_4)] \geq (\frac{1}{4}, \frac{1}{2}) \star C$. □

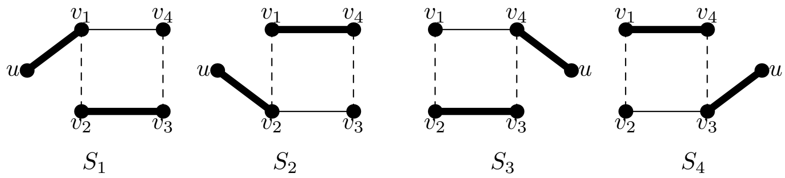

Figure 2: Supports in Case 2 of the proof of Lemma 3.2

Lemma 3.3. *Let C be an unprocessed even-length cycle, $|C| \geq 6$, and assume that there is at least one loose-end. Then there is a 0-support of C of weight at least $(\frac{1}{4}, \frac{1}{2}) \star C$.*

Proof. We partition $E(C)$ into two matchings, M_1 and M_2, let S_1 and S_2 be the supports corresponding to M_1 and M_2 by Lemma 3.1, and let S be the heavier of these supports. We follow all the definitions from the beginning of the proof of the previous lemma to obtain inequality (3).

From that inequality we get $w(S) \geq \frac{3}{8}w(A) + \frac{1}{4}w(F) = \frac{1}{4}w(C) + \frac{1}{8}w(A)$. It follows that $w(S) \geq (\frac{1}{4}, \frac{1}{2}) \star C$ if $|A| \geq 4$.

Since by Observation 2 we have $|A| \geq |C|/2$, the only case we need to consider is that of $|C| = 6$ and $|A| = 3$. W.l.o.g. $M_1 = A$ and $M_2 = F$. Let Q bet the set of the lightest edges from each cycle in $P \cup M_1$ or $P \cup M_2$, one edge from each cycle. There is precisely one such cycle in $P \cup M_1$, since by Observation 1 each such cycle has to contain at least two edges. It follows that $|A \setminus Q| \geq 2$. By inequality (i) in Lemma 3.1 we get $w(S) \geq \frac{1}{2}(w(S_1)+w(S_2)) \geq \frac{1}{2}w(E(C) \setminus Q) + \frac{1}{4}w(Q) = \frac{1}{4}w(E(C) \setminus Q) + \frac{1}{4}w(C) = \frac{1}{4}w(A \setminus Q) + \frac{1}{4}w(C) \geq (\frac{1}{4}, \frac{1}{2}) \star C$, as required. □

3.2 Triangles

For any cycle C, by $\max(C)$ we denote the heaviest edge in C.

Lemma 3.4. *For any unprocessed triangle C, there is a $(+1)$-support of C of weight at least $(\frac{1}{4}, \frac{1}{2}) \star C - \frac{1}{4}w(\max(C))$.*

Proof. Let x, y, z be the vertices of C and assume w.l.o.g. that both xz and yz are allowed. Let S consist of the heavier of the edges xz, yz. Clearly, S is a support and $w(S) \geq \frac{1}{2}(w(xz) + w(yz)) \geq \frac{1}{4}w(C) + \frac{1}{4}(w(xz) + w(yz)) - \frac{1}{4}w(xy) \geq (\frac{1}{4}, \frac{1}{2}) \star C - \frac{1}{4}w(xy) \geq (\frac{1}{4}, \frac{1}{2}) \star C - \frac{1}{4}w(\max(C))$. □

Lemma 3.5. *Let C be an unprocessed triangle and assume that there are two loose-ends. Then there is a (-1)-support of C of weight at least $(\frac{1}{4}, \frac{1}{2}) \star C + \frac{1}{4}w(\max(C))$.*

Proof. Let x, y, z bet the vertices of C and let u and v be the loose-ends. We consider 2 cases:

Case 1 Both loose-ends are connected to C by paths in P, say u is connected to x and v to y. Note that in this case all edges of C are allowed. Let $S_1 = \{xy, zv\}$ and $S_2 = \{zy, xv\}$.

Note that after adding any of these sets to P, both added edges lie on a single path that ends in u (see Figure 3), so P remains sub-Hamiltonian. Hence both S_1 and S_2 are supports of C. The heavier of them has weight $\max\{w(xy) + w(zv), w(zy) + w(xv)\} \geq \frac{1}{2}(w(xy)+w(zy)+w(zv)+w(xv)) \geq \frac{1}{2}(w(xy)+w(zy)+w(xz)) \geq \frac{1}{4}w(C)+\frac{1}{2}w(\min(C))+\frac{1}{4}w(\max(C)) = (\frac{1}{4}, \frac{1}{2}) \star C + \frac{1}{4}w(\max(C))$.

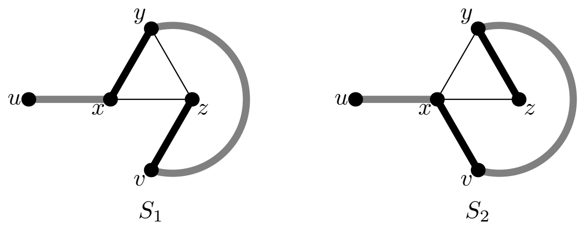

Figure 3: Supports in Case 1 of the proof of Lemma 3.5. Gray lines denote the paths connecting loose-ends with C.

Case 2 At least one loose-end, say u, is not connected to C by a path in P. W.l.o.g. assume that both xz are yz allowed. Let $S_1 = \{xz, yu\}$ and $S_2 = \{yz, xu\}$. Note that adding S_1 to P does not create a cycle. Indeed, yu does not belong to a cycle because yu belongs to a path that ends in a vertex different from x, y or z. Also xz does not belong to a cycle because it was allowed before adding it to P. Similar reasoning shows that adding S_2 to P does not create a cycle. Hence both S_1 and S_2 are supports. Similarly to the previous case we get $\max\{w(S_1), w(S_2)\} \geq \frac{1}{2}(w(xz) + w(yu) + w(yz) + w(xu)) \geq (\frac{1}{4}, \frac{1}{2}) \star C + \frac{1}{4}w(\max(C))$. □

Observation 3. Let C be an unprocessed odd cycle in the last phase of the algorithm and assume that there is exactly one loose-end u. Then u is connected by a path in P to a vertex $z \in C$ and $V(C)$ induces exactly $\lfloor |E(C)|/2 \rfloor$ forbidden edges. These edges can be either edges or chords of C, and none of them is adjacent to z. □

Lemma 3.6. *Let C be an unprocessed triangle in the last phase of the algorithm and assume that there is exactly one loose-end u. Then there is a (-1)-support of C of weight at least $(\frac{1}{4}, \frac{1}{2}) \star C + \frac{1}{4}w(\max(C))$.*

Proof. Let x, y, z denote the vertices of C. By Observation 3 cycle C contains a forbidden edge — assume w.l.o.g. it is xy — and u is connected in P by a path to z. Let $S_1 = \{xz, yu\}$ and $S_2 = \{yz, xu\}$. Clearly, xz and yu are in the same cycle in $P \cup S_1$ and it is a Hamiltonian cycle. Hence, S_1 is a support of C, and similarly S_2. We pick the heavier of these cycles (its weight can be estimated similarly as in the proof of Lemma 3.5). □

3.3 5-cycles

Lemma 3.7. *Let C be an unprocessed 5-cycle with at most one forbidden edge. Then there is a $(+1)$-support of weight at least $(1/4, 1/2) \star C$.*

Proof. Let $v_1, \ldots, v_5$ be the vertices of C in the order they appear on C and assume w.l.o.g. that v_1v_5 is the lightest edge in $E(C)$.

Let $M_1 = \{v_1v_2, v_3v_4\}$ and $M_2 = \{v_2v_3, v_4v_5\}$. Let S_1 and S_2 be the supports corresponding to M_1 and M_2 by Lemma 3.1 and let S be the heavier of them. Also, assume all definitions leading to inequality (3) in the proof of Lemma 3.2.

We consider three cases:

Case 1 v_1v_5 is forbidden. Then v_1v_2 belongs to a path in $P \cup M_1$ (ending in v_5), hence $v_1v_2 \notin D$. By Observation 1, then also $v_3v_4 \notin D$, so $M_1 \cap D = \emptyset$. By symmetry, also $M_2 \cap D = \emptyset$. Hence $A \setminus D = A$. By inequality (ii) in Lemma 3.1 we get $w(S) \geq \frac{1}{2}(w(S_1) + w(S_2)) \geq \frac{1}{2}w(A) \geq \frac{1}{2} \cdot \frac{4}{5}w(C) = \frac{2}{5}w(C) \geq \frac{1}{4}w(C) + \frac{3}{4}\min(C) \geq (\frac{1}{4}, \frac{1}{2}) \star C$.

Case 2 One of the matchings, say M_1, contains a forbidden edge. We have two subcases depending on which edge of M_1 is forbidden.

Case 2a If v_1v_2 is forbidden, then the other edge of M_1, i.e. v_3v_4, is allowed and by Observation 1 it does not belong to D. Also, v_2v_3 does not belong to D, because it lies on a path that ends in v_1. Again, by Observation 1, v_4v_5 does not belong to D. Altogether, this gives $|A \setminus D| \geq 3$.

Using inequality (3) we get $w(S) \geq \frac{1}{2}w(A \setminus D) + \frac{3}{8}w(A \cap D) + \frac{1}{4}w(F) \geq \frac{1}{4}w(C \setminus \{v_1v_5\}) + \frac{1}{4}w(A \setminus D) + \frac{1}{8}w(A \cap D) \geq \frac{1}{4}w(C \setminus \{v_1v_5\}) + \frac{1}{2}w(v_1v_5) = (\frac{1}{4}, \frac{1}{2}) \star C$.

Case 2b If v_3v_4 is forbidden, then each of the following four sets of edges is a (+1)-support: $S_1 = \{v_1v_2, v_4v_5\}$, $S_2 = \{v_1v_3, v_4v_5\}$, $S_3 = \{v_1v_4, v_2v_3\}$, $S_4 = \{v_1v_5, v_2v_3\}$. Their total weight is

$$\Big(w(v_1v_2) + w(v_2v_3) + w(v_3v_1) + w(v_1v_4) + w(v_4v_5) + w(v_5v_1)\Big) + \Big(w(v_2v_3) + w(v_4v_5)\Big)$$

Using the triangle inequality to bound the first part of this expression, and the fact that v_1v_5 is the lightest edge of C to bound the second, we get

$$\sum_{i=1}^{4} w(S_i) \geq w(C) + 2w(v_1v_5),$$

so the heaviest of S_i has weight at least $(\frac{1}{4}, \frac{1}{2}) \star C$.

Case 3 There are no forbidden edges in $E(C)$. Suppose $P \cup M_1$ contains a cycle. Then the chords v_1v_3 and v_2v_4 are forbidden. It follows that the edges of M_2 belong to a path in $P \cup M_2$ (one ending in v_1), so they cannot lie on a cycle in $P \cup M_2$. We conclude that at least one of $P \cup M_1$ and $P \cup M_2$ does not contain cycles, and so $|A \setminus D| \geq 2$. Using inequality (3) we get $w(S) \geq \frac{1}{2}w(A \setminus D) + \frac{3}{8}w(A \cap D) = \frac{3}{8}w(A) + \frac{1}{8}w(A \setminus D) \geq \frac{3}{8} \cdot \frac{4}{5}w(C) + \frac{1}{4}\min(C) = \frac{1}{4}w(C) + \frac{1}{20}w(C) + \frac{1}{4}\min(C) \geq (\frac{1}{4}, \frac{1}{2}) \star C$. □

Lemma 3.8. *Let C be an unprocessed 5-cycle with two forbidden edges. Let e be any of the two forbidden edges of C. Then there is a (+1)-support of C of weight at least $(\frac{1}{4}, \frac{1}{2}) \star C - \frac{1}{4}w(e)$.*

Proof. Let $v_1, \ldots, v_5$ be the vertices of C in the order they appear on C and assume w.l.o.g. that v_1v_5 and v_2v_3 the forbidden edges of C and $e = v_1v_5$. Let $M_1 = \{v_1v_2, v_3v_4\}$ and $M_2 = \{v_2v_3, v_4v_5\}$ and assume the notation from the proof of the previous lemma.

Note that the edges of M_1 belong to a path in $P \cup M_1$ ending in v_5, hence $M_1 \cap D = \emptyset$. It follows that $|A \setminus D| \geq 2$. Using inequality (3) we get $w(S) \geq \frac{1}{2}w(A \setminus D) + \frac{3}{8}w(A \cap D) + \frac{1}{4}w(F) \geq \frac{1}{4}(w(A \setminus D) + w(A \cap D) + w(F)) + \frac{1}{4}w(A \setminus D) = \frac{1}{4}w(C \setminus \{e\}) + \frac{1}{4}w(A \setminus D) \geq \frac{1}{4}w(C \setminus \{e\}) + \frac{1}{2}\min(C) = (\frac{1}{4}, \frac{1}{2}) \star C - \frac{1}{4}w(e)$. □

Lemma 3.9. *Let C be an unprocessed 5-cycle with two forbidden edges and assume that there are two loose-ends. Let e denote any of the two forbidden edges of C. Then there is a (-1)-support of C of weight at least $(\frac{1}{4}, \frac{1}{2}) \star C + \frac{1}{4}w(e)$.*

Proof. Label the vertices of C as in the proof of the previous lemma. Observe that since there are at least two loose-ends, at least one of them, call it u, is not connected by a path to C in P.

Let $M_1 = \{v_1v_2, v_3v_4, v_5u\}$ and $M_2 = \{uv_1, v_2v_3, v_4v_5\}$, let S_1 and S_2 be the supports corresponding to M_1 and M_2 by Lemma 3.1, and let S be the heavier of them.

Note that the edges of M_1 belong to a path in $P \cup M_1$ (the one ending in u), hence $P \cup M_1$ does not contain cycles and we have $S_1 = M_1$. Also, neither uv_1 nor v_4v_5 belong to a cycle in $P \cup M_2$. Of course v_2v_3 belongs to a cycle in $P \cup M_2$.

By inequality (i) in Lemma 3.1 we get $w(S) \geq \frac{1}{2}(w(S_1) + w(S_2)) \geq \frac{1}{2}[w(v_1v_2) + w(v_3v_4) + w(v_5u) + w(uv_1) + w(v_4v_5)] + \frac{1}{4}w(v_2v_3)$. Using the triangle inequality gives $w(S) \geq \frac{1}{2}[w(v_1v_2) + w(v_3v_4) + w(v_1v_5) + w(v_4v_5)] + \frac{1}{4}w(v_2v_3) \geq \frac{1}{4}w(C) + \frac{3}{4}\min(C) + \frac{1}{4}w(v_1v_5) \geq (\frac{1}{4}, \frac{1}{2}) \star C + \frac{1}{4}w(e)$. □

Lemma 3.10. *Let C be an unprocessed 5-cycle with two forbidden edges in the last phase of the algorithm and assume that there is exactly one loose-end u. Let e be any of the two forbidden edges of $E(C)$. Then there is a (-1)-support of C of weight at least $(\frac{1}{4}, \frac{1}{2}) \star C + \frac{1}{4}w(e)$.*

Proof. Label the vertices of C as in Lemma 3.8. By Observation 3, u is connected in P to v_4 by a path.

Let $S_1 = \{v_1v_2, v_3v_4, v_5u\}$, $S_2 = \{uv_1, v_2v_4, v_3v_5\}$ and $S_3 = \{uv_1, v_2v_5, v_3v_4\}$. One may check that for any $i = 1, 2, 3$, S_i is a support and in particular $P \cup S_i$ is a Hamiltonian cycle. Let S be the heaviest of these supports.

Denote $w(v_2v_4) + w(v_3v_5) + w(v_2v_5) + w(v_3v_4)$ by X. Then $w(S) \geq \frac{1}{2}w(S_1) + \frac{1}{4}w(S_2) + \frac{1}{4}w(S_3) = \frac{1}{2}(w(v_1v_2) + w(v_3v_4) + w(v_5u) + w(uv_1)) + \frac{1}{4}X$.

By triangle inequality (used twice), $X \geq 2w(v_2v_3)$. By symmetry, $X \geq 2w(v_4v_5)$. Hence, $X \geq w(v_2v_3) + w(v_4v_5)$. Let us apply triangle inequality one more time: $w(v_5u) + w(uv_1) \geq w(v_1v_5)$.

Putting it all together we get $w(S) \geq \frac{1}{2}(w(v_1v_2) + w(v_3v_4) + w(v_1v_5)) + \frac{1}{4}(w(v_2v_3) + w(v_4v_5)) \geq (\frac{1}{4}, \frac{1}{2}) \star C + \frac{1}{4}w(e)$. □

3.4 Odd cycles of length at least 7

Lemma 3.11. *Let C be an unprocessed odd cycle of length at least 7. Then there is a $(+1)$-support of weight at least $(\frac{1}{4}, \frac{1}{2}) \star C$.*

Proof. Let $|C| = 2k+1$, $k \geq 3$. We enumerate vertices in $V(C)$ so that $C = v_0v_1v_2 \dots v_{2k-1}v_{2k}v_0$, both v_0v_1 and v_0v_{2k} are allowed and $w(v_0v_1) \geq w(v_0v_{2k})$. Consider two subsets of $E(C)$: $M_1 = \{v_{2i}v_{2i+1} \mid 0 \leq i \leq k-1\}$ and $M_2 = \{v_{2i+1}v_{2i+2} \mid 0 \leq i \leq k-1\}$. In other words we partition $E(C) \setminus \{v_0v_{2k}\}$ into two matchings.

Let $C_1, \dots, C_p$ be all cycles in $P \cup M_1$ and Let $C_{p+1}, \dots, C_q$ be all cycles in $P \cup M_2$. Similarly as in Lemma 3.1, let $D = \bigcup_{i=1}^{q} C_i$ and we partition edges in $M_1 \cup M_2$ into two sets: F containing forbidden edges, and A containing allowed edges. Further, let us choose for each cycle C_i, $i = 1, \dots, q$, some edge e_i in $C_i \cap E(C)$ and let $Q = \{e_1, \dots, e_q\}$. Since by Observation 1 each cycle C_i that contains v_0v_1 contains also another edge from A, we assume w.l.o.g. that $v_0v_1 \notin Q$.

Using Lemma 3.1 we obtain supports S_1, S_2. Let S be the heavier of these supports. Then $w(S) \geq \frac{1}{2}(w(S_1) + w(S_2))$.

By inequality (i) in Lemma 3.1, $w(S) \geq \frac{1}{2}w((M_1 \cup M_2) \setminus Q) + \frac{1}{4}w(Q) = \frac{1}{4}w(E(C) \setminus \{v_0v_{2k}\}) + \frac{1}{4}w((M_1 \cup M_2) \setminus Q)$. Since $v_0v_1 \notin Q$ and $w(v_0v_1) \geq w(v_0v_{2k})$, $w(S) \geq \frac{1}{4}w(E(C)) + \frac{1}{4}w((M_1 \cup M_2) \setminus (Q \cup \{v_0v_1\}))$. As $F \subseteq Q$, $(M_1 \cup M_2) \setminus (Q \cup \{v_0v_1\}) = (A \setminus \{v_0v_1\}) \setminus Q$ and hence

$$w(S) \geq \tfrac{1}{4}w(E(C)) + \tfrac{1}{4}w((A \setminus \{v_0v_1\}) \setminus Q). \tag{4}$$

It follows that $|(A \setminus \{v_0v_1\}) \setminus Q| \geq 2$ implies $w(S) \geq (1/4, 1/2) \star C$.

First assume there are k forbidden edges in $E(C)$. Then one of the matchings, say M_1, contains only allowed edges (and the other matching contains all the forbidden edges of C). Note that in $P \cup M_1$ all edges of M_1 belong to a path with one end in v_{2k}. It follows that $M_1 = S_1$ and $S_1 \cap Q = \emptyset$. It follows that $A \cap Q = \emptyset$ and hence $(A \setminus \{v_0v_1\}) \setminus Q$ contains at least $k - 1 \geq 2$ edges, as required.

Now assume there are at most $k-1$ forbidden edges in $E(C)$. Then $|A| \geq k+1$. By Observation 1, $|A \setminus Q| \geq \lceil \frac{|A|}{2} \rceil$. It follows that $|(A \setminus \{v_0v_1\}) \setminus Q| \geq \lceil \frac{|A|}{2} \rceil - 1$. For $|A| \geq 5$, we get $\lceil \frac{|A|}{2} \rceil - 1 \geq 2$.

Hence we are left with the case $|A| \leq 4$. Since $|A| \geq k+1$, $k \leq 3$. So $k = 3$, $|A| = 4$ and $|F| = 2$. We consider two subcases.

Case 1. v_5v_6 is forbidden. Then v_4v_5 is allowed and after adding the matching containing v_4v_5 to P, v_4v_5 is on a path ending in v_6, hence v_4v_5 does not belong to any C_i. Hence the three remainig edges in A belong at most one cycle C_i, so $|A \cap Q| \leq 1$ and further $|(A \setminus \{v_0v_1\}) \setminus Q| \geq 2$, as required.

Case 2. v_5v_6 is allowed. If $F = \{v_2v_3, v_4v_5\}$, one of the matchings, namely M_2, contains only allowed edges. Moreover, these edges belong to a path in $P \cup M_2$ (ending in v_6), so $M_2 = S_2$ and $S_2 \cap Q = \emptyset$. There is just one allowed edge in M_1 and hence it cannot belong to a cycle C_i. It follows that $Q = F$ and hence $|(A \setminus \{v_0v_1\}) \setminus Q| \geq 3$. The case $F = \{v_1v_2, v_3v_4\}$ is symmetric. Finally, assume $F = \{v_1v_2, v_4v_5\}$. By Observation 1, in

$P \cup M_1$ and $P \cup M_2$ there are at most 2 cycles with edges from A. If $P \cup M_1$ contains such cycle, then v_0v_3 is forbidden. However, then $P \cup M_2$ contains no such cycle. Hence $|A \cap Q| \leq 1$ and $|(A \setminus \{v_0v_1\}) \setminus Q| \geq 2$, as required. □

4 Ordering the cycles

4.1 Basic setup

Based on the results from the previous section, we can see that every cycle C belongs to one of three categories:

even cycles: C has a 0-support of weight $(\frac{1}{4}, \frac{1}{2}) \star C$, if there exists at least one loose-end,

good odd cycles: C has a $(+1)$-support of weight at least $(\frac{1}{4}, \frac{1}{2}) \star C$ — that is the case if C is an odd cycle of length ≥ 7 or a 5-cycle with at most one forbidden edge,

bad odd cycles: C has a $(+1)$-support of weight smaller than $(\frac{1}{4}, \frac{1}{2}) \star C$, and it also has a (-1)-support of weight greater than $(\frac{1}{4}, \frac{1}{2}) \star C$, but only if there exist at least two loose-ends or it is the last cycle processed — that is the case for all 3-cycles and for 5-cycles with two forbidden edges.

Remark 4.1. Notice that a good odd cycle might become bad when other cycles are processed, if it is initially a 5-cycle with zero (or one) forbidden edges and two (one, resp.) of its allowed edges becomes forbidden. However, a bad odd cycle can never become a good one.

We say that a cycle C is k-processed, if it is processed using a k-support. The general order of processing the cycles consists of 4 stages:

(1) as long as there exists a good odd cycle, $(+1)$-process it,

(2) $(+1)$-process bad odd cycles until the number of loose-ends is greater or equal to the number of remaining bad odd cycles,

(3) 0-process even cycles,

(4) (-1)-process the remaining odd cycles.

When we use the above processing order all the assumptions of previous section's lemmas are satisfied. In particular in stage 3, there exists at least one loose-end, so we can process the even cycles. This is because we can assume that $\mathcal{C}$ contains at least one triange, otherwise already the Kostochka-Serdyukov algorithm gives 7/8-approximation.

It is clear that we are getting enough weight from cycles processed in stages 1 and 3. We also loose some extra weight in stage 2 and gain weight in stage 4. We want to select the cycles to be processed in stage 2 in such a way that the overall weight of edges added during stages 2 and 4 is at least $\sum_i (\frac{1}{4}, \frac{1}{2}) \star C_i$, where the sum is over all cycles processed in these stages.

4.2 Ordering bad odd cycles

Let us first define certain useful notions. For any bad odd cycle C, let $B_{-1}(C)$ ($B_{+1}(C)$) be the lower bound on the weight of the (-1)-support ($(+1)$-support), as guaranteed by the appropriate lemma in the previous section. Suppose that $\mathcal{C}_i$ is the set of bad odd cycles processed in stage i, $i = 2, 4$. If we use previous section's lemmas to lowerbound the weight of all edges added in stages 2 and 4, we are going to get

$$\sum_{C \in \mathcal{C}_2} B_{+1}(C) + \sum_{C \in \mathcal{C}_4} B_{-1}(C),$$

and we need to show that $\mathcal{C}_2$ and $\mathcal{C}_4$ can be chosen so that the value of this expression is at least

$$\sum_{C \in \mathcal{C}_2 \cup \mathcal{C}_4} (\tfrac{1}{4}, \tfrac{1}{2}) \star C.$$

For every bad odd cycle C there exists a non-negative number, which we call the *loose-end value for* C and denote $\mathrm{LEV}(C)$ such that

$$B_{+1}(C) \geq (\tfrac{1}{4}, \tfrac{1}{2}) \star C - \mathrm{LEV}(C) \quad \text{and} \quad B_{-1}(C) \geq (\tfrac{1}{4}, \tfrac{1}{2}) \star C + \mathrm{LEV}(C).$$

Note, that this number is equal to $\frac{1}{4}w(e)$, where e is the heaviest edge of C if C is a triangle, or the heavier of the two forbidden edges of C if C is a bad 5-cycle.

The reason why we call this number the loose-end value for C is that it is essentially the price at which C should be willing to buy/sell a loose-end. In this economic analogy, the cycles that are $(+1)$-processed are selling loose-ends to the cycles that are (-1)-processed. If we can make every cycle trade a loose-end at a preferred price (LEV or better), the weight of a support of any cycle C together with its profit/loss coming from trading a loose-end adds up to at least $(\frac{1}{4}, \frac{1}{2}) \star C$. But it is obvious how to make every cycle trade a loose-end at a preferred price! It is enough to make the cycles with smallest LEV sell loose-ends (process them in stage 2), and make the remaining cycles buy loose-ends (process them in stage 4).

Note here, that some bad odd cycles will get loose-ends for free from good odd cycles processed in stage 1. Since we assume that the total number of vertices in the graph is even, the number of the remaining bad odd cycles is also even, and so they can be divided evenly into sellers and buyers.

Using Lemma 2.1 we get

Theorem 4.2. *Metric MAX-TSP problem can be* $7/8$*-approximated for graphs with even number of vertices.*

This can be extended to graphs with odd number of vertices, at a cost of increasing the running time by a factor of $O(n^4)$, see the next section.

5 Processing graphs with odd number of vertices

When the input graph has an odd number of vertices the algorithm described before does not work because there is no perfect matching. It is easy to see that when we use a maximum weight near-perfect matching instead (i.e. such that exactly one vertex is not matched) our algorithm gives $(7/8 - \frac{1}{4n})$-approximation, which is already better than the best known previous results. Luckily, even for the odd case we can still retain 7/8-approximation by applying our algorithm in a more sophisticated way.

The modified algorithm for the odd case also begins with a cycle cover $\mathcal{C}$ and a maximum weight matching M. Since the edge weights are nonnegative, we can assume that there is precisely one unmatched vertex v. Our new algorithm processes cycles of $\mathcal{C}$ as before, but the cycle C^* that contains v is processed in a special way. We show that this algorithm returns a Hamiltonian cycle of weight at least $\frac{7}{8}\mathrm{OPT}$, provided that the initial cycle cover $\mathcal{C}$ and the matching M satisfy certain special conditions. We show that such a pair of a matching and a cover is contained in a set of $O(n^4)$ pairs which can be constructed in polynomial time. For each of these pairs we apply the modified algorithm and we return the heaviest of the Hamiltonian cycles found.

5.1 Finding a special pair of cycle cover and matching

Now we are going to describe the aforementioned set of $O(n^4)$ matching-cover pairs. In what follows we assume that the graph contains at least 4 vertices (otherwise the problem can be solved exactly in $O(1)$ time). A simple path $vxyz$ will be called a *candidate path* when $w(xy) \geq w(vx)$ and $w(xy) \geq w(yz)$. For each candidate path p we find $\mathcal{C}_p$, the maximum weight cycle cover containing path p. (Such a cover can be found by finding a maximum weight cycle cover in a modified graph, i.e. with weights of edges on path p very large). Similarly, for each candidate path $p = vxyz$ we find M_x^p, the maximum weight matching in $G - \{x\}$ that contains edge yz (again, we make the weight of edge yz very large and we find the maximum weight matching). Next, for each candidate path $p = vxyz$ we find M_y^p, the maximum weight matching in $G - \{y\}$ that contains edge vx. Note that

Proposition 5.1. *For any candidate path $p = vxyz$,*

(a1) $\mathcal{C}_p$ *contains a cycle of length at least 4 containing edge xy,*

(a2) matching M_x^p contains yz and matching M_y^p contains vx, and

(a3) $w(xy) \geq w(vx)$ *and* $w(xy) \geq w(yz)$.

Proposition 5.2. *For some candidate path $p = vxyz$ we have*

(b1) $w(\mathcal{C}_p) \geq \mathrm{OPT}$*, and*

(b2) $w(M_a^p) + \frac{1}{2}w(xy) \geq \frac{1}{2}\mathrm{OPT}$ *where* $a \in \{x, y\}$

Proof. Let H be a maximum weight Hamiltonian cycle. Let xy be the heaviest edge on H and let vx and yz be the two edges incident with xy in H. Condition (b1) is obvious then. Let M_x and M_y be the the near perfect matching that leaves x (resp. y) unmatched and consists of edges of H only. Note that $w(M_x^p) \geq w(M_x)$ and $w(M_y^p) \geq w(M_y)$. Clearly $w(M_x) + w(M_y) + w(xy) = \text{OPT}$. It follows that $w(M_x^p) + w(M_y^p) + w(xy) \geq \text{OPT}$ and hence $\max\{w(M_x^p) + \frac{1}{2}w(xy), w(M_y^p) + \frac{1}{2}w(xy)\} \geq \frac{1}{2}[w(M_x^p) + w(M_y^p) + w(xy)] \geq \frac{1}{2}\text{OPT}$, which is equivalent to (b2). □

In what follows let $\mathcal{C}$ and M denote a cover and a matching satisfying conditions (a1)–(a3) and (b1)–(b2) and let $p = vxyz$ be the corresponding candidate path. Let C^* be the cycle of length at least 4 in $\mathcal{C}$ that contains xy and assume w.l.o.g. that x is unmatched in M and $yz \in M$.

Now we can prove an analog of Lemma 2.1.

Lemma 5.3. *If during processing the cycles in $\mathcal{C}$, we can add edges of total weight at least $[\sum_{C_i \in \mathcal{C} \setminus \{C^*\}} (\frac{1}{4}, \frac{1}{2}) \star C_i] + [(\frac{1}{4}, \frac{1}{2}) \star C^* + \frac{1}{2}w(xy)]$ to M, then a Hamiltonian cycle of weight at least $\frac{7}{8}\text{OPT}$ is returned.*

Proof. The sum of the weights of the two Hamiltonian cycles found by the algorithm is at least $w(\mathcal{C}) - \sum_{C \in \mathcal{C}} w(\min(C)/2 + w(M) + \frac{1}{4}w(\mathcal{C}) + \sum_{C \in \mathcal{C}} w(\min(C))/2 + \frac{1}{2}w(xy) = \frac{5}{4}w(\mathcal{C}) + w(M) + \frac{1}{2}w(xy)$. By (b1) and (b2) this is at least $\frac{7}{4}\text{OPT}$, so the better of the two solutions is a $\frac{7}{8}$-approximation. □

5.2 Processing the cycle C^* containing an unmatched vertex

Let us denote the vertices of C^* by $x_1, \ldots, x_{|C^*|}$, in the order they appear around C^* and so that $v = x_1$, $x = x_2$, $y = x_3$ and $z = x_4$.

Lemma 5.4. *Assume C^* is even-length and consider any phase of the algorithm with C^* unprocessed. Then there is a $(+1)$-support of C^* of weight at least $(\frac{1}{4}, \frac{1}{2}) \star C^* + \frac{1}{2}w(xy)$.*

Proof. We partition $E(C^*)$ into two matchings and then we replace edge yz in one of them by xy, i.e. finally we have $M_1 = \{x_{2t-1}x_{2t} \mid t = 1, \ldots, |C^*|/2\} \setminus \{x_3x_4\} \cup \{x_2x_3\}$ and $M_2 = \{x_{2t}x_{2t+1} \mid t = 1, \ldots, |C^*|/2\}$ (indices modulo $|C^*|$). Note that M_1 and M_2 are $(+1)$-semi-supports (after adding M_1 to P vertex x_4 becomes a loose-end, and after adding M_2 to P vertex x_2 becomes a loose-end). Similarly as in Lemma 3.1, choose one edge from M_1 in each cycle in $P \cup M_1$ and one edge from M_2 in each cycle in $P \cup M_2$, and let Q be the set of these edges.

Let S_1 and S_2 be the $(+1)$-supports obtained from M_1 and M_2 using Lemma 3.1. Let S denote the heavier of them.

Note that edges $x_1x_2 = vx$ and $x_2x_3 = xy$ belong to a path in $P \cup M_1$ (ending in x_4), because $x_3x_4 = yz$ is in M. Also $x_2x_3 = xy$ and x_4x_5 belong to a path in $P \cup M_2$ (ending in $x_2 = x$). It follows that $vx, xy, x_4x_5 \notin Q$.

By inequality (i) in Lemma 3.1, $w(S) \geq \frac{1}{2}(w(S_1) + w(S_2)) \geq w(xy) + \frac{1}{2}w(vx) + \frac{1}{2}w(x_4x_5) + \frac{1}{4}\sum_{i=5}^{|C^*|} w(x_ix_{i+1}) = \frac{1}{4}w(C^* \setminus \{yz\}) + \frac{3}{4}w(xy) + \frac{1}{4}w(vx) + \frac{1}{4}w(x_4x_5)$. Since

$w(xy) \geq w(yz)$, $w(vx) \geq \min(C^*)$ and $w(x_4x_5) \geq \min(C^*)$ we get finally $w(S) \geq (\frac{1}{4}, \frac{1}{2}) \star C^* + \frac{1}{2}w(xy)$. □

Lemma 5.5. *Assume C^* is odd-length. Consider any phase of the algorithm with C^* unprocessed and with at least one loose-end. Then there is a 0-support of C^* of weight at least $(\frac{1}{4}, \frac{1}{2}) \star C^* + \frac{1}{2}w(xy)$.*

Proof. Note that $|C^*| \geq 5$. Let $|C^*| = 2k+1$ and let u be a loose-end. Let $M_1 = \{x_{2t-1}x_{2t} \mid t = 1, \ldots, k\} \setminus \{x_3x_4\} \cup \{x_2x_3, x_{2k+1}u\}$ and $M_2 = \{x_{2t}x_{2t+1} \mid t = 1, \ldots, k\} \cup \{u, x_1\}$. Note that M_1 and M_2 are 0-semi-supports (after adding M_1 to P vertex x_4 becomes a loose-end, after adding M_2 to P vertex x_2 becomes a loose-end, and in both cases u ceases to be a loose-end). Similarly as in Lemma 3.1, choose one edge from M_1 in each cycle in $P \cup M_1$ and one edge from M_2 in each cycle in $P \cup M_2$, and let Q be the set of these edges.

Let S_1 and S_2 be the 0-supports obtained from M_1 and M_2 using Lemma 3.1. Let S denote the heavier of them.

By the same argument as in the proof of Lemma 5.4, $vx, xy, x_4x_5 \notin Q$. Hence by inequality (i) in Lemma 3.1, $w(S) \geq \frac{1}{2}(w(S_1) + w(S_2)) \geq w(xy) + \frac{1}{2}w(vx) + \frac{1}{2}w(x_4x_5) + \frac{1}{4}[w(x_{2k+1}u) + w(ux_1) + \sum_{i=5}^{2k} x_ix_{i+1})] = \frac{1}{4}w(C^* \setminus \{yz, x_{2k+1}x_1\}) + \frac{3}{4}w(xy) + \frac{1}{4}[w(vx) + w(x_4x_5) + w(x_{2k+1}u) + w(ux_1)]$. Since $w(x_{2k+1}u) + w(ux_1) \geq w(x_{2k+1}x_1)$, $w(xy) \geq w(yz)$, $w(vx) \geq \min(C^*)$ and $w(x_4x_5) \geq \min(C^*)$ we get finally $w(S) \geq (\frac{1}{4}, \frac{1}{2}) \star C^* + \frac{1}{2}w(xy)$. □

5.3 Final remarks

Note that if C^* is even-length then it "behaves" like a good odd cycle in the even case algorithm, i.e. it always has a (+1)-support of large enough weight. On the other hand, if C^* is odd-length, it "behaves" like an even cycle in the even case algorithm, i.e. if there is a loose-end, C^* has a 0-support of large enough weight. Hence, if C^* is even, we process it in stage 1 (thus making a loose-end which may be needed by some bad odd cycle) and otherwise we process it in stage 3.

Since the assumpions of Lemma 5.3 are satisfied we get

Theorem 5.6. *Metric MAX-TSP problem can be 7/8-approximated in polynomial time for any input graph.*

It is an interesing question whether one can avoid the overhead of $O(n^4)$ in the time complexity of the odd case.

Acknowledgments

We are indebted to an anonymous Theoretical Computer Science reviewer for reading the paper carefully and for numerous helpful remarks.

References

[1] Z.-Z. Chen and T. Nagoya. Improved approximation algorithms for metric max TSP. In *Proc. ESA'05*, pages 179–190, 2005.

[2] R. Hassin and S. Rubinstein. A 7/8-approximation algorithm for metric Max TSP. *Inf. Process. Lett.*, 81(5):247–251, 2002.

[3] A. V. Kostochka and A. I. Serdyukov. Polynomial algorithms with the estimates 3/4 and 5/6 for the traveling salesman problem of the maximum (in Russian). *Upravlyaemye Sistemy*, 26:55–59, 1985.

[4] Ł. Kowalik and M. Mucha. 35/44-approximation for asymmetric maximum TSP with triangle inequality. In *Proc. 10th Workshop on Algorithms and Data Structures (WADS'07)*, pages 590–601, 2007.

[5] A. I. Serdyukov. The traveling salesman problem of the maximum (in Russian). *Upravlyaemye Sistemy*, 25:80–86, 1984.