Algorithmica (2011) 59: 240–255
DOI 10.1007/s00453-009-9306-3

35/44-approximation for Asymmetric Maximum TSP with Triangle Inequality

Lukasz Kowalik · Marcin Mucha

Received: 25 June 2008 / Accepted: 5 April 2009 / Published online: 16 April 2009

Abstract We describe a new approximation algorithm for the asymmetric maximum traveling salesman problem (ATSP) with triangle inequality. Our algorithm achieves approximation factor 35/44 which improves on the previous 31/40 factor of Bläser, Ram and Sviridenko (Lecture Notes in Computer Science, vol. 3122, pp. 350–359, 2005).

Keywords Traveling salesman problem · Asymmetric · Maximum · Triangle inequality · Approximation · Algorithm

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 $w : V^2 \to \mathbb{R}$ one has to find a Hamiltonian cycle of minimum weight. This problem is probably the most widely known example of an inapproximable NP-hard problem. However, there is a lot of research on approximation of several natural variants of TSP. These variants are still NP-hard, but allow approximation. One of them is the

Part of this work was done while both authors were staying at the Max Planck Institute in Saarbruecken, Germany. 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 Proc. 10th International Workshop on Algorithms and Data Structures (WADS 2007), LNCS 4619, 2007, pp. 589–600.

L. Kowalik (✉) · M. Mucha
Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warszawa, Poland
e-mail: kowalik@mimuw.edu.pl

M. Mucha
e-mail: mucha@mimuw.edu.pl

maximization version (max-TSP for short), where w is assumed to have only nonnegative values (otherwise min-TSP would reduce to it). There are several variants of max-TSP, e.g. the weight function can be symmetric or asymmetric, it can satisfy the triangle inequality or not, etc. (For some results on max-TSP variants see e.g. [2–6].)

In this paper, we are concerned with the variant, where the weight function is asymmetric (in other words, the graph is directed) and satisfies the triangle inequality. This variant is often called *the semimetric max-TSP*.

The first approximation algorithm for this problem was proposed by Kostochka and Serdyukov [7] in 1985 and had approximation ratio of $\frac{3}{4}$. Quite recently, Kaplan, Lewenstein, Shafrir and Sviridenko [8] provided a very general and powerful framework for approximating asymmetric TSP variants and gave improved approximation ratios for 3 different problems: $\frac{4}{3}\log_3 n$ for semimetric min-TSP, $\frac{10}{13}$ for semimetric max-TSP and $\frac{2}{3}$ for asymmetric max-TSP. Chen and Nagoya [9] followed the approach of Kaplan et al. obtaining a slight improvement of approximation ratio for semimetric max-TSP to $\frac{27}{35}$. Using a different approach, Bläser et al. [1] obtained a $\frac{31}{40}$-approximation algorithm for this problem.

We show that in the case of semimetric max-TSP the ideas of Kaplan et al. can be combined with a new patching procedure yielding a $\frac{35}{44}$-approximation.

Overview of the Paper The semimetric max-TSP approximation algorithm of Kaplan et al. combines two ideas: Kostochka and Serdyukov's "patching" algorithm for the same problem and a new framework based on pairs of cycle covers. In Sect. 2 we briefly review both ideas and the way they can be combined. In Sect. 3 we introduce a new patching procedure based on Kaplan et al.'s framework. This immediately leads to a relatively simple $\frac{11}{14}$-approximation for semimetric max-TSP. In Sect. 4 we describe a more elaborate patching method which improves the approximation ratio to $\frac{35}{44}$ by lowerbounding the weight of almost every edge used to form a Hamiltonian cycle.

2 Preliminaries

Throughout the remainder of this paper we assume all graphs to be directed and weighted with a nonnegative weight function w satisfying the triangle inequality.

2.1 Kostochka and Serdyukov's Algorithm

Maximum weight directed cycle cover (possibly containing 2-cycles) can be found in polynomial time by a reduction to maximum weight perfect matching (see e.g. [10]). Many approximation algorithms for TSP problems begin with finding a minimum (maximum) weight cycle cover and then patch it to a Hamiltonian cycle. The following theorem shows how this is done in Kostochka and Serdyukov's algorithm.

Theorem 2.1 *Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a cycle cover in a directed weighted graph G with edge weights satisfying the triangle inequality. Let m_i be the number of edges in C_i and let $w_i = w(C_i)$ be the weight of C_i. Given the cycle cover $\mathcal{C}$, we can find in polynomial time a Hamiltonian cycle of weight $\sum_{i=2}^{k}(1 - \frac{1}{2m_i})w_i$.*

A slightly weaker version of the above theorem is due to Kostochka and Serdyukov [7]. The version in this paper is taken from Kaplan et al. [8].

The maximum weight cycle cover has weight at least as large as the maximum weight Hamiltonian cycle. From Theorem 2.1 it follows that

Theorem 2.2 *There exists a* $\frac{3}{4}$*-approximation algorithm for semimetric max-TSP.*

2.2 The Algorithm of Kaplan et al.

The 2-cycles are the obvious bottleneck of the above approach. If we could find, in polynomial time, a maximum weight cycle cover with no 2-cycles, we would get a 5/6-approximation algorithm. Unfortunately, finding such a cover is an NP-hard problem (see e.g. [11]). Kaplan et al. [8] proposed the following alternative approach.

Theorem 2.3 *Let* $G=(V,E)$ *be a directed weighted graph. We can find in polynomial time a pair of cycle covers* $\mathcal{C}_1$, $\mathcal{C}_2$ *such that* (i) $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *share no* 2*-cycles,* (ii) *total weight* $w(\mathcal{C}_1)+w(\mathcal{C}_2)$ *of the two covers is at least* 2OPT, *where* OPT *is the weight of the maximum weight Hamiltonian cycle in* G.

We will call such pairs of cycle covers *nice pairs of cycle covers*.

Observation 1 (Kaplan et al.) In the above theorem, we can assume that the graph consisting of all the 2-cycles of $\mathcal{C}_1$ and $\mathcal{C}_2$ does not contain oppositely oriented cycles. For if it does contain such cycles, say C and its opposite $\hat{C}$, we can remove all the 2-cycles forming C and $\hat{C}$ from $\mathcal{C}_1$ and $\mathcal{C}_2$ and instead add C to $\mathcal{C}_1$ and $\hat{C}$ to $\mathcal{C}_2$.

Theorem 2.4 *There exists a* $\frac{10}{13}$*-approximation algorithm for semimetric max-TSP.*

The proof of the above theorem can be found in [8]. Since our approach extends that of Kaplan et al., we include it here for completeness. Let us first introduce a few definitions. A *bi-path* is a pair of oppositely oriented paths, i.e. a path and its opposite. As a special case, a *bi-edge* is a single edge together with its opposite edge, i.e. a 2-cycle. A *bi-cycle* is a pair of oppositely oriented cycles. Finally, a *Hamiltonian bi-cycle* is a pair of oppositely oriented Hamiltonian cycles.

Proof of Theorem 2.4 Let $\mathcal{C}_1, \mathcal{C}_2$ be a nice pair of cycle covers. Applying Theorem 2.1 to $\mathcal{C}_1$ and $\mathcal{C}_2$, we get two Hamiltonian cycles H_1, H_2 with total weight $w(H_1)+w(H_2)\geq\frac{3}{4}W_2+\frac{5}{6}W_{3+}$, where W_2 is the total weight of 2-cycles in $\mathcal{C}_1$ and $\mathcal{C}_2$ and W_{3+} is the total weight of all the other cycles.

Another way to construct a Hamiltonian cycle using $\mathcal{C}_1$ and $\mathcal{C}_2$ is to consider the graph H consisting of all the 2-cycles of $\mathcal{C}_1$ and $\mathcal{C}_2$. It follows from Observation 1 that H is a union of disjoint bi-paths. We can patch these bi-paths arbitrarily to get a Hamiltonian bi-cycle $\hat{H}$ of weight $w(\hat{H})\geq W_2$.

Picking the heaviest cycle out of H_1, H_2 and the two cycles of $\hat{H}$ gives a Hamiltonian cycle of weight at least $\frac{1}{2}\max\{\frac{3}{4}W_2+\frac{5}{6}W_{3+}, W_2\}$. Since $W_2+W_{3+}\geq 2\text{OPT}$, easy calculation (or solving a corresponding linear program) shows that the weight of this heaviest cycle is at least $\frac{10}{13}\text{OPT}$. □

3 Spanning Bi-paths and 11/14-approximation

Kaplan et al.'s algorithm (see Theorem 2.4) balances two solutions. The first one is based on Kostochka and Serdyukov's algorithm and the second one on Kaplan et al.'s approach of constructing a nice pair of cycle covers. However, from these cycle covers they pick only the 2-cycles. The basic idea of our approach is to partially incorporate longer cycles into this second solution by constructing additional bi-paths and/or extending existing ones.

Remark 3.1 Cycles of length > 2 do not contain pairs of opposite edges. Hence, not all the new bi-path edges will belong to some cycle.

Let P be a family of disjoint bi-paths. We say that set of bi-edges S is *allowed* w.r.t. P, if S is disjoint from P and the union of the edge sets of P and S is a family of disjoint bi-paths (in particular adding S does not create a bi-cycle in P). A bi-edge e is *allowed* w.r.t. P if $\{e\}$ is allowed w.r.t. P, otherwise e is *forbidden*.

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

Algorithm 1 MAIN ALGORITHM

```
Let $\mathcal{C}_1, \mathcal{C}_2$ be a nice pair of cycle covers
Let $P$ be the family of bi-paths constructed in Kaplan et al.'s Algorithm
Mark all 2-cycles as *processed*
for all unprocessed cycles $C$ in $\mathcal{C}_1$ and $\mathcal{C}_2$ do
    use $C$ to construct a heavy set $S$ of bi-edges, allowed w.r.t. $P$
    $P := P \cup S$
    mark $C$ as processed
end for
arbitrarily patch $P$ to a Hamiltonian bi-cycle
```

Let the *bi-degree* of a vertex v in a family P of bi-paths, denoted as $\deg_P(v)$, be the number of bi-edges in P incident with v (and not the number of edges). In the above algorithm S will always be chosen in such a way that the following is satisfied:

Invariant 1 For any vertex v, $\deg_P(v)$ is not greater than the number of processed cycles containing v.

Remark 3.2 Note that Invariant 1 implies that any vertex has bi-degree at most 2 because every vertex belongs to exactly two cycles.

How do we construct a heavy set of bi-edges S using a cycle C? In this section, S will contain only a single bi-edge e with both ends in C. When choosing $S = \{e\}$, we could pick e to be any of the bi-edges allowed w.r.t. P. However, we want e to have a large weight.

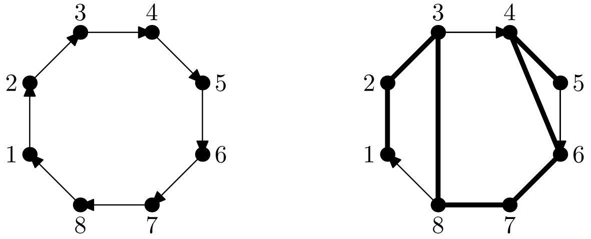

Fig. 1 The proof idea of Lemma 3.3. The weight of the path 1,2,3,8,7,6,4,5 upperbounds the weight of the path 1,2,3,4,5 and the weight of the path 5,4,6,7,8,3,2,1 upperbounds the weight of the path 5,6,7,8,1

Let C be a cycle and let the vertices of C be numbered $1,\dots,k$ along the cycle. A bi-path T is *plane* w.r.t. C if T does not contain two bi-edges u_1u_2, v_1v_2 such that $u_1 < v_1 < u_2 < v_2$ (intuitively, this means that there is a plane embedding of the graph $C \cup T$ such that C bounds the infinite face). We say that T is a *plane spanning bi-path* of C if T is plane w.r.t. C and is incident with all vertices of C. Plane spanning bi-paths are interesting because they have large weight.

Lemma 3.3 *Let T be a plane spanning bi-path of a cycle C. Then $w(T) \geq w(C)$.*

Proof Let the vertices of C be numbered $1,\dots,k$ clockwise and let $T = v_1,\dots,v_k$ be a plane spanning bi-path of C. W.l.o.g. we can assume that $v_1 = 1$. Then for every two vertices $1 \leq u < w \leq v_k$, u has to appear on T before w, because otherwise the part of T connecting v_1 and w would intersect the part connecting u and v_k. Similarly, for every two vertices $v_k \leq u \leq w$, w has to appear on T before u. In other words, vertices $1,\dots,v_k$ appear on T in increasing order, and vertices $v_k,\dots,k$ appear on T in decreasing order (see Fig. 1.)

Bi-path T can be decomposed into two directed paths T_1, T_2, where T_1 begins in v_1 and ends in v_k, and T_2 begins in v_k and ends in v_1. Also, let Q_1 be the directed path $1,2,\dots,v_k$ and let Q_2 be the directed path $v_k, v_k+1,\dots,k,1$, i.e. Q_1 is a part of C from 1 to v_k and Q_2 is the part of C from v_k back to 1.

Vertices $1,\dots,v_k$ appear on T_1 in increasing order, so $w(Q_1) \leq w(T_1)$ by using the triangle inequality. Similarly, vertices $v_k,\dots,k$ appear on T_2 in increasing order, so $w(Q_2) \leq w(T_2)$. We have

$$w(C) = w(Q_1) + w(Q_2) \leq w(T_1) + w(T_2) = w(T). \qquad \square$$

Observation 2 Consider an execution of the Main Algorithm and let C be an unprocessed cycle. If P satisfies Invariant 1, then every vertex on C has bi-degree at most 1. It follows that forbidden bi-edges between vertices of C are the ones that connect two endpoints of some bi-path in P. In particular, these forbidden bi-edges form a matching (i.e. are not incident).

Lemma 3.4 *Consider an execution of the Main Algorithm, let C be an unprocessed cycle, and let P satisfy Invariant 1. Then, there exists T, a plane spanning bi-path w.r.t. C, whose all bi-edges are allowed w.r.t. P.*

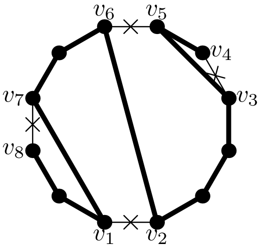

Fig. 2 Finding a plane bi-path avoiding forbidden bi-edges

Proof The bi-path T is constructed as follows. First, for each edge (u,v) of cycle C put bi-edge uv in T whenever it is allowed. Note that at this point T already contains all vertices of C (because forbidden bi-edges with endvertices on C form a matching). Let k be the number of forbidden bi-edges corresponding to edges in $E(C)$. If $k=0$ we remove any bi-edge from T and we are done. Otherwise enumerate the endvertices of the k bi-edges on C from v_1 to v_{2k} along the cycle C. Finally, for every $i=1,\dots,k-1$ add bi-edge $v_i v_{2k-i}$ to T. (See Fig. 2). All these bi-edges are allowed since their endvertices are endvertices of distinct forbidden bi-edges and forbidden bi-edges with ends on C form a matching. Also, T forms a bi-path, since all its vertices are of bi-degree 2 except for v_k and v_{2k}, which are of bi-degree 1. Finally, bi-path T is plane: the only bi-edges that may cross are chords of C, however, for any pair of such distinct chords $v_i v_{2k-i}$, $v_j v_{2k-j}$ either $i<j<2k-j<2k-i$ or $j<i<2k-i<2k-j$. This proves the claim. □

Theorem 3.5 *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be a nice pair of cycle covers of G. Then, there exists a Hamiltonian bi-cycle in G with weight at least $\sum_{i=2}^{\infty}\frac{W_k}{k-1}$, where W_k is the total weight of k-cycles in $\mathcal{C}_1$ and $\mathcal{C}_2$.*

Proof We use the Main Algorithm. The bi-cycle returned by this algorithm contains all bi-edges corresponding to all 2-cycles for a total weight of W_2. When processing a cycle C of length $k\geq 3$, we use Lemma 3.4 to construct T, a plane spanning bi-path w.r.t. C, whose all bi-edges are allowed w.r.t. P. Then we set $S=\{e\}$, where e is the heaviest bi-edge of T. By Lemma 3.3 $w(e)\geq\frac{w(C)}{k-1}$, which proves the claim. □

Theorem 3.6 *There exists a $\frac{11}{14}$-approximation algorithm for semimetric max-TSP.*

Proof As in the proof of Theorem 2.4 we construct a nice pair of cycle covers $\mathcal{C}_1$, $\mathcal{C}_2$ and use Theorem 2.1 to get Hamiltonian cycles H_1, H_2 with total weight

$$w(H_1)+w(H_2)\geq\sum_{i=2}^{\infty}\left(1-\frac{1}{2k}\right)W_k.$$

Next, by Theorem 3.5 to get two more Hamiltonian cycles H_3, H_4 with total weight

$$w(H_3)+w(H_4)\geq\sum_{i=2}^{\infty}\frac{1}{k-1}W_k.$$

Picking the heaviest cycle out of all the H_i gives a Hamiltonian cycle H of weight

$$w(H) \geq \frac{1}{2} \max \left\{ \sum_{i=2}^{\infty} \left(1 - \frac{1}{2k}\right) W_k, \sum_{i=2}^{\infty} \frac{1}{k-1} W_k \right\}.$$

It turns out that we can substitute both inifinite sums with the following finite sums, without changing the resulting approximation ratio:

$$w(H) \geq \max \left\{ \frac{3}{8} W_2 + \frac{5}{12} W_3 + \frac{7}{16} W_4 + \frac{9}{20} W_5 + \frac{11}{24} W_{6+}, \right.$$
$$\left. \frac{1}{2} W_2 + \frac{1}{4} W_3 + \frac{1}{6} W_4 + \frac{1}{8} W_5 \right\}.$$

Now we are going to lowerbound $w(H)$ in terms of OPT, using the fact that $\sum_{i=2}^{\infty} W_k \geq 2\mathrm{OPT}$. Clearly, to this end we can solve the following linear program:

$$\begin{aligned}
&\text{minimize } W \text{ with}\\
&W \geq \frac{3}{8} W_2 + \frac{5}{12} W_3 + \frac{7}{16} W_4 + \frac{9}{20} W_5 + \frac{11}{24} W_{6+},\\
&W \geq \frac{1}{2} W_2 + \frac{1}{4} W_3 + \frac{1}{6} W_4 + \frac{1}{8} W_5,\\
&W_2 + W_3 + W_4 + W_5 + W_{6+} \geq 2\mathrm{OPT},\\
&W_i \geq 0.
\end{aligned}$$

After solving it we get an optimal solution of value $W = \frac{11}{14}\mathrm{OPT}$ for $W_2 = \frac{8}{7}\mathrm{OPT}$, $W_3 = \frac{6}{7}\mathrm{OPT}$, and $W_i = 0$ for $i \geq 4$. □

4 Making Ends Meet and 35/44-approximation

In this section we introduce two improvements. First, we will add more than one bi-edge to the family P of bi-paths, while processing a single cycle C. This is possible if C is long enough. Moreover, recall that in the last step of the algorithm from the previous section we construct a Hamiltonian cycle by patching the bi-paths with arbitrary bi-edges. The endvertices of these bi-edges could belong to distinct cycles and we do not lowerbound their weight in any way. The second improvement we are going to present here is to partially incorporate the patching process into the main algorithm in order to be able to lowerbound this weight. We use this approach for processing short cycles.

4.1 Long Cycles

Lemma 4.1 *Let P be a family of disjoint bi-paths satisfying Invariant* 1 *and let C be an unprocessed cycle of length at least* 5. *Then there exists an allowed family of bi-edges S, such that* (i) *after processing C, the family $P \cup S$ satisfies Invariant* 1,

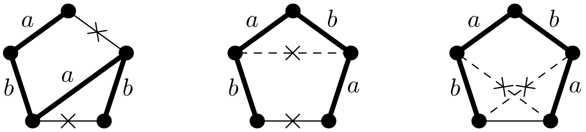

Fig. 3 Coloring a bi-path spanning a 5-cycle. Crossed out edges are forbidden

(ii) $w(S) \geq \frac{1}{4}w(C)$, (iii) *if* $|C| \leq 7$ *then* $w(S) \geq \frac{1}{3}w(C)$, (iv) *if* $|C| = 5$ *then* $w(S) \geq \frac{1}{2}w(C)$.

Proof In order to keep Invariant 1 satisfied, we make S a set of vertex-disjoint allowed bi-edges with endvertices in C. Let Q be the plane bi-path spanning C with no forbidden bi-edges, which exists by Lemma 3.4. We color the bi-edges of Q with two colors: a and b, so that incident bi-edges get distinct colors. Adding all bi-edges of one color, say a, to P may create one or more bi-cycles (note that such a bi-cycle contains at least two bi-edges from Q). For each such bi-cycle we pick one bi-edge from Q and we recolor it to a new color c. Similarly, we recolor some bi-edges from b to d.

It is clear that each of the four color classes is an allowed family of bi-edges. Let S be the heaviest of these four sets. Clearly $w(S) \geq \frac{1}{4}w(Q)$. Since $w(Q) \geq w(C)$ by Lemma 3.3, we get (ii).

Now, let $|C| \leq 7$. Again, we find the bi-path Q and we 2-color it. Suppose that adding all the bi-edges of color a to P gives a bi-cycle. Since there are at most 3 bi-edges colored a and any bi-cycle contains at least 2 such bi-edges, we can only get one such bi-cycle. Similarly, at most one bi-cycle is formed by P and bi-edges colored b. Suppose that both bi-cycles exist (the remaining cases are trivial). We need to recolor one (colored) bi-edge from each cycle to a new color, so that the recolored bi-edges are not adjacent.

Let us start at one end of Q and go along Q until we encounter a colored cycle bi-edge. Assume w.l.o.g. that its color is a. Then, we can recolor both this bi-edge and the furthest cycle bi-edge colored b to a new color c. Clearly, each of the three color classes is an allowed family of bi-edges. Again, we let S be the heaviest of them, obtaining $w(S) \geq \frac{1}{3}w(C)$.

Finally, consider the case of $|C| = 5$. W.l.o.g. we can assume that there are two forbidden bi-edges with endvertices on C (if not, we can just "forbid" additional bi-edges). Figure 3 shows all three possible configurations of these bi-edges together with our choice of the bi-path Q in each case. As before, we 2-color Q, and then set S to be the heavier of the two color classes. This gives $w(S) \geq \frac{1}{2}w(C)$. Observe that in each case both color classes contain a bi-edge with an endvertex not adjacent to a forbidden bi-edge. Such a bi-edge cannot be a part of a bi-cycle in $P \cup S$, so S is allowed. □

4.2 Short Cycles

To get the approximation ratio better than $\frac{11}{14}$ we need to extract more weight from the 3- and 4-cycles when constructing the bi-paths in the Main Algorithm. Unfortunately, it turns out that it is impossible to take more than one bi-edge from *each* such

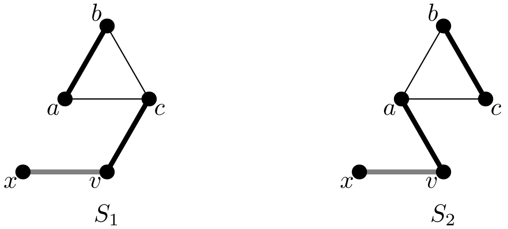

Fig. 4 Case 1 in the proof of Lemma 4.4 (x is the other end of the bi-path beginning in v, $x \notin \{a, b, c\}$)

cycle. Note however, that when only a single bi-edge is put into P when processing a cycle C, at least one vertex v of C becomes a *loose end*, i.e. $\deg_P(v)$ is smaller than the number of processed cycles containing v.

Remark 4.2 If $\deg_P(v) = 0$ and both cycles containing v have already been processed, we consider v to be *two* loose ends.

We can link loose ends from distinct cycles without violating Invariant 1. Surprisingly, it is possible to lowerbound the weight of such links. First let us see how loose ends are created.

Lemma 4.3 *Let P be a family of disjoint bi-paths satisfying Invariant* 1 *and let C be an unprocessed k-cycle. Then there exists an allowed family of bi-edges S such that* (i) $w(S) \geq \frac{1}{k-1} w(C)$, (ii) *after processing C family $P \cup S$ satisfies Invariant* 1, *and* (iii) *the number of loose ends increases by $k - 2$.*

Proof We use the approach described in the proof of Theorem 3.5, i.e. $S = \{e\}$ where e is the heaviest bi-edge of the plane spanning bi-path of C. All the vertices of C except for the two endvertices of e become loose ends. □

The following two lemmas show how loose ends can be used to extract more weight from 3-cycles and 4-cycles.

Lemma 4.4 *Let P be a family of disjoint bi-paths satisfying Invariant* 1 *with at least* 2 *loose ends and let C be an unprocessed* 3*-cycle. Then there exists an allowed family of bi-edges S such that* (i) $w(S) \geq \frac{3}{4} w(C)$, (ii) *after processing C, the family $P \cup S$ satisfies Invariant* 1, *and* (iii) *the number of loose ends decreases by* 1.

Proof Our plan here is to make S contain one bi-edge with both endvertices in C and one bi-edge linking the remaining vertex of C with one of the loose ends. This obviously satisfies (ii) and (iii). We only need to guarantee that S is allowed and that it has weight at least $\frac{3}{4} w(C)$. We consider one of the following two cases, depending on whether or not there exists a loose end v that is not connected to C with a bi-path in P (this bi-path might have length 0 in which case one of the vertices of C is a loose end).

Case 1. There exists such v. Let a, b, c be the vertices of C and suppose $Q = abc$ is a plane spanning bi-path of C with no forbidden bi-edges. Consider two possibilities

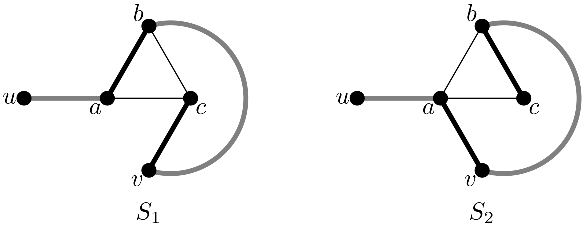

Fig. 5 Case 2 in the proof of Lemma 4.4

for S: $S_1 = \{ab, cv\}$ (ab and cv denote bi-edges here) and $S_2 = \{bc, av\}$. Both are allowed. For example, if we add S_1 to P, cv lies on a bi-path (not a bi-cycle) because v is not connected with C in P, and ab by itself cannot form a bi-cycle because it is allowed as a bi-edge of Q. Similar argument works for S_2. We also have

$$\begin{aligned} w(S_1) + w(S_2) &= w(ab) + w(bc) + w(cv) + w(va) \geq w(ab) + w(bc) + w(ca) \\ &\geq \frac{1}{2}[(w(ab) + w(bc)) + (w(bc) + w(ca)) + (w(ca) + w(ab))] \\ &\geq \frac{3}{2} w(C), \end{aligned}$$

where the second inequality follows from the triangle inequality and the last inequality follows from Lemma 3.3. Taking S to be the heavier of S_1 and S_2 we get the required lower bound of $\frac{3}{4}w(C)$.

Case 2. Such v does not exists, so we have two loose ends u, v connected to two different vertices of C, say u connected to a, and v connected to b. Let c be the remaining vertex of C. Notice that all bi-edges of C are allowed. For if any of them, call it xy, were not allowed, then x and y would be connected with a bi-path in P, and that cannot happen, since we know that either the bi-path starting in x or the bi-path starting in y ends in a loose end.

Consider the two solutions defined in the previous case: $S_1 = \{ab, cv\}$ and $S_2 = \{bc, av\}$. They are both allowed. For example, adding S_1 to P forms a bi-path $\dots cv \dots ba \dots u$ ending in a loose end u, so no bi-cycles are formed. Similar argument works for S_2. The weight argument is the same as in Case 1. □

Lemma 4.5 *Let P be a family of disjoint bi-paths satisfying Invariant 1 with at least 2 loose ends and let C be an unprocessed 4-cycle. Then there exists an allowed family of bi-edges S such that* (i) $w(S) \geq \frac{1}{2}w(C)$, (ii) *after processing C, the family $P \cup S$ satisfies Invariant 1, and* (iii) *the number of loose ends does not change.*

Proof Our plan is to make S contain two bi-edges with both endvertices on C or one bi-edge with both endvertices on C and one bi-edge linking a vertex of C with one of the loose ends. This satisfies (ii) and (iii) and again we only need to guarantee that S is allowed and that it has weight at least $\frac{1}{2}w(C)$. We consider the same two cases as in the previous lemma.

Case 1. There exists a loose end v not connected to C in P.

Let $C = abcd$ and let Q be a plane spanning bi-path of C with no forbidden bi-edges. We consider all solutions of the following form: a bi-edge of Q and a bi-edge connecting one of the remaining vertices of C and v. There a six such solutions since Q has 3 bi-edges and there are always 2 remaining vertices. All these solutions are allowed. That is because the bi-path edge is allowed by itself, and the linking edge cannot form a cycle in P since v is not connected with C in P.

Let us now bound the total weight of these six solutions. Consider a pair of solutions corresponding to a single bi-edge of Q, say xy. The total weight of these two solutions is $2w(xy) + w(vz) + w(vw) \geq 2w(xy) + w(zw)$ (by triangle inequality), where z, w are the two remaining vertices. So we get twice the weight of the bi-path bi-edge and the weight of the complementary bi-edge. Now, notice that for any plane spanning bi-path Q of a 4-cycle, the complementary bi-edges of bi-edges of Q also form a plane spanning bi-path. It follows from Lemma 3.3 that the total weight of all six solutions is at least $3w(C)$. Taking S to be the heaviest of the six solutions gives the required lower bound of $\frac{1}{2}w(C)$.

Case 2. Such v does not exists, so we have two loose ends u, v connected to two different vertices of C. Let $C = abcd$. We have two cases.

Case 2a. v and u are connected to two successive cycle vertices, say u is connected to a and v is connected to b. Consider two solutions: $S_1 = \{da, bc\}$ and $S_2 = \{ab, cv\}$ (here cv is a dummy bi-edge, added only to keep the number of loose ends constant for simplicity). Both solutions are allowed, because if we add any of them to P, each of the added bi-edges lies on a bi-path ending in a loose end. Also $w(S_1) + w(S_2) \geq w(C)$ by Lemma 3.3, because $\{da, bc, ab\}$ is a plane spanning bi-path of C.

Case 2b. v and u are connected to opposite cycle vertices, say u is connected to a and v is connected to c. Consider two solutions: $S_1 = \{ab, cd\}$ and $S_2 = \{ad, bc\}$. The rest of the argument is the same as in the previous Case 2a. □

For technical reasons, that will become clear in the proof of Theorem 4.9, the very last cycle needs to be processed even more effectively. This is possible, because when processing the last cycle we can make P a Hamiltonian bi-cycle. To deal with this special case we use the following lemmas.

Lemma 4.6 *Let P be a family of disjoint bi-paths satisfying Invariant 1 with exactly 1 loose end. Assume that all cycles have been processed except for one 3-cycle C. Then there exists an allowed family of bi-edges S such that* (i) *$P \cup S$ is a Hamiltonian bi-cycle*, (ii) $w(S) \geq \frac{3}{4}w(C)$.

Proof Let $C = abc$. All vertices of G have bi-degree 2 in P except for a, b, c and the loose end v, which all have bi-degree 1 (it might happen that v is one of a, b, c, then it has bi-degree 0). It easily follows that two of the vertices of C, say a and b are connected with a bi-path in P and similarly c and v are connected with a bi-path in P (it might happen that c and v are the same vertex). We consider two solutions: $S_1 = \{ac, bv\}$ and $S_2 = \{bc, av\}$. It's easy to see that both solutions complete P to a Hamiltonian bi-cycle.

An argument similar to the one in the proof of Lemma 4.4 gives a lower bound of $\frac{3}{4}w(C)$ on the weight of the heavier of S_1 and S_2 which ends the proof. □

Lemma 4.7 *Let P be a family of disjoint bi-paths satisfying Invariant 1 with exactly* 2 *loose ends. Assume that all cycles have been processed except for one* 4*-cycle C. Then there exists an allowed family of bi-edges S such that* (i) *$P \cup S$ is a Hamiltonian bi-cycle*, (ii) $w(S) \geq \frac{2}{3}w(C)$.

Proof Let $C = abcd$ and u, v be the loose ends. By analyzing the bi-degrees of all vertices of G in P we arrive in one of the following two cases.

Case 1. u and v are connected to two vertices of C in P. The vertices u and v are connected to may either be successive or opposite vertices on C.

Case 1a. u and v are connected to two successive vertices on C, say u is connected to a and v is connected to b. In this case c and d are also connected with a bi-path in P. We consider four solutions $S_1 = S_2 = \{ad, bc, uv\}$, $S_3 = \{ab, cu, dv\}$ and $S_4 = \{ab, cv, du\}$. It can easily be checked that each of these solutions completes P to a Hamiltonian bi-cycle.

The total weight of the four solutions is

$$\sum_{i=1}^{4} w(S_i) \geq 2w(ab) + 2w(bc) + 2w(ad) + w(cu) + w(du) + w(cv) + w(dv)$$

$$\geq 2(w(ab) + w(bc) + w(cd) + w(da)) \geq \frac{8}{3}w(C),$$

where the last inequality follows from the following consequence of Lemma 3.3

$$\begin{aligned} &3(w(ab) + w(bc) + w(cd) + w(da)) \\ &\quad = (w(ab) + w(bc) + w(cd)) + (w(ab) + w(bc) + w(da)) \\ &\qquad + (w(ab) + w(cd) + w(da)) + (w(bc) + w(cd) + w(da)) \\ &\quad \geq 4w(C). \end{aligned}$$

We make S the heaviest of the four solutions and get $w(S) \geq \frac{2}{3}w(C)$.

Case 1b. u and v are connected to two opposite vertices on C, say u is connected to a and v is connected to c. In this case b and d are also connected with a bi-path in P. We consider two solutions: $S_1 = \{ab, cd, uv\}$ and $S_2 = \{ad, bc, uv\}$ (we only need the uv bi-edges to close the bi-cycle). Again, it's easy to see that both solutions complete P to a Hamiltonian bi-cycle. Their total weight is

$$w(S_1) + w(S_2) \geq w(ab) + w(bc) + w(cd) + w(da) \geq \frac{4}{3}w(C)$$

using the reasoning from case 1a. Making S the heavier of the two solutions we get $w(S) \geq \frac{2}{3}w(C)$.

Case 2. u and v are connected with a bi-path in P. Again we have two cases.

Case 2a. Pairs of successive vertices of C are connected with bi-paths in P, say a is connected with b and c with d.

We consider six solutions: $S_1 = S_2 = \{ad, bu, cv\}$, $S_3 = S_4 = \{ad, bv, cu\}$, $S_5 = \{ac, bu, dv\}$, $S_6 = \{ac, bv, du\}$. Again, it can easily be verified that each of these solutions completes P to a Hamiltonian cycle.

Their total weight is at least

$$\begin{aligned} &4w(ad) + 2(w(bu) + w(cu) + w(bv) + w(cv)) \\ &\quad + 2w(ac) + w(bu) + w(du) + w(bv) + w(dv) \\ &\geq 4w(ad) + 4w(bc) + 2w(ac) + 2w(bd) \\ &\geq 4w(C), \end{aligned}$$

where the last inequality follows from the following corollary of Lemma 3.3

$$\begin{aligned} &2w(ad) + 2w(bc) + w(ac) + w(bd) \\ &= (w(ad) + w(ac) + w(bc)) + (w(ad) + w(bd) + w(bc)) \\ &\geq 2w(C). \end{aligned}$$

Making S the heaviest of the six solutions gives $w(S) \geq \frac{2}{3}w(C)$.

Case 2b. a and c are connected with a bi-path in P, and the same goes for b and d.

We consider four solutions: $S_1 = \{ab, cu, dv\}$, $S_2 = \{ab, cv, du\}$, $S_3 = \{ad, bu, cv\}$, and $S_4 = \{ad, bv, cu\}$. Again, each of these completes P to a Hamiltonian bi-cycle.

The total weight of these four solutions is

$$\begin{aligned} \sum_{i=1}^{4} w(S_i) &= 2w(ab) + w(cu) + w(du) + w(cv) + w(dv) + 2w(ad) \\ &\quad + w(bu) + w(cu) + w(bv) + w(cv) \\ &\geq 2(w(ab) + w(bc) + w(cd) + w(da)) \end{aligned}$$

which we already know is at least $\frac{8}{3}w(C)$. Thus, making S the heaviest of the 4 solutions gives $w(S) \geq \frac{2}{3}w(C)$. □

Lemma 4.8 *Let P be a family of disjoint bi-paths satisfying Invariant 1 with no loose ends. Assume that all cycles have been processed except for one* 4*-cycle C. Then there is an allowed family of bi-edges S such that* (i) *$P \cup S$ is a Hamiltonian bi-cycle*, (ii) $w(S) \geq \frac{1}{2}w(C)$.

Proof Let $C = abcd$. Again, by analyzing the bi-degrees of all vertices of G in P, we conclude that P consists of two bi-paths, pairing the vertices of C. We have two cases.

Case 1. The paired vertices are neighbors on C, say a and b are connected with a bi-path in P, and c and d are. In this case we consider solutions: $S_1 = \{ac, bd\}$ and $S_2 = \{ad, bc\}$. They both complete P to a Hamiltonian bi-cycle. Also, since $S1 \cup S2$ contains a plane spanning bi-path of C, their total weight is at least $w(C)$, and so the heavier of them has weight at least $\frac{1}{2}w(C)$.

Case 2. The paired vertices are not neighbors, i.e. a and c are connected in P and b and d are. The reasoning is the same, only this time we use $S_1 = \{ab, cd\}$ and $S_2 = \{ad, cb\}$. □

4.3 Putting It All Together

Theorem 4.9 *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be a nice pair of cycle covers of G. Then, there exists a Hamiltonian bi-cycle in G with weight at least $W_2 + \frac{5}{8}W_3 + \frac{1}{2}W_4 + \frac{1}{2}W_5 + \frac{1}{3}W_6 + \frac{1}{3}W_7 + \frac{1}{4}W_{8+}$, where W_k is the total weight of k-cycles in $\mathcal{C}_1$ and $\mathcal{C}_2$ and W_{8+} is the total weight of cycles of length at least 8 in $\mathcal{C}_1$ and $\mathcal{C}_2$.*

Proof We use the Main Algorithm and process all the long (i.e. of length at least 5) cycles before the 3- and 4-cycles. Long cycles are processed using Lemma 4.1. As a result we get a family P of bi-paths satisfying Invariant 1 and such that $w(P) \geq W_2 + \frac{1}{2}W_5 + \frac{1}{3}W_6 + \frac{1}{3}W_7 + \frac{1}{4}W_{8+}$. Depending on the number of loose ends in P, we continue in one of the following ways.

Case 1. There are at least 2 loose ends. Then we first process 4-cycles, in any order, using Lemma 4.5 for each cycle. Note that $w(P)$ increases by at least $\frac{1}{2}W_4$ during this phase. Next we process 3-cycles in order of decreasing weight. The first 3-cycle A is processed using Lemma 4.4. As a result the number of loose ends drops by 1 and $W(P)$ increases by $\frac{3}{4}w(A)$. Then we process the second 3-cycle B using Lemma 4.3. We get one loose end and $W(P)$ increases by $\frac{1}{2}w(B)$. We process all the 3-cycles in this way, alternating between Lemmas 4.4 and 4.3. Clearly that overall $W(P)$ increases by at least $\frac{5}{8}W_3$, hence after patching P to a Hamiltonian bi-cycle we get its total weight as claimed.

Case 2. There are no loose ends. Note that, when a cycle C is processed, the number of loose ends increases by $|C| - 2|S|$. Hence, at any time, the parity of the number of loose ends equals the parity of the sum of lengths of the processed cycles. It follows that if there are no loose ends then the sum of lengths of the processed cycles is even. On the other hand, the sum of lengths of all cycles in $\mathcal{C}_1$ and $\mathcal{C}_2$ is $2n$, hence also the sum of lengths of the unprocessed cycles is even. It implies that the number of 3-cycles is even. Now we will consider subcases regarding the number of 3-cycles and 4-cycles.

Case 2a. There are at least two 4-cycles. Then we start by processing the lightest 4-cycle using Lemma 4.3. This gives us 2 loose ends. Next, all 3-cycles and all but one remaining 4-cycles are processed using the algorithm from Case 1. Again, since the number of 3-cycles is even, we still have 2 loose ends when this phase is finished.

It follows that the remaining 4-cycle can be processed using Lemma 4.7. We see that in total $w(P)$ increases by $\frac{1}{3}$ of the weight of the lightest 4-cycle, $\frac{2}{3}$ of the weight of some other 4-cycle, $\frac{1}{2}$ of the weight of all the other 4-cycles and by $\frac{5}{8}W_3$, which is at least $\frac{5}{8}W_3 + \frac{1}{2}W_4$, as required.

Case 2b. There are at least four 3-cycles. Then we start by processing the two lightest 3-cycles using Lemma 4.3. This gives us 2 loose ends and $w(P)$ increases by $\frac{1}{2}$ of the weight of these 3-cycles. Next, all 4-cycles and all but two remaining 3-cycles are processed using the algorithm from Case 1. This increases $w(P)$ by $\frac{5}{8}$ of the weight of the triangles processed in this phase and by $\frac{1}{2}W_4$. Note that since the number of 3-cycles is even, we still have 2 loose ends after this phase. The two remaining 3-cycles are processed using Lemma 4.4 and Lemma 4.6, respectively. Then $w(P)$ increases by $\frac{3}{4}$ of their weight. During the processing of all short cycles $w(P)$ increases by at least $\frac{5}{8}W_3 + \frac{1}{2}W_4$, as required.

Case 2c. There are two 3-cycles and one 4-cycle. Then we consider two methods of processing these cycles and we choose the more profitable one. Method 1: process the 3-cycles using Lemma 4.3 and obtaining 2 loose ends, then process the 4-cycle using Lemma 4.7. In this case $w(P)$ increases by $\frac{1}{2}W_3 + \frac{2}{3}W_4$. Method 2: process the 4-cycle using Lemma 4.3 and obtaining 2 loose ends, then process the 3-cycles using Lemma 4.4 for the first one and Lemma 4.6 for the second one. In this case $w(P)$ increases by $\frac{3}{4}W_3 + \frac{1}{3}W_4$. Clearly the better method gives us $\max\{\frac{1}{2}W_3 + \frac{2}{3}W_4, \frac{3}{4}W_3 + \frac{1}{3}W_4\} \geq \frac{5}{8}W_3 + \frac{1}{2}W_4$, as required.

Case 2d. There are no 3-cycles and there is one 4-cycle. We use Lemma 4.8.

Case 2e. There are two 3-cycles and no 4-cycles. We process the lighter 3-cycle A using Lemma 4.3 which gives us 1 loose end. Then the second 3-cycle B can be processed using Lemma 4.6. This increases $w(P)$ by at least $\frac{1}{2}w(A) + \frac{3}{4}w(B) \geq \frac{5}{8}W_3$ as required.

Case 3. There is exactly one loose end. By the parity argument from Case 2, the number of 3-cycles is odd. We can treat the single loose end as an imaginary 3-cycle I of weight 0. This way the number of 3-cycles becomes even and we again arrive at Case 2. Note that in the algorithms from subcases 2a, 2b and 2e the imaginary triangle would be processed using Lemma 4.3. If we just do nothing while processing I we get the same effect: $w(P)$ grows by $\frac{1}{2}w(I) = 0$ and we get an additional loose end. Case 2d does not apply since we do have 3-cycles. The only case left is a counterpart of Case 2c: there is one 3-cycle and one 4-cycle. Similarly to Case 2c we consider 2 methods and we choose the more profitable one. Method 1 is: process the 3-cycle using Lemma 4.3 obtaining the second loose end and then process the 4-cycle using Lemma 4.7. Method 2 is: process the 4-cycle using Lemma 4.3 obtaining two more loose ends and then process the 3-cycle using Lemma 4.4. Performing the same calculations as in Case 2c, we see that $w(P)$ increases by at least $\frac{5}{8}W_3 + \frac{1}{2}W_4$, as required. □

Theorem 4.10 *There exists a $\frac{35}{44}$-approximation algorithm for semimetric max-TSP.*

Proof Similarly to the algorithm in Theorem 3.6, our algorithm chooses the heaviest of the four Hamiltonian cycles: two constructed by Kostochka and Serdukov's algorithm and the two cycles of the bi-cycle from Theorem 4.9.

Similarly as in the proof of Theorem 3.6 we can lowerbound the weight of the resulting Hamiltonian cycle by solving the following linear program:

$$\begin{aligned}
&\text{minimize } W \text{ with}\\
&W \geq \frac{3}{8}W_2 + \frac{5}{12}W_3 + \frac{7}{16}W_4 + \frac{9}{20}W_5 + \frac{11}{24}W_6 + \frac{13}{28}W_7 + \frac{15}{32}W_{8+},\\
&W \geq \frac{1}{2}W_2 + \frac{5}{16}W_3 + \frac{1}{4}W_4 + \frac{1}{4}W_5 + \frac{1}{6}W_6 + \frac{1}{6}W_7 + \frac{1}{8}W_{8+},\\
&W_2 + W_3 + W_4 + W_5 + W_6 + W_7 + W_{8+} = 2\mathrm{OPT},\\
&W_i \geq 0.
\end{aligned}$$

The minimum value is $W = \frac{35}{44}$, for $W_2 = \frac{10}{11}\mathrm{OPT}$, $W_3 = \frac{12}{11}\mathrm{OPT}$, and $W_i = 0$ for $i \geq 4$. □

Acknowledgements We would like to thank Kasia Paluch for many helpful discussions. We also thank the referees for careful reading and helpful comments.

References

1. Bläser, M., Ram, S., Sviridenko, M.: Improved approximation algorithms for metric maximum ATSP and maximum 3-cycle cover problems. In: Proc. 9th Workshop on Algorithms and Data Structures (WADS 2005). Lecture Notes in Computer Science, vol. 3122, pp. 350–359. Springer, Berlin (2005)
2. Bläser, M.: An 8/13-approximation algorithm for the asymmetric maximum tsp. J. Algorithms **50**(1), 23–48 (2004)
3. Hassin, R., Rubinstein, S.: Better approximations for max TSP. Inf. Process. Lett. **75**(4), 181–186 (2000)
4. Hassin, R., Rubinstein, S.: A 7/8-approximation algorithm for metric Max TSP. Inf. Process. Lett. **81**(5), 247–251 (2002)
5. Rao Kosaraju, S., Park, J.K., Stein, C.: Long tours and short superstrings (preliminary version). In: Proc. 39th Ann. IEEE Symp. on Foundations of Comput. Sci. (FOCS 1994), pp. 166–177 (1994)
6. Lewenstein, M., Sviridenko, M.: A 5/8 approximation algorithm for the maximum asymmetric TSP. SIAM J. Discrete Math. **17**(2), 237–248 (2003)
7. Kostochka, A.V., Serdyukov, A.I.: Polynomial algorithms with the estimates 3/4 and 5/6 for the traveling salesman problem of the maximum. Upr. Sist. **26**, 55–59 (1985) (in Russian)
8. Kaplan, H., Lewenstein, M., Shafrir, N., Sviridenko, M.: Approximation algorithms for asymmetric TSP by decomposing directed regular multigraphs. J. ACM **52**(4), 602–626 (2005)
9. Chen, Z.-Z., Nagoya, T.: Improved approximation algorithms for metric Max TSP. In: Proc. 13th Annual European Symposium on Algorithms (ESA'05). Lecture Notes in Computer Science, vol. 3669, pp. 179–190. Springer, Berlin (2005)
10. Lovász, L., Plummer, M.D.: Matching Theory. Elsevier, Amsterdam (1986)
11. Bläser, M., Manthey, B.: Two approximation algorithms for 3-cycle covers. In: Proc. 5th Int. Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX 2002). Lecture Notes in Computer Science, vol. 2462, pp. 40–50. Springer, Berlin (2002)