A New Approximation Algorithm for the Asymmetric TSP with Triangle Inequality

Markus Bläser*

Abstract

We present a polynomial time factor $0.999 \cdot \log n$ approximation algorithm for the asymmetric traveling salesperson problem with triangle inequality.

1 Introduction

The traveling salesperson problem is one of the most important NP optimization problems. Given a directed or undirected complete loopless graph G with node set V and a weight function w assigning each edge a nonnegative weight, our goal is to find a minimum weight Hamiltonian cycle, i.e., a cycle that visits each node exactly once. Since most variants of the traveling salesperson problem are NP-hard, much effort has been spent on designing approximation algorithms for this problem.

If w is an arbitrary weight function, then the problem is NPO-complete (see e.g. [19] for definitions). Thus, there is no good approximation algorithm, unless $\mathsf{P} = \mathsf{NP}$. A natural restriction is that w should satisfy the (directed) triangle inequality

$$(1) \quad w(u,v) \le w(u,x) + w(x,v) \quad \text{for all pairwise distinct } u,v,x \in V.$$

We call the corresponding minimization problem Δ-ATSP for directed graphs (also called the asymmetric case) and Δ-TSP for undirected graphs. The latter problem is of course a special case of the former, since we require w also to be symmetric. For Δ-TSP, there is a polynomial time factor $\frac{3}{2}$ approximation algorithm due to Christofides [7]. For Δ-ATSP, the approximation performance of the best polynomial time approximation algorithm known today is $\log n$ (where $n = |V|$), as shown by Frieze, Galbiati, and Maffioli [9]. (Here and in the following, all logarithms are to base 2.) We stress that the bound on the approximation performance by Frieze, Galbiati, and Maffioli is exactly $1 \cdot \log n$ and *not* only $O(\log n)$. Since this result by Frieze, Galbiati, and Maffioli back in 1982, no further improvements have been obtained despite an immense amount of research. Even a reduction of the coefficient 1 of $\log n$, which would also be of interest, has not been achieved yet. To say it with the words of Johnson et al. [14], "nothing better has been found in two decades".

The set cover problem is another important and well known NP optimization problems. Johnson [13] provides a polynomial time approximation algorithm with performance ratio $\ln n + 1$. As in the case of Δ-ATSP, this result withstood any attempts of improvement. Finally, Feige [8] gave an explanation for this phenomenon: The set cover problem cannot be approximated within $(1-\epsilon) \cdot \ln n$ for any $\epsilon > 0$, unless the complexity assumption $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log \log n)})$ holds. It is a natural question whether there is a similar explanation for Δ-ATSP or not.

The main result of the present work is a first (admittedly tiny) nontrivial improvement of the approximation performance for Δ-ATSP: We present a factor $0.999 \cdot \log n$ approximation algorithm for Δ-ATSP with polynomial running time. This shows that the performance ratio achieved by Frieze, Galbiati, and Maffioli is not the final one and particularly rules out a result similar to the one by Feige for the set cover problem.

1.1 Notations and Conventions.

For a set of nodes V, let $K(V)$ denote the set of edges $V \times V \setminus \{(v,v) \mid v \in V\}$. Throughout this work, we are considering directed graphs $G = (V, K(V))$ together with a weight function $w : K(V) \to \mathbb{Q}_{\ge 0}$ assigning each directed edge a nonnegative weight. We always require that w fulfills the triangle inequality (1).

A cycle cover of a directed graph G is a spanning subgraph that consists solely of node disjoint directed cycles. A cycle is called a k-cycle if it has length *exactly* k. A cycle cover C is called a k-cycle cover, if each cycle in C has length *at least* k. (By definition, every cycle cover is a 2-cycle cover.) If C is a collection of node disjoint cycles but not a spanning one, we call C also a *partial* cycle cover. If we speak of the number of cycles of a partial cycle cover, then we count each isolated node, i.e., each node that is not part of a cycle, as one single cycle (for consistency reasons).

*Institut für Theoretische Informatik, Universität zu Lübeck, Wallstr. 40, 23560 Lübeck, Germany, blaeser@tcs.mu-luebeck.de

For any subgraph $S = (V, E)$ of G, the weight $w(S)$ of S is defined as the sum of the weights of the edges in E, that is, $w(S) = \sum_{e \in E} w(e)$.

For a given directed graph G with weight function w, let $\mathrm{AB}(G)$ denote the weight of a minimum weight cycle cover. (This is also called the *assignment bound*.) Furthermore, let $\mathrm{TSP}(G)$ denote the weight of a minimum weight TSP tour of G. Obviously, we have $\mathrm{AB}(G) \leq \mathrm{TSP}(G)$. Note that $\mathrm{AB}(G)$ and a corresponding minimum weight cycle cover can be computed in polynomial time.

1.2 Previous Results. The best previous polynomial time approximation algorithm for Δ-ATSP has approximation performance $\log n$, as shown by Frieze, Galbiati, and Maffioli [9]. Utilizing repeated minimum mean cycle computations, Kleinberg and Williamson design an interesting approximation algorithm with polynomial running time which achieves performance ratio $2 \cdot \ln n \approx 1.3863 \cdot \log n$, see [20]. For the symmetric case, Christofides [7] presents a polynomial time factor $\frac{3}{2}$ approximation algorithm for Δ-TSP.

A well studied special case of ATSP is the one, where w may only attain the values one and two. Already this problem is APX-hard [18]. (Strictly speaking, only the MaxSNP-hardness is shown there.) The currently best approximation ratio for this problem is $\frac{4}{3}$ [3]. For the symmetric case, Papadimitriou and Yannakakis [18] achieve approximation performance $\frac{7}{6}$.

Chandran and Ram [6] study the special case of a strengthened triangle inequality. They assume that w fulfils $w(u, v) \leq \gamma(w(u, x) + w(x, v))$ for some γ with $1/2 \leq \gamma < 1$ instead of (1). Their main result is a constant factor $\frac{\gamma}{1-\gamma}$ approximation algorithm for this special case of Δ-ATSP. For the case where w is in addition symmetric, Böckenhauer et al. [4] present polynomial time approximation algorithms with performance ratios $1 + \frac{2\gamma-1}{3\gamma^2-2\gamma+1}$ as well as $\frac{2}{3} + \frac{\gamma}{3(1-\gamma)}$.

Finally, Carr and Vempala [5] extend the so-called Held–Karp conjecture [12] to the asymmetric case. A proof of this extended conjecture would imply that the integrality gap of a certain linear program formulation of Δ-ATSP is bounded by $\frac{4}{3}$. However, this does not automatically yield a corresponding rounding procedure. Moreover, the Held-Karp conjecture still remains unproved after more than three decades.

1.3 New Results. Our main result is a $0.999 \cdot \log n$ approximation algorithm for Δ-ATSP with polynomial running time. Perhaps this does not look very impressive at a first glance, but after two decades of intense research, this is the first nontrivial improvement of the result by Frieze, Galbiati, and Maffioli at all. (The term "nontrivial" excludes ratios of the from $1 \cdot \log n - O(\log \log n)$, which can be achieved by solving small subinstances in the algorithm of Frieze, Galbiati, and Maffioli exactly.) In our point of view, the main achievement of this work is the encouraging proof that the results of Frieze, Galbiati, and Maffioli are not the last word. Particularly, this rules out a result similar to Feige's [8], who shows a tight threshold of $1 \cdot \ln n$ for the approximability of the set cover problem.

The techniques in this paper may be helpful for achieving further improvements concerning the approximability of Δ-ATSP. Also note that it makes sense to care for the constant involved in the O-notation of $O(f(n))$ approximation algorithms. If the constant is too large, then even for slowly growing functions f, the algorithm itself might not be practical for instances of "interesting" size. Any improvement on the constant improves the overall approximation performance.

2 A Generic Repeated Cycle Cover Algorithm

In this section, we present a generic approximation algorithm for Δ-ATSP (see Figure 1). To this aim, we generalize the ideas by Frieze, Galbiati, and Maffioli [9]. Compared to their algorithm, the main difference here is that we repeatedly compute a partial cycle cover C of the given graph, instead of a full cover. After that we choose one representative among the nodes of each cycle. We recursively proceed by computing a TSP tour T' on the graph induced by the representative nodes together with the nodes not contained in any cycle. Then we combine the partial cover C and the tour T' (viewed as a cycle in G) yielding the final tour T.

2.1 Good Partial Cycle Covers. Frieze, Galbiati, and Maffioli repeatedly compute minimum weight 2-cycle covers, which can be done in polynomial time. Better approximation performances can be achieved, if one would compute, say, minimum weight 3-cycle covers. The latter problem is however already APX-hard, even if the edge weights are between one and two (modify the proof in [2] for the maximization variant). Below, we introduce a relaxation of minimum weight k-cycle covers which we call *good partial cycle covers*:

Problem: b-GPCC (with $0 < b \leq 1$)

Instance: a directed graph $G = (V, K(V))$, a weight function w on $K(V)$ that fulfils the triangle inequality.

Goal: a partial cycle cover of weight $\alpha \cdot \mathrm{TSP}(G)$ with $\beta \cdot |V|$ cycles (where $\beta < 1$) such that $\alpha / (-\log \beta) \leq b$.

(Note that $-\log \beta$ is positive, since $0 < \beta < 1$.) It is crucial for our algorithm (see Case 4b in Section 3) that we allow partial cycle covers in the above problem definition. We call a partial cycle cover fulfilling the above condition $\alpha/(-\log\beta) \leq b$ also *b-good*. The importance of the ratio $\alpha/(-\log\beta)$ will become clear in the next subsection. Our aim is to show that b-GPCC is polynomial time solvable for some value of $b < 1$. By computing a minimum weight cycle cover (which has at most $n/2$ cycles), we see that 1-GPCC is solvable in polynomial time.

2.2 Analysis of the Algorithm. The analysis of the generic algorithm in Figure 1 relies on the following two lemmas. For two graphs $H = (V, E)$ and $H' = (V, E')$ over the same node set V, $H \cup H'$ denotes the graph $(V, E \cup E')$.

LEMMA 2.1. *Let D and D' be two (partial) cycle covers over a node set V. Then each weakly connected component of $D \cup D'$ is also strongly connected and Eulerian.*

Proof. Since D and D' consist solely of cycles, each weakly connected component of $D \cup D'$ is also strongly connected. Furthermore each such component is Eulerian, since the indegree of each node equals its outdegree. ■

LEMMA 2.2. *Let H be a directed graph with a weight function w that fulfills the triangle inequality and let S be a spanning subgraph of H. If each weakly connected component of S is Eulerian, then there is a cycle cover C of H such that the number of cycles in C equals the number of connected components of S and $w(C) \leq w(S)$. Furthermore, C can be constructed in polynomial time.*

Proof. For each component of S, we compute an Eulerian tour. This tour is transformed into a cycle by taking *shortcuts* (see e.g. [15]). That is, whenever the Eulerian tour visits a node it has already visited, we go on with the next node in the Eulerian tour that has not been visited so far. Because w fulfills the triangle inequality, none of these shortcuts increases the overall weight. ■

By the above Lemmas 2.1 and 2.2, we can transform $C \cup T'$ into a TSP tour T of G, as $C \cup T'$ has only one connected component by construction. The weight of T is at most $w(C) + w(T')$.

To achieve polynomial running time, we have to choose b in step 1 of the algorithm in such a way that b-GPCC can be solved in polynomial time. Since in each recursive step, the number of nodes decreases by at least one, the overall running time of the generic algorithm is bounded by $|V|$ times the running time of the algorithm used to solve b-GPCC.

Input: a directed graph $G = (V, K(V))$ with a weight function $w : K(V) \to \mathbb{Q}_{\geq 0}$ fulfilling the triangle inequality.
Output: a TSP tour T.

1. Compute a b-good partial cycle cover C of G.
2. From each cycle in C, choose one (arbitrary) node. Let V' be the set consisting of these nodes together with all nodes in V that are not contained in any cycle of C.
3. Recursively compute a TSP tour T' of the graph G' induced by V'.
4. Combine C and T' to obtain the final tour T as described in Lemmas 2.1 and 2.2.

Figure 1: The generic repeated cycle cover algorithm

Let us now analyze the approximation performance of the generic algorithm (in dependence of b). We claim that its approximation performance is $b \cdot \log n$. The proof is by induction: assume that on instances with $n' < |V|$ nodes, the algorithm computes a factor $b \cdot \log n'$ approximation to a minimum weight TSP tour. By the definition of b-good, there are α and β with $\alpha/(-\log\beta) \leq b$ such that C has weight $\alpha \cdot \mathrm{TSP}(G)$ and $\beta \cdot |V|$ cycles. Henceforth, V' has $\beta \cdot |V|$ many nodes. By the induction hypothesis, we compute a tour T with weight at most

$$\begin{aligned} w(C) + w(T') &\leq \alpha \cdot \mathrm{TSP}(G) + b \cdot \log(\beta \cdot |V|) \cdot \mathrm{TSP}(G') \\ &\leq (\alpha + b \cdot \log\beta) \cdot \mathrm{TSP}(G) \\ &\qquad + b \cdot \log(|V|) \cdot \mathrm{TSP}(G) \\ &\leq b \cdot \log(|V|) \cdot \mathrm{TSP}(G). \end{aligned}$$

Note that $\alpha + b \cdot \log\beta \leq 0$ by the definition of b-good. Moreover, $\mathrm{TSP}(G') \leq \mathrm{TSP}(G)$ by the triangle inequality. Thus we obtain the next result.

THEOREM 2.1. *If b-GPCC is solvable in polynomial time for some $0 < b \leq 1$, then there is a polynomial time $b \cdot \log n$ approximation algorithm for Δ-ATSP.*

By plugging in the fact that 1-GPCC is polynomial time solvable, we get the repeated cycle cover algorithm of Frieze, Galbiati, and Maffioli.

3 A Polynomial Time Algorithm for 0.999-GPCC

Throughout the whole section, $G = (V, K(V))$ denotes a directed graph with n nodes and w is a weight function of G fulfilling the triangle inequality.

The problem of computing a minimum weight cycle cover can be solved by the following well-known relaxed linear program:

$$
\text{(2)}\quad
\begin{aligned}
&\text{Minimize} \sum_{(u,v)\in K(V)} w(u,v)x_{(u,v)} \text{ subject to} \\
&\sum_{u\in V\setminus\{v\}} x_{(u,v)} = 1 \qquad \text{for all } v \in V \\
&\qquad\qquad\qquad\qquad (\textit{indegree constraints}), \\
&\sum_{v\in V\setminus\{u\}} x_{(u,v)} = 1 \qquad \text{for all } u \in V \\
&\qquad\qquad\qquad\qquad (\textit{outdegree constraints}), \\
&x_{(u,v)} \geq 0 \qquad\qquad \text{for all } (u,v) \in K(V).
\end{aligned}
$$

The variable $x_{(u,v)}$ corresponds to the edge (u,v) of G. This is merely the LP-formulation of minimum weight bipartite matching, we just have the same node set V on both sides. The matrix corresponding to (2) is totally unimodular [17], thus any optimum basic solution of (2) is integer valued (indeed $\{0,1\}$ valued) and yields a minimum weight cycle cover. The best (strongly) polynomial time algorithm for solving (2) currently known has a running time of $O(n^3)$, see [1].

In the worst case, the minimum weight cycle cover obtained by the above procedure consists solely of 2-cycles. In terms of good cycle covers, we have shown that 1-GPCC is polynomial time solvable. To improve this, we add further constraints to the linear program, the *2-cycle elimination constraints*:

$$
\text{(3)}\quad x_{(u,v)} + x_{(v,u)} \leq 1 \qquad \text{for all } (u,v) \in K(V) \qquad (\textit{2-cycle constraints}).
$$

These constraints are a subset of the so-called *subtour elimination constraints* (see [15]).

If we consider (2) as an integer linear program, adding the constraints (3) ensures that the optimum solution is a 3-cycle cover, since at most one of $x_{(u,v)}$ and $x_{(v,u)}$ may be one. However it is not clear how to solve this integer linear program in polynomial time. In fact, the problem of computing minimum weight 3-cycle covers is APX-hard, even with the triangle inequality [2]. After adding the constraints (3), the corresponding matrix is not totally unimodular any more. Hence a solution of the relaxed linear program may be fractional. The remainder of this section is devoted to how to construct a good partial cycle cover from such a fractional solution.

3.1 Decomposition of a Fractional Solution.

In this subsection, we show how to obtain a collection of cycle covers from an optimum fractional solution of the linear program. The procedure bases on classical results by König as well as Birkhoff and von Neumann and has been used by Lewenstein and Sviridenko [16] in the context of computing *maximum* weight TSP tours.

Let $x^*_{(u,v)}$ denote an optimum solution of the relaxed linear program (2) together with the 2-cycle constraints (3). Let $W^* = \sum_{(u,v)\in K(V)} w(u,v)x^*_{(u,v)}$. Choose B to be the minimum positive integer such that for all (u,v), $B \cdot x^*_{(u,v)}$ is integral. Let $\xi_{(u,v)} = B \cdot x^*_{(u,v)}$.

We create a multigraph H with node set V as follows: for each edge (u,v), we add $\xi_{(u,v)}$ many copies of (u,v) to the edge set of H. By the degree constraints in (2), H is a B-regular multigraph. The next lemma follows at once from König's edge coloring theorem (see e.g. [10, Chap. 30]).

Lemma 3.1. *Let H be a B-regular multigraph. Then the edges of H can be partitioned into B sets such that each set is a cycle cover. Such a partition can be obtained in time polynomial in B and the number of nodes of H.* ■

However, the number B may not be polynomial in the input size. We can circumvent this problem as follows: Recall that a matrix is *doubly stochastic* if all entries are nonnegative and for each row and for each column, the sum over the elements equals one. If in addition, the matrix is also $\{0,1\}$ valued, then we speak of a *permutation matrix.* Note that the $n \times n$–matrix $X^* = (x^*_{(u,v)})$ is doubly stochastic by the constraints in (2). (To build X^*, we order the nodes in V arbitrarily. All entries on the main diagonal of X^* are zero.)

Lemma 3.2. (Birkhoff–Von Neumann) *Every doubly stochastic $n\times n$–matrix S is a convex combination of at most n^2 permutation matrices, i.e., there are permutation matrices $P_1,\dots,P_t$ with $t \leq n^2$ and nonnegative reals α_i with $\sum_{i=1}^t \alpha_i = 1$ such that $S = \sum_{i=1}^t \alpha_i P_i$. Such a decomposition can be found in polynomial time.*

For a proof of the Birkhoff–Von Neumann theorem, see e.g. [11, Chap. 3].

We decompose X^* according to the previous lemma. Every permutation matrix P_i of this decomposition induces a cycle cover of G. (P_i induces indeed a cycle cover and not just a partial cycle cover. A close inspection of the proof of the Birkhoff–Von Neumann theorem shows that all entries on the main diagonal of P_i are zero, since the entries on the main diagonal of X^* are zero.) Now we choose B to be the minimum positive integer such that all $\gamma_i = B \cdot \alpha_i$ are integral.

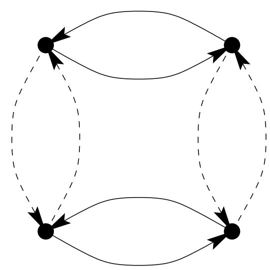

Figure 2: A connected component with four nodes consisting only of 2-cycles from C_i and C_j. Dashed edges are from the one, solid edges from the other cycle cover. Such a component necessarily has an even number of 2-cycles and can be decomposed into two big cycles with opposite directions. One of those cycles is added to C_i, the other to C_j.

(Note that the α_i are all rationals with polynomially many bits in our case.) Each matrix P_i then represents γ_i identical cycle covers. Thus in the following, we work with the cycle covers $C_1, \ldots, C_t$ corresponding to $P_1, \ldots, P_t$, where $t \leq n^2$, instead of working with all B cycle covers explicitly. Each C_i counts as γ_i covers. We have $\sum_{i=1}^{t} \gamma_i w(C_i) = B \cdot W^*$. (Note that each $C_1, \ldots, C_t$ is still a full cycle cover and not a partial one. Later on, we may remove cycles from $C_1, \ldots, C_t$ to obtain partial covers.)

The following lemma, due to Lewenstein and Sviridenko [16], states that every 2-cycle can appear in at most half of the cycle covers $C_1, \ldots, C_t$ (counted with multiplicities $\gamma_1, \ldots, \gamma_t$).

LEMMA 3.3. *Let c be a 2-cycle consisting of two nodes u and v. Let I_1 be the subset of all $i \in \{1, \ldots, t\}$ such that c is a cycle in C_i. Then there is a set of indices I_2 such that for all $j \in I_2$, C_j contains neither the edge (u, v) nor (v, u), and $\sum_{j \in I_2} \gamma_j \geq \sum_{i \in I_1} \gamma_i$.*

3.2 Normalizing the Cycle Covers. In what follows, we will consider the union of two of the cycle covers C_i and C_j. The union $C_i \cup C_j$ consists of strongly connected components that are all Eulerian. Each node is part of one or two cycles. (The case of one cycle arises when C_i and C_j contain an identical cycle.)

In this subsection, we are particularly interested in strongly connected components that are formed solely by 2-cycles from C_i and C_j. If such a component D consists of more than one 2-cycle, then one moment's reflection shows that D consists of an even number of 2-cycles and that its edges can be decomposed into two big cycles, see Figure 2 for an illustration. Each of these two big cycles contains all nodes of D, the only difference is the direction of the cycles. We may now replace the corresponding 2-cycles in C_i and C_j by one of these two big cycles. We again obtain two cycle covers. Lemma 3.3 still holds, since we only removed 2-cycles. Since we only redistributed edges, this process does not change the overall weight $B \cdot W^*$. (Note that the two big cycles necessarily have the same weight, because otherwise we could replace the 2-cycles in C_i and C_j by the lighter of these two big cycles and would have on overall weight of less than $B \cdot W^*$, a contradiction.)

Altogether, we can replace C_i and C_j by two cycle covers C'_i and C'_j such that there is no connected component in C'_i and C'_j with more than two nodes that is formed solely by 2-cycles from C'_i and C'_j. By applying this process $\binom{t}{2}$ times, we may assume in the following that for all i and j, $C_i \cup C_j$ does not contain any connected component with more than two nodes that is build solely from 2-cycles of C_i and C_j. This proves the following lemma.

LEMMA 3.4. *There are cycle covers $C'_1, \ldots, C'_t$ such that the total weight of $C'_1, \ldots, C'_t$ is again $B \cdot W^*$, $C'_1, \ldots, C'_t$ fulfill the claim of Lemma 3.3, and for all $1 \leq i, j \leq t$, no connected component of $C'_i \cup C'_j$ consists solely of 2-cycles of C'_i and C'_j. $C'_1, \ldots, C'_t$ can be computed from $C_1, \ldots, C_t$ in polynomial time.*

For convenience, we use the same names $C_1, \ldots, C_t$ for the new normalized cycle covers in the following.

3.3 Computing Good Partial Cycle Covers. The remainder of this section is devoted to computing a 0.999-good partial cycle cover from the covers $C_1, \ldots, C_t$. To this aim, we introduce a number of parameters. We may of course assume that $B \geq 2$, because otherwise the solution $x^*_{(u,v)}$ is already integral and represents a 3-cycle cover.

- $w_{\min} = \min\{w(C_i) \mid 1 \leq i \leq t\}$ is the minimum among the weights of $C_1, \ldots, C_t$. (Figure 3 shows an example where the $w(C_i)$ are actually distinct.)
- c_i and q_i are chosen such that $c_i \cdot n$ is the total number of cycles in C_i and $q_i \cdot n$ is the number of 2-cycles of C_i. We have $0 \leq q_i \leq c_i \leq 1/2$. Let

  $$\bar{c} = \frac{1}{B} \sum_{i=1}^{t} \gamma_i \cdot c_i$$

  be the average number of cycles (counted with multiplicities). In the same way, let

  $$\bar{q} = \frac{1}{B} \sum_{i=1}^{t} \gamma_i \cdot q_i.$$

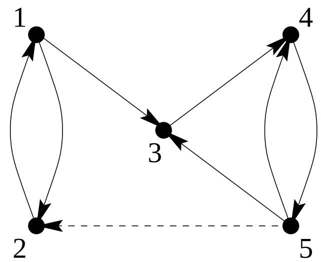

Figure 3: Solid edges have weight 1, dashed ones weight $\frac{3}{2}$, and edges not drawn have weight 2. $(1,3,4,5,2)$ is an optimum 3-cycle cover. It has weight $\frac{11}{2}$. The two cycle covers $(1,3,2),(4,5)$ and $(1,2),(3,4,5)$ are a feasible solution of (2) and (3), both with weight $\frac{1}{2}$. The first cover has weight 6 whereas the second cover has weight 5.

- $s_{i,j}$ is chosen such that $s_{i,j} \cdot n$ is the number of 2-cycles that appear both in C_i and in C_j (where $i \neq j$). Let
$$\bar{s}_i = \frac{1}{B-1} \sum_{j \neq i} \gamma_j s_{i,j}$$
and
$$\bar{s} = \frac{1}{(B-1)B} \sum_{i \neq j} \gamma_i \gamma_j s_{i,j}$$
where the first summation is only over j while the second one is over i and j.

For the analysis, we need the following technical lemma, which bounds the values of $\bar{s}_i$ and $\bar{s}$.

LEMMA 3.5. *For all $1 \leq i \leq t$, we have $\bar{s}_i \leq q_i/2$. Moreover, $\bar{s} \leq \bar{q}/2$ holds.*

Proof. Consider the sum
$$(B-1)\bar{s}_i \cdot n = \sum_{j \neq i} \gamma_j s_{i,j} \cdot n.$$
Since each 2-cycle in C_i can appear in at most $B/2 - 1$ other cycle covers (counted with multiplicities) by Lemma 3.3, we have
$$(B/2-1) q_i \cdot n \geq \sum_{j \neq i} \gamma_j s_{i,j} \cdot n.$$
Consequently,
$$q_i \geq \frac{B-1}{B/2-1} \cdot \bar{s}_i \geq 2\bar{s}_i$$
for $B > 2$. If $B = 2$, then no 2-cycle of C_i can appear in the other cycle cover, thus $\bar{s}_i = 0$. This proves the first inequality. Multiplying with γ_i and summing over all i, we get
$$B\bar{q} = \sum_{i=1}^{t} \gamma_i q_i \geq 2 \sum_{i=1}^{t} \gamma_i \bar{s}_i = 2B\bar{s},$$
which shows the second inequality. ■

We now distinguish a number of cases. In each of them, we compute a 0.999-good partial cycle cover. In the first case, one of the $w(C_i)$ is significantly smaller than W^*, in the second one, one of the C_i has a significant portion of cycles of length greater than two. These cases are obviously easy. In the third case, one $s_{j,k}$ is small and we combine C_j and C_k into one cover. The fourth case is the complement of the first three cases. It splits into two subcases.

Case 1: Assume that $w_{\min} \leq 0.999 \cdot W^*$. In this case, we choose a j with $w(C_j) = w_{\min}$. C_j is a 0.999-good cycle cover, as $W^* \leq \mathrm{TSP}(G)$ and $0.999/(-\log c_j) \leq 0.999$.

For the remaining cases, we may assume that a fraction of at least 0.95 of the $C_1, \ldots, C_t$ have weight at most $1.019 \cdot W^*$ (counted with multiplicities), i.e., there are indices $j_1, \ldots, j_m$ such that $w(C_{j_\mu}) \leq 1.019 \cdot W^*$ for all $1 \leq \mu \leq m$ and $\sum_{\mu=1}^{m} \gamma_{j_\mu} \geq 0.95 \cdot B$. This can be seen as follows: We have $\sum_{i=1}^{t} \gamma_i w(C_i) = B \cdot W^*$ as well as $w(C_i) > 0.999 \cdot W^*$ for all $1 \leq i \leq t$. If a fraction of more than 0.05 of the cycle covers had weight greater than $1.019 \cdot W^*$, then the overall weight of the cycle covers would be $> B \cdot (0.05 \cdot 1.019 + 0.95 \cdot 0.999) \cdot W^* = B \cdot W^*$, a contradiction.

For the remaining cases, let $X = \{j_1, \ldots, j_m\}$ denote this set of indices. Furthermore, let $Y = \sum_{\mu=1}^{m} \gamma_{j_\mu} \geq 0.95 \cdot B$ and define the following variants of $\bar{q}$, $\bar{s}_i$, and $\bar{s}$ with respect to X:

- $\tilde{q} = \frac{1}{Y} \sum_{i \in X} \gamma_i \cdot q_i$,
- $\tilde{s}_i = \frac{1}{Y-1} \sum_{j \in X \setminus \{i\}} \gamma_j s_{i,j}$ and $\tilde{s} = \frac{1}{Y} \sum_{i \in X} \gamma_i \tilde{s}_i$.

Case 2: Assume there is a $j \in X$ with $c_j \leq 0.4931$. Then C_j is 0.999-good, as $w(C_j) \leq 1.019 \cdot W^*$ by the definition of X and $1.019/(-\log 0.4931) \leq 0.999$.

Case 3: Assume there are $j, k \in X$ with $j \neq k$ such that $s_{j,k} \leq 0.2155$. In this case, we combine C_j and C_k to one cycle cover according to the following lemma.

LEMMA 3.6. *For each $j \neq k$, there is a cycle cover of G with weight at most $w(C_j) + w(C_k)$ having no more than $(s_{j,k} + c_j + c_k - q_j - q_k) \cdot n$ cycles. Such a cover can be obtained from C_j and C_k in polynomial time.*

Proof. Consider the graph $C_j \cup C_k$ (we here remove possible double edges). Its weight is at most $w(C_j) + w(C_k)$. Furthermore $C_j \cup C_k$ has $s_{j,k} \cdot n$ many 2-cycles (which appear both in C_j and C_k). The other parts of $C_j \cup C_k$ consist of strongly connected components with at least three nodes. These components are even Eulerian. Since we normalized the covers via Lemma 3.4, each such component contains at least one cycle of length three or greater from C_j or C_k. Since there are $(c_j - q_j) \cdot n$ and $(c_k - q_k) \cdot n$ such cycles in C_j and C_k, respectively, there are at most that many connected components. As in Lemma 2.2, we can replace each connected component by a cycle without incurring any extra weight, as w obeys the triangle inequality. ■

Note that $c_i - q_i \le \frac{1}{3}(1 - 2q_i)$ holds for all i. This is due to the fact that there are $(1 - 2q_i) \cdot n$ nodes in C_i that do not belong to 2-cycles and thus they are contained in cycles of length at least three. From this it follows that $q_i \ge 3c_i - 1$ for all i. Thus $c_j - q_j \le 1 - 2c_j \le 0.0138$, because otherwise, we would be in Case 2. The same holds for $c_k - q_k$. Thus the cycle cover obtained via Lemma 3.6 has at most $(0.2155 + 0.0276) \cdot n = 0.2431 \cdot n$ cycles and has weight at most $2 \cdot 1.019 \cdot W^* = 2.038 \cdot W^*$. Thus it is 0.999-good, because $2.038/(-\log 0.2431) \le 0.999$.

Case 4: For all i, we have

$$(Y-1) \cdot \tilde{s}_i = \sum_{j \in X \setminus \{i\}} \gamma_j s_{i,j} \le \sum_{j \ne i} \gamma_j s_{i,j} = (B-1) \cdot \bar{s}_i.$$

Thus,

$$\tilde{s}_i \le \frac{B-1}{Y-1} \cdot \bar{s}_i \le \frac{B-1}{0.95 \cdot B - 1} \cdot \bar{s}_i.$$

Since $\bar{s}_i \le 1/4$ by Lemma 3.5 for all i (note that $q_i \le 1/2$, since C_i is a 2-cycle cover) and $B \ge 2$, we have

$$\tilde{s}_i \le \frac{1}{0.95 \cdot 2 - 1} \cdot \bar{s}_i \le \frac{1}{0.9} \cdot 0.25 \le 0.2778.$$

Thus for each $j \in X$ there is a $k \in X$ such that $s_{j,k} \le 0.2778$. Fix such a pair j and k. Since we are not in Case 3, we have $s_{j,k} > 0.2155$.

We consider two subcases:

Case 4a: In this case, the 2-cycles that appear both in C_j and in C_k have total weight at least $0.338 \cdot W^*$. We now proceed as in Case 3. But we have to account for the weight of the 2-cycles that appear both in C_j and in C_k only once. (In Case 3, this weight could have been zero, thus we did not mention this there.) Therefore the total weight of the obtained cycle cover is at most $w(C_i) + w(C_j) - 0.338 \cdot W^* \le (2.038 - 0.338) \cdot W^* = 1.7 \cdot W^*$. On the other hand, the number of cycles in the cover is at most $(0.2778 + 2 \cdot 0.0138) \cdot n \le 0.3054 \cdot n$. Therefore the cycle cover obtained is 0.999-good, as $1.7/(-\log 0.3054) \le 0.999$.

Case 4b: Now assume that the 2-cycles that appear both in C_j and in C_k have total weight less than $0.338 \cdot W^*$. Since $s_{i,j} \ge 0.2155$, these 2-cycles are a partial cycle cover with at most $n - 0.2155 \cdot n = 0.7845 \cdot n$ cycles and weight $\le 0.338 \cdot W^*$. (Recall that in the case of a partial cycle cover, we count each isolated node as a single cycle.) Thus this partial cycle cover is 0.999-good, since $0.338/(-\log 0.7845) \le 0.9653 \le 0.999$.

Final Result: Since the above case distinction is exhaustive, we proved the following result.

THEOREM 3.1. *There is a polynomial time algorithm for* 0.999-GPCC.

Together with Theorem 2.1, we obtain the following corollary.

COROLLARY 3.1. *There is a* $0.999 \cdot \log n$ *approximation algorithm for* Δ-ATSP *with polynomial running time.*

REMARK 3.1. *All numerical calculations were checked with two computer algebra systems (Mathematica 4.0, Maple V) using arbitrary precision arithmetic.*

As Case 4b suggests, the above analysis can be somewhat improved, but not very much. Since this it not very illuminating, we refrain from doing so here. More sophisticated analysis techniques also did not yield any significant improvements.

4 Conclusion

In this paper, we presented a polynomial time $0.999 \cdot \log n$ approximation algorithm for Δ-ATSP. This is the first nontrivial improvement of the approximation performance achieved by Frieze, Galbiati, and Maffioli [9] and shows that their algorithm is not the last word. As a main tool, we have introduced the concept of b-good partial cycle covers and designed a polynomial time algorithm for computing 0.999-good partial cycle covers.

A first open question is whether the analysis in Section 3.3 can be significantly improved, perhaps by introducing some more cases. More interesting is probably the challenge of incorporating more of the subtour elimination constraints into the linear program (2). For instance we could add constraints similar to (3) for 3-cycles to the linear program (2). It is not clear to us how to gain some improvements out of these constraints.

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, 1993.

[2] Markus Bläser and Bodo Manthey. Two approximation algorithms for 3-cycle covers. In *Proc. 5th Int. Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX)*, volume 2462 of *Lecture Notes in Comput. Sci.*, pages 40–50. Springer, 2002.

[3] Markus Bläser and Bodo Siebert. Computing cycle covers without short cycles. In *Proc. 9th Ann. European Symp. on Algorithms (ESA)*, volume 2161 of *Lecture Notes in Comput. Sci.*, pages 368–379. Springer, 2001.

[4] J. Böckenhauer, J. Hromkovič, R. Klasing, S. Seibert, and W. Unger. An improved lower bound on the approximability of metric TSP and approximation algorithms for the TSP with sharpened triangle inequality. In *Proc. 17th Int. Symp. on Theoret. Aspects of Comput. Sci. (STACS)*, volume 1770 of *Lecture Notes in Comput. Sci.*, pages 382–394. Springer, 2000.

[5] Robert Carr and Santosh Vempala. Towards a 4/3 approximation for the asymmetric traveling salesman problem. In *Proc. 11th Ann. ACM–SIAM Symp. on Discrete Algorithms (SODA)*, pages 116–125, 2000.

[6] L. Sunil Chandran and L. Shankar Ram. Approximations for ATSP with parametrized triangle inequality. In *Proc. 19th Int. Symp. on Theoret. Aspects of Comput. Sci. (STACS)*, volume 2285 of *Lecture Notes in Comput. Sci.*, pages 227–237, 2002.

[7] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. In J. F. Traub, editor, *Algorithms and Complexity: New Directions and Recent Results*, page 441. Academic Press, 1976.

[8] Uri Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45:634–652, 1998.

[9] A. M. Frieze, G. Galbiati, and F. Maffioli. On the worst-case performance of some algorithms for the asymmetric traveling salesman problem. *Networks*, 12(1):23–39, 1982.

[10] R. L. Graham, M. Grötschel, and L. Lovász, editors. *Handbook of Combinatorics*, volume II. Elvsevier, 1995.

[11] R. L. Graham, M. Grötschel, and L. Lovász, editors. *Handbook of Combinatorics*, volume I. Elvsevier, 1995.

[12] Michael Held and Richard M. Karp. The travelling salesman problem and minimum spanning trees. *Operations Research*, 18:1138–1162, 1970.

[13] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci*, 9:256–278, 1974.

[14] David S. Johnson, Gregory Gutin, Lyle A. McGeoch, Anders Yeo, Weixiong Zhang, and Alexei Zverovitch. Experimental analysis of heuristics for the ATSP. In G. Gutin and A. P. Punnen, editors, *The Traveling Salesman Problem and Its Variations*, volume 12 of *Combinatorial Optimization*. Kluwer, 2002.

[15] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors. *The Traveling Salesman Problem.* Wiley, 1985.

[16] Moshe Lewenstein and Maxim Sviridenko. Approximating asymmetric maximum TSP. In *Proc. 14th Ann. ACM–SIAM Symp. on Discrete Algorithms (SODA)*, 2003.

[17] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, 1982.

[18] C. H. Papadimitriou and M. Yannakakis. The traveling salesman problem with distances one and two. *Math. Operations Research*, 18:1–11, 1993.

[19] Vijay V. Vazirani. *Approximation Algorithms.* Springer, 2001.

[20] David P. Williamson. Lecture notes on approximation algorithms. Technical Report RC 21409, IBM T. J. Watson Research Center, 1999.