# New Approximation Algorithms for (1,2)-TSP


**Anna Adamaszek**
University of Copenhagen, Denmark
anad@di.ku.dk

**Matthias Mnich**[1]
Universität Bonn, Germany
and Maastricht University, The Netherlands
mmnich@uni-bonn.de
https://orcid.org/0000-0002-4721-5354

**Katarzyna Paluch**[2]
Wroclaw University, Poland
abraka@cs.uni.wroc.pl
https://orcid.org/0000-0002-7504-6340



## Abstract

We give faster and simpler approximation algorithms for the $(1,2)$-TSP problem, a well-studied variant of the traveling salesperson problem where all distances between cities are either 1 or 2.

Our main results are two approximation algorithms for $(1,2)$-TSP, one with approximation factor 8/7 and run time $O(n^3)$ and the other having an approximation guarantee of 7/6 and run time $O(n^{2.5})$. The 8/7-approximation matches the best known approximation factor for $(1,2)$-TSP, due to Berman and Karpinski (SODA 2006), but considerably improves the previous best run time of $O(n^9)$. Thus, ours is the first improvement for the $(1,2)$-TSP problem in more than 10 years. The algorithm is based on combining three copies of a minimum-cost cycle cover of the input graph together with a relaxed version of a minimum weight matching, which allows using "half-edges". The resulting multigraph is then edge-colored with four colors so that each color class yields a collection of vertex-disjoint paths. The paths from one color class can then be extended to an 8/7-approximate traveling salesperson tour. Our algorithm, and in particular its analysis, is simpler than the previously best 8/7-approximation.

The 7/6-approximation algorithm is similar and even simpler, and has the advantage of not using Hartvigsen's complicated algorithm for computing a minimum-cost triangle-free cycle cover.



**2012 ACM Subject Classification** Theory of computation → Approximation algorithms analysis

**Keywords and phrases** Approximation algorithms, traveling salesperson problem, cycle cover

**Digital Object Identifier** 10.4230/LIPIcs.ICALP.2018.9


## 1 Introduction

The metric traveling salesperson problem (TSP) is one of the most fundamental combinatorial optimization problems. Given a complete undirected graph $G$ with a metric cost function $c$ on the edges of $G$, the goal is to find a tour $\mathcal{T}$ (i.e., a Hamiltonian cycle) of minimum cost in $G$, where the cost of $\mathcal{T}$ is the sum of costs of the edges traversed by $\mathcal{T}$. Four decades ago, Christofides [8] devised a polynomial-time algorithm that always outputs a tour with cost at most 3/2 times the cost of an optimal tour. Improving this factor remains a major open problem in the area of approximation algorithms.


[1] Supported by ERC Starting Grant 306465 (BeyondWorstCase) and DFG grant MN 59/4-1.
[2] Supported by Polish National Science Center grant UMO-2013/11/B/ST6/01748.




45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 9; pp. 9:1–9:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The metric TSP is well-known to be NP-hard; it is one of Karp's 21 NP-complete problems [16]. In fact, Karp showed that the special case of metric TSP in which all distances between the cities are either 1 or 2, i.e., the cost function is of the form $c : E(G) \to \{1, 2\}$, is NP-hard. This special case is generally known as the $(1, 2)$-TSP problem. Notice that any instance of $(1, 2)$-TSP satisfies the triangle inequality. The $(1, 2)$-TSP problem has been considered in numerous papers [1, 2, 4, 10, 12, 16, 18, 19, 21, 24].

After Karp established the NP-hardness of $(1, 2)$-TSP, Papadimitriou and Yannakakis showed the problem to be APX-hard [24]. The currently best known inapproximability bound for $(1, 2)$-TSP is 535/534 [18]. A certain restriction of $(1, 2)$-TSP was considered by Fernandez de la Vega and Karpinski [10]. It was this restriction of $(1, 2)$-TSP that Trevisan [27] reduced from to establish the inapproximability of TSP in $\mathbb{R}^{\log n}$ under any $\ell_p$ metric. This hardness complemented Arora's breakthrough result [3] that TSP in $\mathbb{R}^2$ admits a PTAS under any $\ell_p$ metric.

One can also view the $(1, 2)$-TSP problem as the problem of finding a traveling salesperson tour that uses the maximum number of 1-edges in the given instance; here and throughout the paper, we will refer to edges of cost $i$ as *$i$-edges*, for $i \in \{1, 2\}$. Alternatively, $(1, 2)$-TSP may be seen as a generalization of the Hamiltonian Cycle problem with non-edges represented by 2-edges.

Both $(1, 2)$-TSP and $(1, 2)$-ATSP (i.e., when the underlying graph $G$ is a complete directed graph) are well-studied from the approximation point of view. For $(1, 2)$-TSP, it is NP-hard to obtain a performance guarantee better than 535/534 [18]. Papadimitriou and Yannakakis [24] gave a 7/6-approximation algorithm for $(1, 2)$-TSP; their algorithm works by successively merging cycles of a triangle-free cycle cover of the graph which they obtained by running Hartvigsen's algorithm [14]. The approximation factor was improved by Bläser and Ram [5] to 65/56, and to 8/7 by Berman and Karpinski [4]. Berman and Karpinski [4] used a local search approach: starting from a path cover they employ local improvements according to certain criteria, and finally connect the paths arbitrarily to a tour. Their algorithm takes time $O(n^9)$ for $n$-city instances.

## 1.1 Our Results

Our main results are novel approximation algorithms for $(1, 2)$-TSP that obtain the approximation ratios of 8/7 and 7/6, respectively.. The 8/7-approximation matches the best approximation factor known for $(1, 2)$-TSP, obtained by Berman and Karpinski [4], while improving the run time from $O(n^9)$ to $O(n^3)$. This is the first improvement for this classical problem in over 10 years.

▶ **Theorem 1.** *The (1,2)-TSP problem admits an $8/7$-approximation in time $O(n^3)$, and a $7/6$-approximation in time $O(n^{2.5})$.*

In this extended abstract we focus on presenting the 7/6-approximation algorithm, which is the simpler of our two algorithms. It is worth noting that it does not rely on Hartvigsen's involved algorithm [14] for computing a minimum-cost triangle-free cycle cover; in contrast, the 7/6-approximation by Papadimitriou and Yannakakis [24] relies on Hartvigsen's algorithm. (Papadimitriou and Yannakakis also gave an 11/9-approximation algorithm that does not use Hartvigsen's algorithm.) We defer the full details of our 8/7-approximation algorithm to the full version of this paper.

**Outline of the approach.** The idea of the 7/6-approximation algorithm is as follows. We start with computing a minimum cost cycle cover $\mathcal{C}_{\min}$ of the input graph $G$. Recall that a *cycle cover* of a graph $G$ is a collection of simple cycles of $G$ such that each vertex belongs to exactly one cycle. Notice that the cost of $\mathcal{C}_{\min}$ is a lower bound on $\mathsf{opt}(G,c)$, where $\mathsf{opt}(G,c)$ denotes the cost of an optimal traveling salesperson tour in the graph $G$ with cost function $c$. The cost of a minimum cost perfect matching $M_{\min}$ of $G$ is also a lower bound, but this time on $\mathsf{opt}(G,c)/2$. This leads to our key idea of constructing a multigraph $\hat{G}$ on $V(G)$ from two copies of $\mathcal{C}_{\min}$ and one copy of $M_{\min}$. It readily follows that the cost of $\hat{G}$ satisfies $c(\hat{G}) \leq \frac{5}{2}\mathsf{opt}(G,c)$.

Next, we would like to color each edge of $\hat{G}$ with one of three colors so that each color class consists of vertex-disjoint paths, i.e., we would like to "path-3-color" $\hat{G}$. Given a path-3-coloring of $\hat{G}$, the paths of the color class that contains the maximum number of 1-edges can be patched in an arbitrary manner to form a traveling salesperson tour of weight not exceeding $\frac{7}{6}\mathsf{opt}(G,c)$. The exact calculation is given in Sect. 3.1.

However, we observe that not every multigraph $\hat{G}$ obtained from $\mathcal{C}_{\min}$ and $M_{\min}$ in the above way is path-3-colorable. For example, a subgraph of $\hat{G}$ obtained from a 4-cycle (called a *square*) $C \in \mathcal{C}_{\min}$ such that two edges of $M_{\min}$ connect vertices of $C$ cannot be path-3-colored. The reason is that $\hat{G}$ has two copies of each edge of $C$ and additionally two more edges coming from $M_{\min}$, and clearly it is not possible to color these ten edges with three colors without creating a monochromatic cycle.

Similarly, a subgraph of $\hat{G}$ obtained from a 3-cycle (called a *triangle*) $C \in \mathcal{C}_{\min}$ such that one of the edges of $M_{\min}$ connects vertices of $C$ cannot be path-3-colored. An edge of $M_{\min}$ connecting two vertices of a cycle $C \in \mathcal{C}_{\min}$ is going to be called an *internal edge* of $C$).

While triangles of the above sort can be handled, by flipping edges, squares with two internal edges of $M_{\min}$ are problematic. Moreover, there are problem instances where every perfect matching of weight at most $\mathsf{opt}/2$ uses two internal edges of some square of $\mathcal{C}_{\min}$.

To get around this obstacle, we relax the notion of a matching and allow it to contain "half-edges". A *half-edge* of an edge $e$ is, informally speaking, half of the edge $e$ that contains exactly one of its endpoints. The notion of half-edges has been introduced by Paluch et al. [23]. We call such a relaxed matching $M^{\frac{1}{2}}$ with half-edges *perfect* if every vertex of the graph has exactly one edge or half-edge of $M^{\frac{1}{2}}$ incident to it. Now, we would like to compute a minimum-cost perfect matching $M^{\frac{1}{2}}_{\min}$ with half-edges, such that the half-edges can appear in a controlled way. In particular, for each 4-cycle of $\mathcal{C}_{\min}$ the matching uses correspondingly at most three "internal" half-edges; here, a half-edge of edge $e$ is *internal* for a cycle $C$ if is derived from an edge of $G$ whose both endpoints belong to $C$. In such a matching the problem described above cannot occur. In Sect. 3.2 we show that $M^{\frac{1}{2}}_{\min}$ can be computed in time $O(n^{2.5})$, and that its weight is at most $\mathsf{opt}(G,c)/2$.

Next, from two copies of $\mathcal{C}_{\min}$ and one copy of $M^{\frac{1}{2}}_{\min}$ we will build a multigraph $\hat{G}$ whose cost is at most $\frac{5}{2}\mathsf{opt}(G,c)$ and that, after some modifications, is path-3-colorable, which yields the desired 7/6-approximation algorithm for (1,2)-TSP.

**Modifying the multigraph $\hat{G}$.** Before the multigraph $\hat{G}$ can be path-3-colored, it needs to be modified in certain ways. First, $\hat{G}$ should not contain any half-edges; so we replace all half-edges by an appropriate number of "whole" edges. Second, while coloring $\hat{G}$ we can restrict ourselves to coloring of edges of cost 1. Third, we remove some 1-edges if some optimal solution contains 2-edges; the exact relationship between the required number of 1-edges in $\hat{G}$ and the number of 2-edges in $\mathsf{opt}(G,c)$ is given in Sect. 3.1.

Fourth, before $\hat{G}$ can be path-3-colored, we need to "flip" certain edges and half-edges. For example, a subgraph of $\hat{G}$ obtained from a triangle $C \in \mathcal{C}_{\min}$ and one internal edge of $C$ contained in $M^{\frac{1}{2}}_{\min}$ cannot be path-3-colored, and we need to flip this edge to the edge of $\hat{G}$ outside of $C$. The algorithm for path-3-coloring essentially comes from Dudycz et al. [9].

## 1.2 Related Work

Despite extensive research, the best approximation algorithm for metric TSP is still Christofides' algorithm [8] from 1976, which has a performance guarantee of 3/2. Generally the bound 3/2 is not believed to be tight. However, the currently largest known lower bound on the performance guarantee obtainable in polynomial time is as low as 123/122 [17]. A promising approach to improving upon the factor of 3/2 for metric TSP is to round a linear programming relaxation known as the Held-Karp relaxation [15], that is widely conjectured to have an integrality gap upper bounded by 4/3. However, even for the graphic TSP, the best known approximation upper bound of 7/5 due to Sebő and Vygen [26] does not match this conjectured upper bound of 4/3.

Another LP relaxation for TSP is the *subtour elimination LP*, which has constraints prescribing any vertex to be incident to exactly two edges of the TSP tour and constraints ruling out incomplete subtours (hence the name) by forcing edges to leave any non-empty proper subset of nodes. The best known integrality gap lower bound of the subtour elimination LP for $(1, 2)$-TSP is 10/9, due to Williamson [29]. Qian et al. [25] showed an integrality gap upper bound of 19/15 for $(1, 2)$-TSP (in a revised version, they improve the integrality gap upper bound to 5/4 and to 26/21 for fractionally Hamiltonian instances), and of 7/6 if the integrality gap is attained by a basic solution of the fractional 2-matching polytope. With the additional assumption that a certain type of modification maintains the 2-vertex connectedness of the support graph, they were able to show a tight integrality gap of 10/9. For fractionally Hamiltonian instances (i.e., where the optimal value of the LP relaxation of the subtour elimination formulation equals the order of the instance), Mnich and Mömke [21] prove integrality upper bounds of 5/4 in the general case and of 10/9 in the case of subcubic support graphs.

For $(1, 2)$-ATSP, it is NP-hard to obtain a performance ratio better than 207/206 [18]. The first non-trivial approximation algorithm for $(1, 2)$-ATSP was given by Vishwanathan [28], with an approximation factor of 17/12. This was improved to 4/3 by Bläser and Manthey [7]. The currently best approximation factor is 5/4, and is due to Bläser [6] and Paluch [22]. For fractionally Hamiltonian instances of $(1, 2)$-ATSP, Mnich and Mömke [21] prove an integrality upper bound of 7/6.

The approach of using half-edges for solving variants of TSP was first used by Paluch et al. [23], who used it to give a 2/3-approximation for Max-ATSP. Later, Paluch [22] used half-edges to improve the approximation guarantee to 3/4 for the special case of Max-ATSP where all edge costs are either zero or one. Recently, Dudycz et al. [9] used half-edges to give a 4/5-approximation for Max-TSP.

# 2 Preliminaries

An instance of the $(1, 2)$-TSP problem consists of pair $(G, c)$, where $G$ is a complete undirected graph and $c : E(G) \to \{1, 2\}$ is an edge cost function, where each edge $e \in E(G)$ has a cost of $c(e) \in \{1, 2\}$. A *tour* for the instance $(G, c)$ is a subset $T \subseteq E(G)$ of edges of $G$ that forms a Hamiltonian cycle of $G$, that is, the edges of $T$ form a cycle that visits each vertex of $G$ exactly once; the *cost* of $T$ is defined as $c(T) = \sum_{e\in T} c(e)$. The goal is to find an *optimal* tour $\mathsf{opt}$ for $(G, c)$, which is a tour of minimum cost. For a real number $r$, a tour $T$ for an instance $(G, c)$ is *$r$-approximate* if $c(T)/c(\mathsf{opt}) \le r$.

For a graph $G$, a *cycle* is a sequence $C = (v_0, \ldots, v_{\ell-1})$ for some $\ell \ge 3$ of pairwise distinct vertices $v_i \in V$ such that $\{v_i, v_{i+1 \pmod \ell}\} \in E$ for $i \in \{0, \ldots, \ell - 1\}$. We refer to $\ell$ as the *length* of $C$, and denote it by $\ell(C)$. For an integer $\ell$, an *$\ell$-cycle* is a cycle of length exactly $\ell$, and an $(\leq \ell)$-*cycle* is a cycle of length at most $\ell$. For the sake of convenience, we also refer to 3-, 4-, 5- and 6-cycles as triangles, squares, pentagons and hexagons, respectively.

Let $C$ be an $\ell$-cycle of $G$. We say that $C$ is *short* if $\ell \leq 6$. Further, an $\ell'$-cycle $C'$ of $G$ with $\ell' < \ell$ is a *subcycle* of $C$ if $V(C') \subset V(C)$. Note that $C$ and $C'$ can visit the vertices of $V(C')$ in different order. An edge $e = \{u, u'\}$ is a *native edge* of $C$ if $u, u'$ are two consecutive vertices of $C$, and a *diagonal* of $C$ if $u, u'$ are two non-consecutive vertices of $C$. When $e$ is a native edge or a diagonal of $C$, we say that $e$ is an *internal edge* of $C$. Finally, we call a cycle $C$ a *1-cycle* if $c(e) = 1$ for all $e \in E(C)$; notice that there is no confusion of this notion with $\ell$-cycles as we consider simple undirected graphs without loops.

**Cycle covers.** Our algorithm utilizes the concept of cycle covers. A *cycle cover* of $G$ is a collection of cycles of $G$ such that each vertex of $G$ belongs to exactly one cycle of the collection. Thus, a Hamiltonian cycle of $G$ is a cycle cover of $G$ that consists of a single cycle. Cycle covers of undirected graphs are also known as 2-factors, because every vertex is incident to exactly two edges.

A cycle cover of $G$ is *triangle-free* if each of its cycles has a length of at least 4. An essential ingredient of our 8/7-approximation algorithm is the following result by Hartvigsen [14]; the algorithm can be implemented to run in time $O(n^3)$ for an $n$-vertex graph [13].

▶ **Proposition 2** ([14])**.** *There is an algorithm that, given a complete graph $G$ with edge costs $c : E(G) \to \{1, 2\}$, in strongly polynomial time computes a triangle-free cycle cover of $G$ with minimum cost under $c$.*

***b*-matchings.** We will use the classical notion of *b-matchings* in graphs, which are a generalization of matchings. Let $H$ be a graph. For a vector $b = (b_v)_{v \in V(H)} \in \mathbb{N}^{|V(H)|}$ where each coordinate corresponds to a vertex of $H$, a *b-matching* in $H$ is a collection of edges $E(b) \subseteq E(H)$ that contains at most $b_v$ edges incident to any vertex $v \in V(H)$. Notice that a $b$-matching with $b_v = 1$ for all $v \in V(H)$ is a classical matching in $H$.

A $b$-matching in $H$ is said to be *maximum* if among all $b$-matchings in $H$ it contains a maximum number of edges. Maximum matchings as well as maximum cost $b$-matchings can be computed in polynomial time. We refer to Lovász and Plummer [20] for further background on $b$-matchings.

We are interested in computing a $b$-matching in a graph $H$ where each vertex $v \in V(H)$ has a lower bound $\ell_v$ and an upper bound $b_v$ - we say that a vertex $v$ has capacity interval $[\ell_v, b_v]$; the $b$-matching $E(\ell, b) \subseteq E(G)$ then contains at least $\ell_v$ edges and at most $b_v$ edges incident to any vertex $v \in V(H)$. Such $b$-matchings can also be computed efficiently:

▶ **Proposition 3** ([11])**.** *There is an algorithm that, given a graph $H$ and capacity intervals $[\ell_v, b_v]$, in time $O(\sqrt{\sum_{v \in V(H)} b_v}|E(H)|)$, computes a largest subgraph $H'$ of $H$ for which $\ell_v \leq d_{H'}(v) \leq b_v$ for every $v \in V(H')$.*

It is possible to reduce the problem of computing a $b$-matching with capacity intervals to the computation of a matching in which each vertex has capacity interval $[0, 1]$ or $[1, 1]$, i.e., a matching in which every vertex with capacity interval $[1, 1]$ is required to be matched; we defer the details to the full version of the paper.

**Half-edges.** Intuitively, half-edges correspond to halves of the edges of a graph and incident to only one vertex of the graph. Formally, from an instance $(G, c)$ of $(1, 2)$-TSP we construct an *extended instance* $(G', c')$ from $(G, c)$, as follows. We start by setting $V(G') = V(G)$ and

$E(G') = E^1(G)$, where $E^1(G)$ denotes the subset of $E(G)$ containing all 1-edges. Next, for each edge $e = \{u, u'\} \in E^1(G)$ we add to $V(G')$ a new vertex $v_e$, and to $E(G')$ the edges $\{u, v_e\}, \{v_e, u'\}$. We refer to the vertices $v_e$ as *extended vertices*, and to the remaining vertices of $G'$ as *basic vertices*. We denote the new edges of $E'$ as *half-edges* and the other edges as *basic edges*. Put concisely, $G'$ is the *extended graph* of $G$ with $V(G') = V(G) \cup \{v_e \mid e \in E^1(G)\}$, $E(G') = E^1(G) \cup \{\{u, v_e\}, \{v_e, u'\} \mid e = \{u, u'\} \in E^1(G)\}$.

A *matching with half-edges* $M^{\frac{1}{2}}$ in $G'$ is a collection of edges in $G'$, in which each vertex has degree 0 or 1 Intuitively, a matching with half-edges in $G'$ corresponds to a relaxation of a matching in $G$, where we can take halves of the edges, incident to only one vertex, to the matching. We define the cost of a matching with half-edges $M^{\frac{1}{2}}$ in $G'$ as $c'(M^{\frac{1}{2}}) = \frac{1}{2}|\{v \in V(G) : v \text{ is matched in } M^{\frac{1}{2}}\}| + |\{v \in V(G) : v \text{ is unmatched in } M^{\frac{1}{2}}\}|$. In other words, a basic vertex that is unmatched in $M^{\frac{1}{2}}$ contributes twice as much cost to $c'(M^{\frac{1}{2}})$ as a matched basic vertex. (We might say that we treat an unmatched basic vertex as if it was matched to a half-edge of a basic 2-edge in a perfect matching with half-edges in a graph $G''$ in which we also add basic 2-edges and their half-edges.) Therefore, any maximum matching with half-edges in $G'$ has minimum cost.

# 3 A Fast and Simple 7/6-Approximation Algorithm for (1,2)-TSP

## 3.1 Outline of the Algorithm

We give an outline of our 7/6-approximation algorithm for (1, 2)-TSP, which is listed as Algorithm 1. For an instance $(G, c)$ of the problem and a fixed tour $\mathcal{T}$ of $G$, let $\alpha_{\mathcal{T}}$ and $\beta_{\mathcal{T}}$ denote the number of 1-edges and 2-edges, respectively, in $\mathcal{T}$.

▶ **Observation 4.** *It holds that* $c(\mathcal{T}) = \alpha_{\mathcal{T}} + 2\beta_{\mathcal{T}} = \alpha_{\mathcal{T}} + 2(|V(G)| - \alpha_{\mathcal{T}}) = 2|V(G)| - \alpha_{\mathcal{T}}$.

Let $G_1$ denote the subgraph of $G$ containing all 1-edges. In step 1 of the algorithm we compute a path-cycle cover $\mathcal{C}_{\min}$ of minimum cost in $(G, c)$, using the algorithm from Proposition 3. A *path-cycle cover* of $G$ is any $b$-matching of $G_1$ such that each vertex $v$ has capacity interval $[0, 2]$. The cost of a path-cycle cover $\mathcal{C}$ of $G$ is defined as $2n - |\mathcal{C}|$. Let $\mathcal{C}_{\min}$ denote a minimum cost path-cycle cover of $G$. Then, clearly, its cost is a lower bound on $c(\mathsf{opt})$.

In step 2 we use $\mathcal{C}_{\min}$ to construct a minimum cost matching with half-edges (and some additional properties) $M^{\frac{1}{2}}$; this construction is described in Sect. 3.2. In Sect. 3.3 we describe step 3, i.e., the construction of the graph $G^1$ from $\mathcal{C}_{\min}$ and $M^{\frac{1}{2}}$. In step 4 we path-3-color $G^1$, for which we use a modification of a path-3-coloring proposed by Dudycz et al. [9] for Max-TSP.

In summary, the algorithm works as follows:

---

**Algorithm 1** Computing a 7/6-approximate solution for an instance $(G, w)$ of (1,2)-TSP.

---

**Input:** An instance $(G, c)$ of (1,2)-TSP.

**Output:** A tour $\mathcal{T}$ of $G$ with cost $c(\mathcal{T}) \le \frac{7}{6}c(\mathsf{opt})$.

1: Find a minimum-cost path-cycle cover $\mathcal{C}_{\min}$ of $(G, c)$.
2: Find a minimum cost matching with half-edges (and some additional properties) $M^{\frac{1}{2}}$.
3: Based on $\mathcal{C}_{\min}$ and $M^{\frac{1}{2}}$, construct a multigraph $G^1$ on vertex set $V(G)$ with at least $\frac{5}{2}\alpha_{\mathsf{opt}} - \beta_{\mathsf{opt}}$ edges of cost 1 from $G$.
4: Path-3-color the edges of $G^1$.
5: Extend the set of edges of $G^1$ from the largest color class arbitrarily to a tour $\mathcal{T}$ of $G$.
6: **return** $\mathcal{T}$

---

▶ **Lemma 5.** *Algorithm 1 gives a* $7/6$*-approximate solution for (1,2)-TSP.*

**Proof.** Let $\mathsf{alg}$ be a solution output by Algorithm 1 on input $(G, c)$, and let $\alpha_{\mathsf{alg}}$ and $\beta_{\mathsf{alg}}$ be the number of 1-edges resp. 2-edges in $\mathsf{alg}$. Then

$$\alpha_{\mathsf{alg}} \geq \frac{\frac{5}{2}\alpha_{\mathsf{opt}} - \beta_{\mathsf{opt}}}{3} = \frac{5}{6}\alpha_{\mathsf{opt}} - \frac{1}{3}\beta_{\mathsf{opt}} \ .$$

By multiplying by 6 and using that $n = \alpha_{\mathsf{opt}} + \beta_{\mathsf{opt}}$, we obtain

$$6\alpha_{\mathsf{alg}} \geq 7\alpha_{\mathsf{opt}} - 2n = 7\alpha_{\mathsf{opt}} - 2(\alpha_{\mathsf{opt}} + \beta_{\mathsf{opt}}) = 5\alpha_{\mathsf{opt}} - 2\beta_{\mathsf{opt}} \ .$$

Subtracting $14n$ from both sides, and substituting $2(\alpha_{\mathsf{opt}} + \beta_{\mathsf{opt}})$ for $2n$ on the right hand side yields $12n - 6\alpha_{\mathsf{alg}} \leq 14n - 7\alpha_{\mathsf{opt}}$, which is equivalent to the desired result of

$$\frac{c(\mathsf{alg})}{c(\mathsf{opt})} = \frac{2n - \alpha_{\mathsf{alg}}}{2n - \alpha_{\mathsf{opt}}} \leq \frac{7}{6} \ .$$ ◀

## 3.2 Computing a Minimum Cost Matching with Half-Edges

In this section we describe the construction of a minimum-cost matching with half-edges (and some additional properties) $M^{\frac{1}{2}}$ in the extended instance $(G', c')$, as defined in Sect. 2. Recall that a square is a 4-cycle. We refer to a diagonal of cost $i$ as an *$i$-diagonal*, for $i = 1, 2$.

Intuitively, we want to ensure that $M^{\frac{1}{2}}$ matches at least one vertex of each 1-square of $\mathcal{C}_{\min}$ with some vertex from outside the square or leaves at least one vertex of a 1-square unmatched. As we want to find such a matching $M^{\frac{1}{2}}$ efficiently, and we want the cost of $M^{\frac{1}{2}}$ to be at most $c(\mathsf{opt})/2$, we allow $M^{\frac{1}{2}}$ to contain half-edges. However, we will only allow half-edges in a controlled manner. The idea is to allow a half-edge $\{u, v_e\}$ within $M^{\frac{1}{2}}$ only when the corresponding edge $e$ is a native edge of a 1-square of $\mathcal{C}_{\min}$. Also, we want to allow at most one half-edge per each 1-square of $\mathcal{C}_{\min}$. For technical reasons (due to parity issues), we have to relax these simple conditions to a more complex set of conditions.

▶ **Definition 6.** A matching with half-edges $M^{\frac{1}{2}}$ in $(G', c')$ is *good* for $\mathcal{C}_{\min}$ if it satisfies the following properties.

**($M^{\frac{1}{2}}$.1)** For each half-edge $\{u, v_e\} \in M^{\frac{1}{2}}$, except at most one *special half-edge*, the edge $e$ is a native edge of a 1-square of $\mathcal{C}_{\min}$. Also, each 1-square of $\mathcal{C}_{\min}$ is incident to at most one half-edge of $M^{\frac{1}{2}}$.

**($M^{\frac{1}{2}}$.2)** For every 1-square $C \in \mathcal{C}_{\min}$ (i) there is a 1-edge $e_C \in M^{\frac{1}{2}}$ incident to $C$ such that the other endpoint of $e_C$ is incident to a cycle of $\mathcal{C}_{\min}$ different from $C$, or (ii) at least one of the vertices of $C$ is unmatched in $M^{\frac{1}{2}}$.

**($M^{\frac{1}{2}}$.3)** $M^{\frac{1}{2}}$ may contain a special half-edge $\{u, v_e\}$ only if $|V(G)|$ is odd and $\mathcal{C}_{\min}$ does not contain a 2-edge. The following conditions are satisfied for the special half-edge $\{u, v_e\}$:

- **a.** $e$ is an edge of a fixed odd-length 1-cycle $C_0 \in \mathcal{C}_{\min}$ called a *special cycle*.
- **b.** If $\mathcal{C}_{\min}$ contains a cycle of length at least 7, then $C_0$ has length at least 7.
- **c.** If $C_0$ is a triangle or pentagon and $\mathcal{C}_{\min}$ consists of at least two cycles, then at least two vertices of $C_0$ are incident to external edges of $M^{\frac{1}{2}}$, or some vertex of $C_0$ is unmatched in $M^{\frac{1}{2}}$.

▶ **Lemma 7.** *Consider an instance $(G, c)$ of $(1, 2)$-TSP with minimum-cost path-cycle cover $\mathcal{C}_{\min}$ and extended instance $(G', c')$. A matching $M^{\frac{1}{2}}$ of minimum cost among all matchings with half-edges $M^{\frac{1}{2}}$ which are good for $\mathcal{C}_{\min}$ can be computed in time $O(n^{2.5})$ where $n = |V(G)|$.*

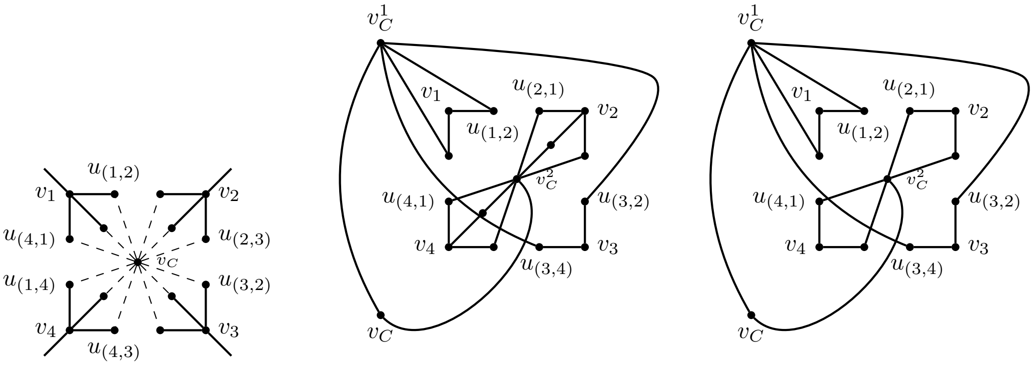


**Figure 1** The gadget for modifying a 1-square $\{v_1, v_2, v_3, v_4\}$ of $\mathcal{C}_{\min}$ which has (a) two 1-diagonals, (b) one 1-diagonal $\{v_2, v_4\}$, and (c) no 1-diagonals.

**Proof.** First, from the instance $(G, c)$ we create an unweighted graph $G_0''$ with a vertex capacity interval for each vertex of $G_0''$. We do that by locally and independently modifying each 1-square in $\mathcal{C}_{\min}$. The modification introduces new vertices and edges, as well as vertex capacity intervals $[\ell_v, u_v]$ for each vertex $v \in V(G_0'')$.

We start by setting $V(G_0'') = V(G)$, and assigning the capacity interval $[0, 1]$ to each vertex. For each edge $e \in E(G)$ which is not an internal 1-edge of a 1-square of $\mathcal{C}_{\min}$, we add $e$ to $E(G_0'')$. Then, for each 1-square $C = (v_1, \dots, v_4) \in \mathcal{C}_{\min}$ we proceed as follows. For each internal 1-edge $e_{\{i,j\}} = \{v_i, v_j\} \in E(G)$ of $C$ (note that we consider both the native 1-edges and the 1-diagonals of $C$ here), we introduce two new vertices $u_{(i,j)}, u_{(j,i)}$ with capacity intervals $[1, 1]$, and two new edges $\{v_i, u_{(i,j)}\}, \{u_{(j,i)}, v_j\}$. We call these added vertices *subdivision vertices.*

The exact type of further modification depends on whether the number of 1-diagonals of $C$ in $(G, c)$ is two, one, or zero.

- **If $C$ has two 1-diagonals:** (See Fig. 1a) Introduce a vertex $v_C$ of capacity interval $[9, 12]$; then connect $v_C$ to all 12 subdivision vertices $u_{(i,j)}, u_{(j,i)}$.
- **If $C$ has exactly one 1-diagonal $\{v_2, v_4\}$:** (See Fig. 1b) Introduce two vertices $v_C^1, v_C^2$ of capacity intervals $[3, 4]$ and $[5, 5]$, respectively, and one vertex $v_C$ of capacity interval $[0, 1]$. Connect $v_{C_1}$ to each vertex $u_{(i,j)}, u_{(j,i)}$ that is a neighbour of $v_1$ or $v_3$, and connect $v_{C_2}$ to each vertex $u_{(i,j)}, u_{(j,i)}$ that is a neighbour of $v_2$ or $v_4$. Further, add two edges $\{v_C, v_C^1\}, \{v_C, v_C^2\}$.
- **If $C$ is a square with no 1-diagonal:** (See Fig. 1c) Introduce two vertices $v_C^1$ and $v_C^2$ of capacity interval $[3, 3]$ and one vertex $v_C$ of capacity interval $[0, 1]$. Connect $v_{C_1}$ to each vertex $u_{(i,j)}, u_{(j,i)}$ that is a neighbour of $v_1$ or $v_3$, and connect $v_{C_2}$ to each vertex $u_{(i,j)}, u_{(j,i)}$ that is a neighbour of $v_2$ or $v_4$. Further, add twoedges $\{v_C, v_C^1\}, \{v_C, v_C^2\}$.

This completes the construction of the graph $G_0''$ with vertex capacity intervals $[\ell_v, u_v]$ for each $v \in V(G_0'')$.

The cost of a $b$-matching $M_0''$ in $G_0''$ is defined as $c'(M_0'') = \frac{1}{2}|\{v \in V(G) : v \text{ is matched in } M_0''\}| + |\{v \in V(G) : v \text{ is unmatched in } M_0''\}|$. For the graph $G_0''$ and vertex capacity intervals $[\ell_v, u_v]$ for each $v \in V(G_0'')$, we compute a minimum-cost $b$-matching $M_0''$ that respects the vertex capacity intervals or, equivalently a $b$-matching that respects the vertex capacity intervals and minimizes the number of basic vertices unmatched in $M_0''$.

▶ **Claim 1.** *For the graph $G_0''$ and vertex capacity intervals $[\ell_v, u_v]$ for each $v \in V(G_0'')$, a minimum-cost b-matching $M_0''$ that respects the vertex capacity intervals can be computed in $O(n^{2.5})$ time.*

We defer the proof of Claim 1 to the full version of this paper.

▶ **Claim 2.** *The b-matching $M_0''$ in $G_0''$ can be transformed into a matching with half-edges $M^{\frac{1}{2}}$ which is good for $\mathcal{C}_{\min}$, has the same cost as $M_0''$, and contains no special half-edge.*

**Proof of Claim 2.** We construct the matching $M^{\frac{1}{2}}$ as follows. For any edge $e = \{u, v\} \in E(G_0'')$ such that both $u, v$ are basic vertices of $G_0''$ (i.e., they correspond to the vertices of $G$, and not to the vertices introduced during the gadget construction), we add the edge $e$ to $M^{\frac{1}{2}}$. Then, consider each 1-square $C = \{v_1, .., v_4\} \in \mathcal{C}_{\min}$ and a gadget corresponding to it. If there are some two vertices $v_i, v_j \in C$ such that both $v_i$ and $v_j$ are matched by the $b$-matching with subdivision vertices, and the edge $e = \{v_i, v_j\}$ in $G'$ is a 1-edge, we add $e$ to $M^{\frac{1}{2}}$. We construct such pairings greedily. For all vertices $v_i$ which are matched by the $b$-matching with subdivision vertices, and which have not been paired, we add a half-edge $\{v_i, v_e\}$ to $M^{\frac{1}{2}}$, where $e = \{v_i, v_{(i \mod 4)+1}\}$.

The degree of each basic vertex $v \in V(G_0'')$ in $M^{\frac{1}{2}}$ is the same as the degree of $v$ in the $b$-matching. The degree of each extended vertex $v \in V(G_0'')$ in $M^{\frac{1}{2}}$ is either 0 or 1. Therefore, $M^{\frac{1}{2}}$ is a matching with half-edges. Also, is it easy to see that the cost of $M^{\frac{1}{2}}$ is the same as the cost of the $b$-matching. We now have to prove that $M^{\frac{1}{2}}$ is good for $\mathcal{C}_{\min}$. As we did not denote any half-edge of $M^{\frac{1}{2}}$ as special, we only need to check properties 1 and 2 of Definition 6.

Consider a 1-square $C = \{v_1, \ldots, v_4\} \in \mathcal{C}_{\min}$. From the gadgets construction we can see that the vertex capacities for $v_C, v_C^1, v_C^2$ enforce that at most three of the vertices $\{v_1, \ldots, v_4\}$ are matched with a subdivision vertex. Therefore, at least one of the vertices $\{v_1, \ldots, v_4\}$ is matched by the $b$-matching via an edge not belonging to the gadget, i.e., an external 1-edge or at least one vertex of $C$ is unmatched in $M_0''$. Therefore, Property 2 holds.

From the construction of $M^{\frac{1}{2}}$, each half-edge of $M^{\frac{1}{2}}$ corresponds to a native edge $e = \{v_i, v_{(i \mod 4)+1}\}$ of a 1-square. To prove Property 1, we need to show that each 1-square $C \in \mathcal{C}_{\min}$ is incident to at most one half-edge. We already know that at most three of the vertices $\{v_1, \ldots, v_4\}$ of $C$ are matched with a subdivision vertex. If there are three, some two of them are incident to neighboring vertices of $C$, and will be transformed into one native edge in $M^{\frac{1}{2}}$, which will result in only one half-edge of $M^{\frac{1}{2}}$ incident to $C$. If exactly two of the vertices $\{v_1, \ldots, v_4\}$ of $C$ were matched with a subdivision vertex, they yield two half-edges within $M^{\frac{1}{2}}$ only if they are incident with the opposite corners of $C$, and the corresponding diagonal has cost 2. We show that the construction of the gadgets prevents this from happening.

First, consider the case when $C$ has no 1-diagonal, see Fig. 1c. Assume, without loss of generality, that exactly the vertices $v_1, v_3$ are matched by the $b$-matching with the subdivision vertices. Then, as the capacity interval of $v_C^1$ is $[3, 3]$, and the capacity interval of $v_C$ is $[0, 1]$, $v_C$ must be matched with $v_C^1$, and $v_C^1$ must be matched with 2 of the subdivision vertices. Then, as the capacity interval of $v_C^2$ is $[3, 3]$, $v_C^2$ must be matched with 3 subdivision vertices. But that leaves one subdivision vertex unmatched, and it therefore cannot happen.

Now, consider the case when $C$ has exactly one 1-diagonal $\{v_2, v_4\}$, see Fig. 1b. Assume that exactly the vertices $v_1, v_3$ are matched by the $b$-matching with the subdivision vertices. Then, by the capacity intervals of $v_C^1$ and $v_C$, again $v_C$ must be matched with $v_C^1$, and $v_C^1$ must be matched with 2 of the subdivision vertices. Then, as the capacity interval of $v_C^2$ is $[5, 5]$, $v_C^2$ must be matched with 5 subdivision vertices. But that leaves one subdivision vertex unmatched, and it therefore cannot happen.

Each 1-square of $\mathcal{C}_{\min}$ is incident to at most one half-edge, and therefore Property 1 holds and the matching $M^{\frac{1}{2}}$ is good. This completes the proof of Claim 2. ◀

▶ **Claim 3.** *Any matching with half-edges $M^{\frac{1}{2}}$ which is good for $\mathcal{C}_{\min}$ and contains no special half-edge can be transformed into a b-matching $M_0''$ in $G_0''$ of the same cost.*

**Proof of Claim 3.** Consider a matching with half-edges $M^{\frac{1}{2}}$ which is good for $\mathcal{C}_{\min}$ and contains no special half-edge. We will construct a corresponding $b$-matching for $M^{\frac{1}{2}}$. For any edge $e = \{u, v\} \in M^{\frac{1}{2}}$ which is not a half-edge or a 1-edge of a 1-square, $e$ is also present in the graph $G_0''$, and we add $e$ to the $b$-matching. Now, consider any half-edge $\{u, v_e\} \in M^{\frac{1}{2}}$, where $e = \{u, u'\}$. From Property 1 of Definition 6, $e$ is a native edge of a square $C \in \mathcal{C}_{\min}$. In the $b$-matching, we connect $u$ with any subdivision edge neighbouring with it. Last, for any edge $\{u_i, u_j\}$ which is a 1-edge of a 1-square, we take the two edges $\{u_i, u_{(i,j)}\}$ and $\{u_{(j,i)}, u_j\}$ into the $b$-matching. From Property 2 of Definition 6, at most 3 subdivision edges corresponding to any 1-square $C \in \mathcal{C}_{\min}$ have been matched by this procedure. Moreover, if there were two or three, then some two of them must be incident to two endpoints of a 1-edge of $C$ (either a native edge, or a diagonal).

We now show how to extend this matching to a $b$-matching. For any 1-square $C \in \mathcal{C}_{\min}$ with two 1-diagonals, we match $v_C$ with the at least 9 unmatched subdivision vertices.

This completes the proof of Claim 3. ◀

If $|V(G)|$ is odd and $\mathcal{C}_{\min}$ contains only 1-edges, we also build another unweighted graph $G_1''$ from $G'$, in which we find a b-matching $M_1''$. The graph $G_1''$ is quite similar to $G_0''$. The details of constructing $G_1''$ and computing $M_1''$ are given in the full version.

From $M_1''$ we obtain a matching $M_1^{\frac{1}{2}}$ with half-edges good for $\mathcal{C}_{\min}$. If $|V(G)|$ is odd and $\mathcal{C}_{\min}$ does not contain any 2-edge, we set as $M^{\frac{1}{2}}$ that one of the matching $M^{\frac{1}{2}}$ and $M_1^{\frac{1}{2}}$ that has smaller cost. This completes the proof. ◀

▶ **Lemma 8.** *Any minimum-cost matching $M^{\frac{1}{2}}$ of $(G', c')$ that is good for $\mathcal{C}_{\min}$ satisfies $c'(M^{\frac{1}{2}}) \leq c(\mathsf{opt})/2$.*

**Proof.** Let $(G'', c'')$ denote the extension of the graph $G$, in which we add two half-edges of each edge of $G$, also those of cost 2. Each edge $e$ of $G$ has cost $c''(e) = c(e)$ in $G''$. Each half-edge of a 1-edge $e \in G$ has cost $\frac{1}{2}$ and each half-edge of a 2-edge $e \in G$ has cost 2. The cost of a matching $M$ in $G''$ is defined in the usual way as $c''(M) = \sum_{e \in M} c''(e)$. We notice that for any matching $M$ in $G''$ it holds $c''(M) = c'(M')$, where $M' = M \cap E(G')$.

To prove the lemma, we partition the edges of a fixed but arbitrary tour $\mathsf{opt}$ of minimum cost in $(G, c)$ into two perfect matchings $M_1 \cup M_2$ in $(G'', c'')$ with half-edges, each of which constitutes in $(G', c')$ a matching with half-edges, which is good for $\mathcal{C}_{\min}$.

To this end, let $\mathcal{S}_2$ denote the set of squares in $\mathcal{C}_{\min}$ such that $\mathsf{opt}$ uses two of its internal 1-edges. Similarly, let $\mathcal{S}_3$ denote the set of squares in $\mathcal{C}_{\min}$ such that $\mathsf{opt}$ uses three of its internal 1-edges. Let us note that if $\mathcal{S}_2 \cup \mathcal{S}_3 \neq \emptyset$, then partitioning $\mathsf{opt}$ into two perfect matchings might yield a matching or matchings that are not good for $\mathcal{C}_{\min}$. Therefore, for each square $C \in \mathcal{S}_2 \cup \mathcal{S}_3$, we take one of its internal 1-edges $e_C$ belonging to $\mathsf{opt}$, and split $e_s$ into two half-edges. For each such edge $e_C$, we place one of its half-edges into $M_1$ and place its other half-edge into $M_2$.

If the parities of $|\mathcal{S}_2 \cup \mathcal{S}_3|$ and $|V(G)|$ are the same, then this way we have already decomposed $\mathsf{opt}$ into two perfect matchings with half-edges $M_1$ and $M_2$. Assume, without loss of generality, that $c'(M_1) \leq c'(M_2)$. From $M_1$ we construct a matching $M'$ in $G'$. We first initialize $M' = M_1$. This way, the condition $c'(M') \leq c'(\mathsf{opt})/2$ is clearly satisfied.

However, $M'$ potentially is not a perfect matching with half-edges, as it might contain a half-edge of a diagonal of a square $C \in \mathcal{S}_2$. We can, however, replace such a half-edge with a half-edge of an edge of $C$, without increasing the cost of $M'$.

If the parities of $|\mathcal{S}_2 \cup \mathcal{S}_3|$ and $|V(G)|$ differ and opt uses a 2-edge, then we choose any such 2-edge $e \in \mathsf{opt}$ and split it into two half-edges. Otherwise, if $\mathcal{S}_2 \cup \mathcal{S}_3$ is empty, $|V(G)|$ is odd. Then any path-cycle cover of $G$ must contain at least one odd cycle $C$. We split any edges of opt which is incident to a vertex of $C$ into two half-edges. We decompose opt into two perfect matchings $M_1$ and $M_2$. Since $c'(M_1) = c'(M_2)$, we may choose that one which contains a half-edge incident to a vertex of $C$.

The remaining case is when the parities of $|\mathcal{S}_2 \cup \mathcal{S}_3|$ and $|V(G)|$ differ, each edge of opt has cost 1 and $\mathcal{S}_2 \cup \mathcal{S}_3$ is non-empty. Then we choose one square $C \in S_2 \cup S_3$ and do not split any of its edges. At least one of the perfect matchings $M_1, M_2$ from the decomposition of opt is such that it does not use two internal edges of $C$. Since again $c'(M_1) = c'(M_2)$, we may choose that one, which does not use two internal edges of $C$. ◀

## 3.3 Constructing the Multigraph

We will now construct a multigraph $G^1$ from the path-cycle cover $\mathcal{C}_{\min}$, and the minimum-cost matching $M^{\frac{1}{2}}$ that is good for $\mathcal{C}_{\min}$. We set $V(G^1) = V(G)$. The idea is to take into $G^1$ two copies of each 1-edge of $\mathcal{C}_{\min}$, and one copy of each 1-edge and of each 1/2-half-edge of $M^{\frac{1}{2}}$. However, to ensure that we will be able to color the graph at a later stage, and as we do not have the extended vertices $v_e$ in $V(G^1)$ to accommodate half-edges, we first need to modify the matching $M^{\frac{1}{2}}$ into a collection of edges $M$. The set $M$ does not have to be a matching—it may contain multiple edges incident to the same vertex, and even multiple copies of the same edge of $G$. Also, we need to ensure that the multigraph $G^1$ has at least $\frac{5}{2}\alpha_{\mathsf{opt}} - \beta_{\mathsf{opt}}$ edges of cost 1 in $(G, c)$, i.e., that the set $M$ has at least $\frac{1}{2}\alpha_{\mathsf{opt}} - \beta_{\mathsf{opt}}$ edges of cost 1 in $(G, c)$.

We start by setting $M$ to be the collection of edges and half-edges of $M^{\frac{1}{2}}$. Then we modify $M$ by executing the following sequence of steps:

**(O1)** For every 1-triangle $(u, v, w)$ of $\mathcal{C}_{\min}$ for which $\{u, v\} \in M$ and $\{w, t\} \in M$ for some $t$, we remove the edge $\{u, v\}$ from $M$, and instead we add a second copy of $\{w, t\}$ into $M$. Notice that if we perform a similar operation with the cycle of $\mathcal{C}_{\min}$ containing the vertex $t$, it will result in the third copy of $\{w, t\}$ being added to the graph.

**(O2)** For every 1-triangle $(u, v, w)$ of $\mathcal{C}_{\min}$ for which $\{u, v\} \in M$ and $w$ is unmatched in $M$, we remove the edge $\{u, v\}$ from $M$.

**(O3)** For every 1-square $(u, v, w, z)$ of $\mathcal{C}_{\min}$ for which $\{u, v\}, \{w, v_e\} \in M$ and $\{z, t\} \in M$ for some $t$, we remove $\{w, v_e\}$ from $M$, and instead we add a second copy of $\{z, t\}$ into $M$. Notice that in this case we really need half of the additional edge $\{z, t\}$, so if we perform such operation twice for $\{z, t\}$ (i.e., the component containing $t$ is also a 1-square), we have to add only one, and not two copies of $\{z, t\}$. We also perform the same operation if $\{u, v\}$ is a diagonal, and not a native edge of the square.

**(O4)** For every 1-square $(u, v, w, z)$ of $\mathcal{C}_{\min}$ for which $\{u, v\}, \{w, v_e\} \in M$ and $z$ is unmatched in $M$, we remove the half-edge $\{w, v_e\}$ from $M$. As before, we perform the same operation if $\{u, v\}$ is a diagonal, and not a native edge of the square.

**(O5)** For every 1-square $(u, v, w, z)$ of $\mathcal{C}_{\min}$ for which $\{u, v_e\} \in M$ and no other edge or diagonal of the square (or its half) is in $M$, we remove $\{u, v_e\}$ from $M$ and instead we add $\{u, v\}$ into $M$ where $e = \{u, v\}$.

**(O6)** If the matching $M$ contains a special half-edge $\{u, v_e\}$, then we add $e$ to $M$.

After these operations, there are no half-edges left in $M$. We now construct the multigraph $G^1$ by setting $V(G^1) = V(G)$, and by adding to $G^1$ two copies of each 1-edge of $\mathcal{C}_{\min}$ and all edges of $M$. We can show a lower bound on the number of edges of $G^1$.

▶ **Lemma 9.** *The multigraph $G^1$ has at least $\frac{5}{2}\alpha_{\mathsf{opt}} - \beta_{\mathsf{opt}}$ edges which are 1-edges of $(G, c)$.*

**Proof.** The minimum-cost path-cycle cover $\mathcal{C}_{\min}$ contains at least $\alpha_{\mathsf{opt}}$ edges of cost 1 in $(G, c)$. Therefore, two copies of $\mathcal{C}^1_{\min}$ contain at least $2\alpha_{\mathsf{opt}}$ edges. The matching $M^{\frac{1}{2}}$ has cost at most $c(opt)/2$. Therefore, the number of 1-edges in $M^{\frac{1}{2}}$ is at least $\alpha_{\mathsf{opt}}/2$, where half-edges count as half of an edge each. We further have that $\beta_{\mathsf{opt}} \geq \beta_M$, where $\beta_M$ denotes the number of basic verices unmatched in $M^{\frac{1}{2}}$.

The only modifications that decrease the number of edges in $G^1$ are operations 2 and 4. However, for each triangle or square for which we remove one 1-edge or 1/2-half-edge, we can uniquely charge it to an unmatched basic vertex. Thus, the number of such deletions is at most $\beta_M$ and can be charged against $\beta_{\mathsf{opt}}$. Consequently, we always have at least $\frac{5}{2}\alpha_{\mathsf{opt}} - \beta_{\mathsf{opt}}$ edges in the resulting multigraph $G^1$. ◀

The multigraph $G^1$ can be essentially path-3-colored using the path-3-coloring procedure by Dudycz et al. [9]. The multigraph colored by Dudycz et al. [9] is built from two copies of a maximum-cost cycle cover and a maximum cost perfect matching. Several not very serious modifications are needed in order to deal with double and triple edges of $M$; note that the existence of such edges means that some vertices in $G^1$ have degree greater than 5. Again, details are deferred to the full version of the paper.

## 4 A New 8/7-Approximation Algorithm for (1,2)-TSP

The 8/7-approximation algorithm is quite similar to the algorithm with an approximation factor of 7/6. Instead of a minimum-cost path-cycle cover $\mathcal{C}_{\min}$ of $(G, c)$ we use a minimum-cost *triangle-free* cycle cover $\mathcal{C}^t_{\min}$. We also compute a minimum-cost matching $M^{\frac{1}{2}}$ with half-edges with additional properties. To obtain an 8/7-approximation, $M^{\frac{1}{2}}$ has to additionally satisfy the condition that for every 1-hexagon $C$ from $\mathcal{C}^t_{\min}$ at least one of the vertices of $C$ must be incident to an external edge of $M^{\frac{1}{2}}$ or be unmatched in $M^{\frac{1}{2}}$. Next, we build a multigraph $G^1$ that consists of three copies of $\mathcal{C}^t_{\min}$ and one copy of $M^{\frac{1}{2}}$. We do some flipping of edges and half-edges and path-4-color the multigraph $G^1$. Path-4-coloring is based on the same ideas as path-3-coloring but is a little more complicated. We defer the details to the full version of this paper.

## References


1 Eric Angel, Evripidis Bampis, and Laurent Gourvès. Approximating the Pareto curve with local search for the bicriteria TSP(1,2) problem. *Theoret. Comput. Sci.*, 310(1):135–146, 2004.

2 Eric Angel, Evripidis Bampis, Laurent Gourvès, and Jérôme Monnot. (Non)-approximability for the multi-criteria TSP(1,2). In *Proc. FCT 2005*, volume 3623 of *Lecture Notes Comput. Sci.*, pages 329–340, 2005.

3 Sanjeev Arora. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. In *Proc. FOCS 1996*, pages 2–11. IEEE Computer Society, 1996. doi:10.1109/SFCS.1996.548458.

4 Piotr Berman and Marek Karpinski. 8/7-approximation algorithm for $(1, 2)$-TSP. In *Proc. SODA 2006*, pages 641–648, 2006.

**5** M. Bläser and L. Shankar Ram. An improved approximation algorithm for TSP with distances one and two. In *Proc. FCT 2005*, volume 3623 of *LNCS*, pages 504–515. Springer, 2005. doi:10.1007/11537311_44.

**6** Markus Bläser. A 3/4-approximation algorithm for maximum ATSP with weights zero and one. In *Proc. APPROX-RANDOM 2004*, volume 3122 of *Lecture Notes Comput. Sci.*, pages 61–71. Springer, 2004. doi:10.1007/978-3-540-27821-4_6.

**7** Markus Bläser and Bodo Manthey. Approximating maximum weight cycle covers in directed graphs with weights zero and one. *Algorithmica*, 42(2):121–139, 2005.

**8** Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem, 1976. TR 388, Graduate School of Industrial Administration, CMU.

**9** Szymon Dudycz, Jan Marcinkowski, Katarzyna Paluch, and Bartosz Rybicki. *A 4/5-Approximation Algorithm for the Maximum Traveling Salesman Problem*, volume 10328 of *Lecture Notes Comput. Sci.*, pages 173–185. Springer, 2017. doi:10.1007/978-3-319-59250-3_15.

**10** W. Fernandez de la Vega and M. Karpinski. On the approximation hardness of dense TSP and other path problems. *Inform. Process. Lett.*, 70(2):53–55, 1999.

**11** Harold N. Gabow. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In *Proc. STOC 1983*, pages 448–456, 1983.

**12** Mikael Gast, Matthias Hauptmann, and Marek Karpinski. Approximability of TSP on power law graphs. *CoRR*, abs/1509.03976, 2015. arXiv:1509.03976.

**13** David B. Hartvigsen. Personal communication.

**14** David B. Hartvigsen. *Extensions of Matching Theory.* PhD thesis, Carnegie-Mellon University, 1984.

**15** Michael Held and Richard M. Karp. The traveling-salesman problem and minimum spanning trees. *Operations Res.*, 18:1138–1162, 1970.

**16** Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, pages 85–103. Plenum, New York, 1972.

**17** Marek Karpinski, Michael Lampis, and Richard Schmied. New inapproximability bounds for TSP. In *Proc. ISAAC 2013*, volume 8283 of *Lecture Notes Comput. Sci.*, pages 568–578. Springer, 2013. doi:10.1007/978-3-642-45030-3_53.

**18** Marek Karpinski and Richard Schmied. On approximation lower bounds for TSP with bounded metrics. *Electr. Colloquium Computat. Complexity*, 19:8, 2012.

**19** Marek Karpinski and Richard Schmied. Approximation hardness of graphic tsp on cubic graphs. *RAIRO Oper. Res.*, 49(4):651–668, 2015.

**20** László Lovász and Michael D. Plummer. *Matching theory.* AMS Chelsea Publishing, Providence, RI, 2009. Corrected reprint of the 1986 original.

**21** Matthias Mnich and Tobias Mömke. Improved integrality gap upper bounds for traveling salesperson problems with distances one and two. *European J. Oper. Res.*, 266(2):436–457, 2018.

**22** Katarzyna Paluch. Maximum ATSP with weights zero and one via half-edges. *Theory Comput. Syst.*, 62(2):319–336, 2018.

**23** Katarzyna Paluch, Khaled Elbassioni, and Anke van Zuylen. Simpler approximation of the maximum asymmetric traveling salesman problem. In *Proc. STACS 2012*, volume 14 of *Leibniz Int. Proc. Informatics*, pages 501–506, 2012.

**24** Christos H. Papadimitriou and Mihalis Yannakakis. The traveling salesman problem with distances one and two. *Math. Oper. Res.*, 18(1):1–11, 1993.

**25** Jiawei Qian, Frans Schalekamp, David P. Williamson, and Anke van Zuylen. On the integrality gap of the subtour LP for the 1,2-TSP. In *Proc. LATIN 2012*, volume 7256 of *Lecture Notes Comput. Sci.*, pages 606–617, 2012.

26 András Sebő and Jens Vygen. Shorter tours by nicer ears: 7/5-approximation for the graph-TSP, 3/2 for the path version, and 4/3 for two-edge-connected subgraphs. *Combinatorica*, 34(5):597–629, 2014.

27 Luca Trevisan. When Hamming meets Euclid: the approximability of geometric TSP and Steiner tree. *SIAM J. Comput.*, 30(2):475–485, 2000.

28 Sundar Vishwanathan. An approximation algorithm for the asymmetric travelling salesman problem with distances one and two. *Inform. Process. Lett.*, 44(6):297–302, 1992.

29 David P. Williamson. Analysis of the Held-Karp heuristic for the traveling salesman problem. Master's thesis, MIT, 1990.