# A Constant-Factor Approximation Algorithm for the Asymmetric Traveling Salesman Problem

(Extended Abstract)

Ola Svensson[*] Jakub Tarnawski[†] László A. Végh[†]

November 6, 2017

**Abstract**

We give a constant-factor approximation algorithm for the asymmetric traveling salesman problem. Our approximation guarantee is analyzed with respect to the standard LP relaxation, and thus our result confirms the conjectured constant integrality gap of that relaxation.

Our techniques build upon the constant-factor approximation algorithm for the special case of node-weighted metrics. Specifically, we give a generic reduction to structured instances that resemble but are more general than those arising from node-weighted metrics. For those instances, we then solve Local-Connectivity ATSP, a problem known to be equivalent (in terms of constant-factor approximation) to the asymmetric traveling salesman problem.

**Keywords:** approximation algorithms, asymmetric traveling salesman problem, combinatorial optimization, linear programming

**This extended abstract gives an overview of the results and techniques. To keep it short and to emphasize the ideas, we have simplified the statements and used less precise bounds than in the full version of the paper. Formal proofs are deferred to the full version, which may be found at https://arxiv.org/abs/1708.04215.**

[*]École Polytechnique Fédérale de Lausanne. Email: *firstname.lastname@epfl.ch*. Supported by ERC Starting Grant 335288-OptApprox.

[†]London School of Economics. Email: *l.vegh@lse.ac.uk*. Supported by EPSRC First Grant EP/M02797X/1.

# 1 Introduction

The traveling salesman problem — to find the shortest tour visiting $n$ given cities — is one of the best-known NP-hard optimization problems.

Without any assumptions on the distances, a simple reduction from the problem of deciding whether a graph is Hamiltonian shows that it is NP-hard to approximate the shortest tour to within any factor. Therefore it is common to relax the problem by allowing the tour to visit cities more than once. This is equivalent to assuming that the distances satisfy the triangle inequality: the distance from city $i$ to $k$ is no larger than the distance from $i$ to $j$ plus the distance from $j$ to $k$. All results mentioned and proved in this paper refer to this setting.

If we further assume the distances to be symmetric, then Christofides' classic algorithm from 1976 [Chr76] is guaranteed to find a tour of length at most $3/2$ times the optimum. Improving this approximation guarantee is a notorious open question in approximation algorithms. There has been a flurry of recent progress in the special case when distances are given as unweighted shortest path metrics [GSS11, MS16, Muc12, SV14]. However, even though the standard linear programming (LP) relaxation is conjectured to approximate the optimum within a factor of $4/3$, it remains an elusive problem to improve upon Christofides' algorithm.

If we do not restrict ourselves to symmetric distances, we obtain the more general *asymmetric* traveling salesman problem (ATSP). Compared to the symmetric setting, the gap in our understanding is much larger, and the current algorithmic techniques have failed to give *any* constant approximation guarantee. This is intriguing especially since the standard LP relaxation, also known as the *Held-Karp lower bound*, is conjectured to approximate the optimum to within a small constant. In fact, it is only known that its integrality gap[1] is at least 2 [CGK06].

The first approximation algorithm for ATSP was given by Frieze, Galbiati and Maffioli [FGM82], achieving an approximation guarantee of $\log_2(n)$. Their elegant "repeated cycle cover" approach was refined in several papers [Blä08, KLSS05, FS07], but there was no *asymptotic* improvement in the approximation guarantee until the more recent $O(\log n / \log\log n)$-approximation algorithm by Asadpour et al. [AGM+10]. They introduced a new and influential approach to ATSP based on a connection to the graph-theoretic concept of thin spanning trees. This has further led to improved algorithms for special cases of ATSP, such as graphs of bounded genus [GS11]. Moreover, Anari and Oveis Gharan recently exploited this connection to significantly improve the best known upper bound on the integrality gap of the standard LP relaxation to $O(\text{poly}\log\log n)$ [AG15]. This implies an efficient algorithm for estimating the optimal value of a tour within a factor $O(\text{poly}\log\log n)$ but, as their arguments are non-constructive, no approximation algorithm for finding a tour of matching guarantee.

Around the same time, an alternative approach was introduced by Svensson [Sve15]. It reduces the task of approximating ATSP to a seemingly easier problem called Local-Connectivity ATSP. The paper [Sve15] also gave an algorithm for Local-Connectivity ATSP restricted to the special case of node-weighted metrics, implying a constant-factor approximation algorithm for that special case. We have generalized this to graphs with at most two different edge weights in subsequent work [STV16]. In this paper, we build upon and generalize both of these results to give a constant-factor approximation algorithm for all metrics.

**Theorem 1.1.** *There is a polynomial-time algorithm for ATSP that returns a tour of value at most a constant times the Held-Karp lower bound.*

[1]Recall that the integrality gap is defined as the maximum ratio between the optimum values of the exact (integer) formulation and of its relaxation.

We remark that we have not optimized the constant of the approximation guarantee, instead favoring simplicity. However, we believe that further developments are needed to get close to the lower bound of 2 on the integrality gap [CGK06] and the inapproximability of 75/74 [KLS13].

**Outline.** The paper [Sve15] has introduced the problem *Local-Connectivity ATSP* and showed that it is equivalent (in terms of constant-factor approximation) to the asymmetric traveling salesman problem. Further, it gave an efficient solution to Local-Connectivity ATSP for node-weighted graphs. In [STV16] we gave a solution for graphs with two different edge weights. This, however, turned out to be technically challenging. In fact, it is unclear if the same approach can be extended even to a fixed number of different edge weights.

In the current paper we take a different route. Instead of trying to directly tackle Local-Connectivity ATSP in arbitrary weighted graphs, the first part of our argument uses a sequence of natural reductions to reduce the problem of approximating ATSP in general to that of approximating ATSP on special, structured instances called *vertebrate pairs*. These instances enjoy properties that make them amenable for Local-Connectivity ATSP. The reduction of the first part proceeds in multiple stages:

– We first solve the standard Held-Karp LP relaxation for ATSP. By applying the uncrossing technique on the optimal dual solution, we are able to show that we can focus on *laminarly-weighted ATSP instances* – ones where the edge weights are defined by a laminar family of vertex subsets. We discuss this in Section 2.

– In the next step, we define a natural recursive algorithm that solves smaller instances obtained by contracting tight vertex sets in the laminar family. The analysis of this approach shows that it works as long as the contraction of a set causes a large decrease in the LP value. We refer to such sets as *reducible*. Thus, we reduce the problem further to *irreducible instances*: ones that do not contain any such reducible set. This is outlined in Section 3.

– Given an irreducible instance, we can utilize its structure together with the constant-factor approximation algorithm for node-weighted instances [Sve15] to obtain a special subtour that we call the *backbone*. Intuitively, the backbone visits most of the vertices in the instance. In particular, it is required to visit at least one vertex in each non-singleton set in the laminar family. We call an instance together with a backbone a *vertebrate pair*. In Section 4 we outline the reduction that shows that if we can deal with vertebrate pairs, then we can deal with irreducible instances.

In each of the above stages, we prove a theorem of the form: if there is a constant-factor approximation for ATSP on more structured instances, then there is a constant-factor approximation for ATSP on less structured instances. For instance, an algorithm for irreducible instances implies an algorithm for laminarly-weighted instances. One can also think of making a stronger and stronger assumption on the instance without loss of generality, making it increasingly resemble a node-weighted metric. The second part, i.e., solving Local-Connectivity ATSP on vertebrate pairs, is described in Section 5.

## 2 Held-Karp Relaxation and Reduction to Laminarly-Weighted ATSP

It will be convenient to define ATSP in terms of its *graphic formulation*:

**Definition 2.1.** The input for ATSP is a pair $(G, w)$, where $G = (V, E)$ is a strongly connected directed graph (digraph) and $w$ is a nonnegative weight function defined on the edges. The objective is to find a closed walk of minimum weight that visits every vertex at least once.

| LP$(G,w)$ | DUAL$(G,w)$ |
|---|---|
| $\min \sum_{e\in E} w(e)x(e)$ | $\max \sum_{\emptyset\neq S\subset V} 2\cdot y_S$ |
| s.t. $x(\delta^+(v)) = x(\delta^-(v))$ for $v\in V$, | s.t. $\sum_{S:\,(u,v)\in\delta(S)} y_S + \alpha_u - \alpha_v \leqslant w(u,v)$ for $(u,v)\in E$, |
| $x(\delta(S)) \geqslant 2$ for $\emptyset\neq S\subsetneq V$, | $y \geqslant 0.$ |
| $x \geqslant 0.$ | |

**Figure 1:** The Held-Karp relaxation LP$(G,w)$ and its dual DUAL$(G,w)$. For a function $f : A \to \mathbb{R}_+$ and a subset $B \subseteq A$, we use the notation $f(B) = \sum_{a\in B} f(a)$. In particular, $x(F) = \sum_{e\in F} x(e)$ for an edge set $F$.

Without loss of generality one could assume that $G$ is a complete digraph. However, for our reductions, it will be important that $G$ may not be complete.

A closed walk that visits every vertex at least once is equivalent to an Eulerian set[2] of edges that connects the graph. This brings us to the well-known Held-Karp relaxation LP$(G,w)$ shown on the left of Figure 1. It has a variable $x(e) \geqslant 0$ for every edge $e \in E$, and the intended solution is that $x(e)$ should equal the number of times $e$ is used in the tour. Here, $\delta^+(S)$ denotes the outgoing edges of a vertex set $S$, $\delta^-(S)$ denotes the incoming edges, and $\delta(S)$ is the union of both. The optimum value of this LP is called the *Held-Karp lower bound*. The first set of constraints says that the in-degree should equal the out-degree for each vertex, i.e., the solution should be Eulerian. The second set of constraints forbids the existence of *subtours*, i.e., Eulerian components that are connected but do not connect the entire graph. They are called subtour elimination constraints.

The dual linear program DUAL$(G,w)$, shown on the right of Figure 1, is obtained by associating variables $(\alpha_v)_{v\in V}$ and $(y_S)_{\emptyset\neq S\subset V}$ with the first and second set of constraints of LP$(G,w)$, respectively.

Now consider primal optimal and dual optimal solutions $x$ and $(\alpha, y)$, respectively. By a standard uncrossing argument (see e.g. [CFN85] for an early application of this technique to the Held-Karp relaxation of the symmetric traveling salesman problem), we may assume that the support $\mathcal{L} = \{S : y_S > 0\}$ of $y$ is a *laminar* family of vertex sets, i.e., any two sets in $\mathcal{L}$ are either disjoint or one is a subset of the other. We may further assume that every edge $e \in E$ has $x(e) > 0$ (since we can always solve the smaller instance where we disregard all edges $e$ with $x(e) = 0$). Hence, complementarity slackness gives the following:

- For every $(u,v) \in E$, we have $w(u,v) = \sum_{S\in\mathcal{L}:\, e\in\delta(S)} y_S + \alpha_u - \alpha_v$.
- For every $S \in \mathcal{L}$, we have $x(\delta(S)) = 2$.

We refer to a vertex set $S \subseteq V$ with $x(\delta(S)) = 2$ (and thus $x(\delta^+(S)) = x(\delta^-(S)) = 1$) as a *tight set* (with respect to $x$). Notice that the first condition says that the weights of the edges are determined by the dual solution $(\alpha, y)$. Now consider the weight function $w'$ induced by the dual solution where we disregard the $\alpha$-variables: $w'(u,v) = \sum_{S\in\mathcal{L}:\, e\in\delta(S)} y_S$. A key observation is that $w'$ is equivalent to $w$, in the sense that it assigns the same weight to any Eulerian solution. We can therefore consider the weight function $w'(u,v) = w(u,v) - \alpha_u + \alpha_v$ that is determined by the vector $(y_S)_{S\in\mathcal{L}}$. This motivates the following definition (see also Figure 2 for an example):

**Definition 2.2.** A tuple $\mathcal{I} = (G, \mathcal{L}, x, y)$ is called a *laminarly-weighted ATSP instance* if $G$ is a strongly connected digraph, $\mathcal{L}$ is a *laminar* family of vertex subsets, $x$ is a feasible solution to the LP$(G, \mathbf{0})$, and $y : \mathcal{L} \to \mathbb{R}_+$. We further require that $x_e > 0$ for every $e \in E$ and that every set in $\mathcal{L}$ be tight with respect to $x$. We define the *induced weight function* $w_\mathcal{I} : E \to \mathbb{R}_+$ as $w_\mathcal{I}(e) = \sum_{S\in\mathcal{L}:\, e\in\delta(S)} y_S$ for every $e \in E$.

[2]By an edge set, we always mean an edge multiset: the same edge can be present in multiple copies.

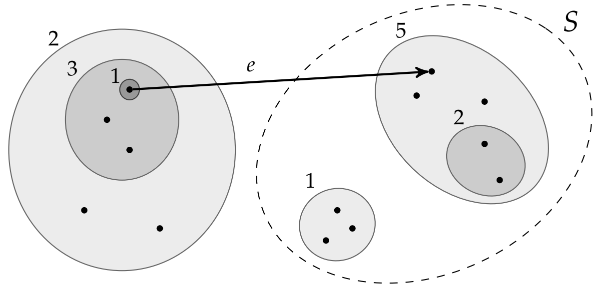


**Figure 2:** An example of a laminarly-weighted ATSP instance $\mathcal{I}$. The sets of the laminar family are shown in gray, with their $y$-values written on their borders. We depict a single edge $e$ that crosses four sets in the laminar family and has $w(e) = 1 + 3 + 2 + 5 = 11$. Also, if we let $S$ be the dashed set, then $\text{value}(S) = 2 \cdot (1 + 2 + 5) = 16$ and $\text{value}(\mathcal{I}) = 2 \cdot (1 + 1 + 2 + 2 + 3 + 5) = 28$.

It is worth noting that whenever both $(u, v)$ and $(v, u)$ are present in $E$, then $w_{\mathcal{I}}(u, v) = w_{\mathcal{I}}(v, u)$. Based on the above ideas, we can prove the following theorem.

**Theorem 2.3.** *Assume we have a polynomial-time $\alpha$-approximation algorithm with respect to the Held-Karp relaxation for every laminarly-weighted ATSP instance. Then there is a polynomial-time $\alpha$-approximation algorithm with respect to the Held-Karp relaxation for the general ATSP problem.*

We remark that the concept of laminarly-weighted instances generalizes the special case of *node-weighted instances*. Indeed, node-weighted instances are those laminarly-weighted instances $\mathcal{I}$ where the laminar family $\mathcal{L}$ consists only of singletons. Thus for any edge $(u, v) \in E$ we have $w_{\mathcal{I}}(u, v) = y_{\{u\}} + y_{\{v\}}$ (the numbers $y_{\{v\}}$ for $v \in V$ are called node weights). For that special case, [Sve15] gave a $(27 + \varepsilon)$-approximation algorithm for any $\varepsilon > 0$.

For future reference, we refer to the Held-Karp lower bound as the *value* of the instance $\mathcal{I}$ and we define it as a function of the dual: $\text{value}(\mathcal{I}) := 2 \sum_{S \in \mathcal{L}} y_S$. For a given subset $S \subseteq V$ of the vertices, it will also be convenient to localize the contribution of the dual variables contained strictly inside $S$: we let $\text{value}(S) = 2 \cdot \sum_{R \in \mathcal{L}: R \subsetneq S} y_S$. See Figure 2 for examples of these definitions.

## 3 Reduction to Irreducible Instances

By the previous section, we may assume that we are given a laminarly-weighted instance $\mathcal{I} = (G, \mathcal{L}, x, y)$ as input. Now an important observation for our approach is the following: since each set $S \in \mathcal{L}$ is tight, we can obtain a smaller instance $\mathcal{I}/S = (G', \mathcal{L}', x', y')$ by contracting the set $S$ into a new vertex $s$. We get $G'$, $\mathcal{L}'$ and $x'$ in the natural way (see Figure 3 for an example and the full version for the formal definition). For instance, $\mathcal{L}'$ is obtained from $\mathcal{L}$ by removing sets $R \subseteq S$ and adding the singleton $\{s\}$. To complete the description of $\mathcal{I}/S$, it remains to specify how to set the new value $y'_{\{s\}}$. To that end, we define the "distance" functions $d_S$ and $D_S$. For $u, v \in S$, define $d_S(u, v)$ to be the minimum weight of a path from $u$ to $v$ (inside $S$) and let $D_S(u, v) = \sum_{R \in \mathcal{L}:\, u \in R \subsetneq S} y_R + d_S(u, v) + \sum_{R \in \mathcal{L}:\, v \in R \subsetneq S} y_R$. We now set

$$y'_{\{s\}} = y_S + \max_{u \in S_{\text{in}}, v \in S_{\text{out}}} D_S(u, v)/2\,,$$

where $S_{\text{in}}$ and $S_{\text{out}}$ denote those vertices of $S$ that have an incoming edge from outside of $S$ or an outgoing edge to outside of $S$, respectively. This completes the description of $\mathcal{I}/S$.

One can show that every vertex in $S_{\text{in}}$ is connected to every vertex in $S_{\text{out}}$ by a directed path inside $S$, and that $D_S(u, v)$ can be upper-bounded by the value of $S$:

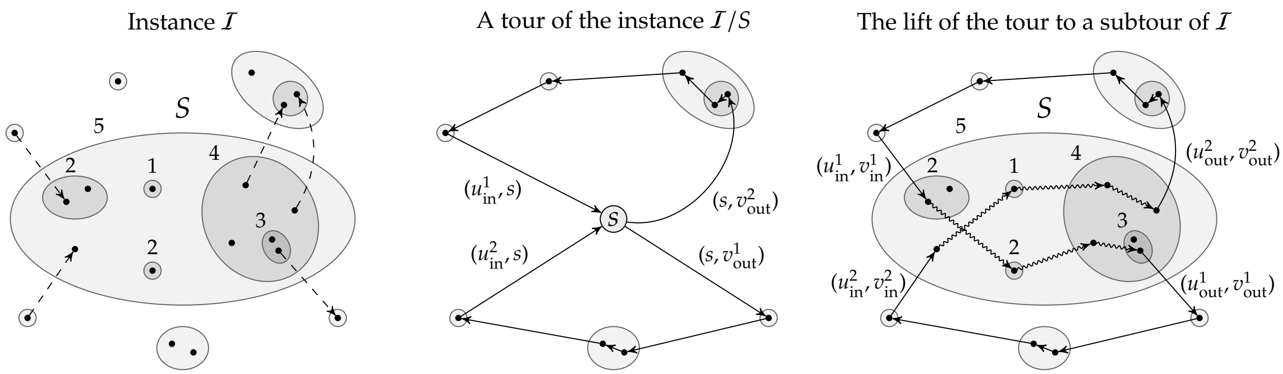


**Figure 3:** An example of the contraction of a tight set $S$ and the lift of a tour. Only $y$-values of the sets $R \in \mathcal{L} : R \subseteq S$ are depicted. On the left, only edges that have one endpoint in $S$ are shown. These are exactly the edges that are incident to $s$ in the contracted instance. In the center, a tour of $\mathcal{I}/S$ is illustrated, and on the right we depict the lift of that tour.

**Lemma 3.1.** *For every $u \in S_{in}$ and $v \in S_{out}$ there is a path from $u$ to $v$ inside $S$ that crosses each laminar set $R \subsetneq S$ at most $2 - |R \cap \{u, v\}|$ times. Consequently, $D_S(u, v) \leqslant \text{value}_{\mathcal{I}}(S)$.*

The intuition of the definition of $D_S$ and the setting of $y'_{\{s\}}$ is as follows. After contracting $S$, all sets of the laminar family are still present in the contracted instance except for the sets contained in $S$. Now, after finding a tour in the contracted instance, we *lift* it back to a subtour in the original instance. We obtain this subtour by, for each visit of the tour to $s$ on some edges $(u^i_{\text{in}}, s), (s, v^i_{\text{out}})$, replacing $(u^i_{\text{in}}, s), (s, v^i_{\text{out}})$ by the corresponding edges (i.e., by their preimages) $(u^i_{\text{in}}, v^i_{\text{in}}), (u^i_{\text{out}}, v^i_{\text{out}})$ of $G$ together with a minimum-weight path inside $S$ from $v^i_{\text{in}}$ to $u^i_{\text{in}}$ (depicted by swirly edges in Figure 3). The change in weight incurred by this operation (for the $i$-th visit) is

$$\underbrace{2 \cdot y_S + D_S(v^i_{\text{in}}, u^i_{\text{out}})}_{\text{the weight incurred "inside" } S \text{ in } \mathcal{I}} \quad - \quad \underbrace{2 \cdot y'_{\{s\}}}_{\text{the weight of visiting } s \text{ in } \mathcal{I}/S} . \tag{3.1}$$

Indeed, consider the example depicted in Figure 3. In each visit to $s$ in the tour of $\mathcal{I}/S$, the set $\{s\}$ is crossed twice, incurring a weight of $2 \cdot y'_{\{s\}}$. Now, say in the first visit to $s$, the lift of the tour to $\mathcal{I}$ incurs the following weight instead of $2 \cdot y'_{\{s\}}$:

$$\underbrace{y_S}_{=5} + \underbrace{\sum_{R \in \mathcal{L}:\, v^1_{\text{in}} \in R \subsetneq S} y_R}_{=2} + \underbrace{d_S(v^1_{\text{in}}, u^1_{\text{out}})}_{=2+2\cdot 2+4+3} + \underbrace{\sum_{R \in \mathcal{L}:\, u^1_{\text{out}} \in R \subsetneq S} y_R}_{=3+4} + \underbrace{y_S}_{=5} = \underbrace{2 \cdot y_S + D_S(v^1_{\text{in}}, u^1_{\text{out}})}_{=32} .$$

The selection of $y'_{\{s\}} = y_S + \max_{u \in S_{\text{in}}, v \in S_{\text{out}}} D_S(u, v)/2$ is such as to guarantee that (3.1) is never positive, which implies the the following:

**Lemma 3.2.** *Let $T$ be a tour of the instance $\mathcal{I}/S$. Then the lift $F$ of $T$ satisfies $w_{\mathcal{I}}(F) \leqslant w_{\mathcal{I}/S}(T)$.*

The lift $F$ is not guaranteed to be a tour in $\mathcal{I}$: it visits all the vertices in $V \setminus S$ but only a subset of the vertices in $S$ (in the example in Figure 3, there are two vertices not visited). However, if we can obtain a "cheap" $F$, then we can complete it inside $S$ using our remaining budget. This idea is formalized in a recursive framework. By definition of the contraction we have $\mathrm{value}(\mathcal{I}/S) = \mathrm{value}(\mathcal{I}) - \left(\mathrm{value}_{\mathcal{I}}(S) - \max_{u\in S_{\mathrm{in}}, v \in S_{\mathrm{out}}} D_S(u,v)\right)$. Recall from Lemma 3.1 that $\max_{u\in S_{\mathrm{in}}, v\in S_{\mathrm{out}}} D_S(u,v) \leqslant \mathrm{value}_{\mathcal{I}}(S)$, and therefore the value cannot increase after contraction: $\mathrm{value}(\mathcal{I}/S) \leqslant \mathrm{value}(\mathcal{I})$. Any slack in this inequality can be used to pay for completing the lift $F$ into a tour of the original instance $\mathcal{I}$. This motivates the following definition.

**Definition 3.3.** We say that a set $S \in \mathcal{L}$ is *reducible* if $\max_{u\in S_{\mathrm{in}}, v\in S_{\mathrm{out}}} D_S(u,v) < 3/4\,\mathrm{value}(S)$. An instance $\mathcal{I}$ is called *irreducible* if no set $S \in \mathcal{L}$ is reducible.

Note that if we contract a reducible set $S$, then we are guaranteed that the value decreases by at least $1/4\,\mathrm{value}(S)$. This decrease is sufficient to employ our recursive strategy and to reduce the problem of approximating ATSP to that of approximating irreducible instances:

**Theorem 3.4.** *Let $\mathcal{A}$ be a polynomial-time $\rho$-approximation algorithm for irreducible instances. Then there is a polynomial-time $8\rho$-approximation algorithm for general laminarly-weighted instances.*

*Proof sketch.* Consider a laminarly-weighted instance $\mathcal{I}$. First, if there are no reducible sets in $\mathcal{I}$, then we can just use $\mathcal{A}$ to find a $\rho$-approximate tour of $\mathcal{I}$. Otherwise we proceed recursively as follows:

1. Select a minimal (inclusion-wise) set $S \in \mathcal{L}$ that is reducible.
2. Recursively find a tour $T$ of $\mathcal{I}/S$ of weight at most $8\rho\,\mathrm{value}(\mathcal{I}/S) \leqslant 8\rho\,\mathrm{value}(\mathcal{I}) - 2\rho\,\mathrm{value}(S)$.
3. Use $\mathcal{A}$ to complete the lift of $T$ into a tour of $\mathcal{I}$.

By Lemma 3.2 we have that the weight of the lift of $T$ is no larger than that of $T$ and so it is at most $8\rho\,\mathrm{value}(\mathcal{I}) - 2\rho\,\mathrm{value}(S)$. Therefore, the statement will follow if we can show how to use $\mathcal{A}$ to find a set $F$ of edges with $w(F) \leqslant 2\rho\,\mathrm{value}(S)$ such that $F$ plus the lift of $T$ form a tour of $\mathcal{I}$.

We now argue that this is possible under the following simplifying assumption: the restriction of $x$ to the smaller instance $\mathcal{I}'$ obtained by only considering the vertices in $S$ is a feasible solution to the Held-Karp relaxation of $\mathcal{I}'$. With this assumption, $\mathcal{I}'$ is a laminarly-weighted instance with $\mathrm{value}(\mathcal{I}') = \mathrm{value}(S)$. It is furthermore an irreducible instance, since $S$ was selected to be a *minimal* reducible set. We can thus use $\mathcal{A}$ to find a tour $F$ of $\mathcal{I}'$ with $w(F) \leqslant \rho\,\mathrm{value}(S)$. The lift of $T$ plus $F$ form a tour of $\mathcal{I}$ and so the statement follows, under this simplifying assumption. In general, the assumption is not true and, in the full version, we define an operation of *inducing* on the set $S$ which results in an instance of value equal to $2 \cdot \mathrm{value}(S)$ (instead of $\mathrm{value}(S)$ as above). While this loses a factor of 2, it allows us to find a set $F$ of weight $w(F) \leqslant 2\rho\,\mathrm{value}(S)$ (which is still sufficient) such that the lift of $T$ plus $F$ form a tour of $\mathcal{I}$. ☐

# 4 Reduction to Vertebrate Pairs

Theorem 3.4 shows that it suffices to find a constant-factor approximation algorithm for ATSP for any irreducible instance $\mathcal{I}$. Recall that this means that for every set $S \in \mathcal{L}$ there are two vertices $u^S_{\max}, v^S_{\max} \in S$ with $D_S(u^S_{\max}, v^S_{\max}) \geqslant 3/4\,\mathrm{value}(S)$. Informally, the shortest path from $u^S_{\max}$ to $v^S_{\max}$ crosses a large (weighted) fraction of all laminar sets inside $S$. (Indeed, if we had $D_S(u^S_{\max}, v^S_{\max}) = \mathrm{value}(S)$, then it would cross all laminar sets inside $S$.) Our objective is to use this property, together with the constant-factor approximation algorithm for node-weighted instances [Sve15], to construct a low-weight subtour $B$ that does not necessarily visit every vertex, but crosses every non-singleton set of $\mathcal{L}$.

**Definition 4.1.** We say that an instance $\mathcal{I} = (G, \mathcal{L}, x, y)$ and a subtour $B$ form a *vertebrate pair* if every $S \in \mathcal{L}$ with $|S| \geqslant 2$ is crossed by $B$, i.e., $\delta(S) \cap B \neq \emptyset$. The set $B$ is referred to as the *backbone* of the instance.

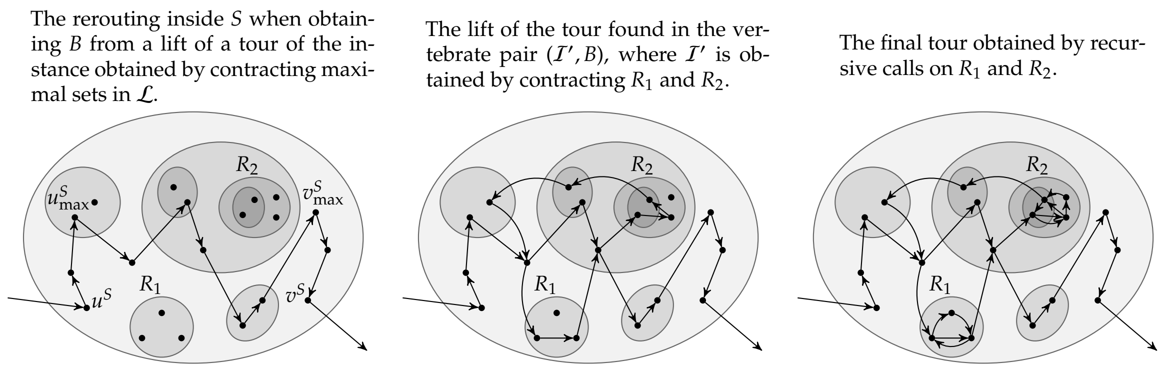


**Figure 4:** An illustration of the steps in the proof of Theorem 4.2. Only one maximal set $S \in \mathcal{L}$ is shown.

Our main result of this section further reduces the problem of approximating ATSP in general to that of approximating ATSP on vertebrate pairs.

**Theorem 4.2.** *Let $\mathcal{A}$ be a polynomial-time algorithm that, given a vertebrate pair $(\mathcal{I}', B)$, returns a tour of $\mathcal{I}'$ with weight at most $\beta\,(\text{value}(\mathcal{I}') + w(B))$. Then there is a polynomial-time $64\beta$-approximation algorithm for irreducible instances.*

*Proof sketch.* Consider an irreducible instance $\mathcal{I}$. We begin by contracting all maximal sets in $\mathcal{L}$ to obtain an instance $\mathcal{I}'$. As noted in Section 3, we have $\text{value}(\mathcal{I}') \leqslant \text{value}(\mathcal{I})$. Furthermore, the new instance is node-weighted, since all laminar sets are now singletons. Therefore we can use the node-weighted algorithm [Sve15] to find a tour $T$ of $\mathcal{I}'$ with $w_{\mathcal{I}'}(T) \leqslant 28\,\text{value}(\mathcal{I}')$.

Now we wish to obtain a subtour in $\mathcal{I}$ from $T$. Thus we perform the lift operation, just as in the previous section, to get a subtour $B'$. By Lemma 3.2, the lift $B'$ satisfies $w_{\mathcal{I}}(B') \leqslant w_{\mathcal{I}'}(T)$. Thus we have $w_{\mathcal{I}}(B') \leqslant w_{\mathcal{I}'}(T) \leqslant 28\,\text{value}(\mathcal{I}') \leqslant 28\,\text{value}(\mathcal{I})$. However, $B'$ might not cross every non-singleton set in $\mathcal{L}$, or even a large weighted fraction of all sets.

We therefore slightly modify $B'$ to obtain our subtour $B$, as follows. For each maximal set $S \in \mathcal{L}$, suppose the first visit to $S$ in the subtour $B'$ arrives at a vertex $u^S \in S$ and departs from a vertex $v^S \in S$. Then we replace the segment of $B'$ from $u^S$ to $v^S$ by (see also the left part of Figure 4):

– a shortest path from $u^S$ to $u^S_{\max}$,
– a path from $u^S_{\max}$ to $v^S_{\max}$ inside $S$ as guaranteed by Lemma 3.1,
– and a shortest path from $v^S_{\max}$ to $v^S$.

Recall that, intuitively, a path from $u^S_{\max}$ to $v^S_{\max}$ crosses a large (weighted) fraction of the sets $R \in \mathcal{L} : R \subsetneq S$. We seek out these long paths, because taking them in every maximal set $S \in \mathcal{L}$ will allow the subtour $B$ to cross a large fraction of the LP value of the entire instance. On the other hand, it is a detour we can afford to make: it can be shown that the weight of each of the paths is at most $\text{value}(S)$, we take only one detour (consisting of three paths) per set $S$, and all these sets are disjoint by laminarity. Thus we have $w(B) \leqslant w(B') + 3\,\text{value}(\mathcal{I}) \leqslant 31\,\text{value}(\mathcal{I})$.

The formal statement concerning this part of our argument is summarized in the following claim (see the concept of quasi-backbone in the full version).

*Claim* 4.3. There is a polynomial-time algorithm that, given an irreducible instance $\mathcal{I}$, constructs a subtour $B$ such that $w(B) \leqslant 31\,\text{value}(\mathcal{I})$ and $2\sum_{S\in\mathcal{L}^*} y_S \leqslant 1/4\,\text{value}(\mathcal{I})$, where $\mathcal{L}^*$ consists of those sets in $\mathcal{L}$ that $B$ does *not* cross.

It is possible that the subtour $B$ is already a backbone: it might cross all non-singleton sets in $\mathcal{L}$. But even if it does not, the sets that it does not cross are now far and between: their total LP value is at most a $1/4$ fraction of the LP value of the instance. This allows us to use a recursive approach similar to the one used in the proof of Theorem 3.4:

1. Let $\mathcal{I}'$ be the instance yielded by contracting all maximal $S \in \mathcal{L}^*$. (In Figure 4, $R_1$ and $R_2$ are such maximal sets.) Then $B$ is a backbone for $\mathcal{I}'$ and $(\mathcal{I}', B)$ is a vertebrate pair. Invoke $\mathcal{A}$ on this pair to obtain a tour $T'$ of $\mathcal{I}'$ with $w(T') \leqslant \beta(\text{value}(\mathcal{I}') + w(B)) \leqslant \beta(\text{value}(\mathcal{I}) + 31\,\text{value}(\mathcal{I})) = 32\beta\,\text{value}(\mathcal{I})$.
2. Complete the lift of $T'$ to a tour by making one recursive call for each maximal set $S \in \mathcal{L}^*$.

See the right part of Figure 4 for an example of a tour of $\mathcal{I}$ created in this way. By Lemma 3.2, lifting a tour does not increase its weight and so the weight of the lift of $T'$ is at most $32\beta\,\text{value}(\mathcal{I})$. Hence, the statement will follow if we can show how to implement Step 2 in such a way that we complete the lift into a tour by incurring an additional weight of at most $32\beta\,\text{value}(\mathcal{I})$.

As in the previous section, we argue that this is possible under the following simplifying assumption: for each maximal $S \in \mathcal{L}^*$, the restriction of $x$ to the smaller instance $\mathcal{I}_S$ obtained by only considering the vertices in $S$ is a feasible solution to the Held-Karp relaxation of $\mathcal{I}_S$. Then $\mathcal{I}_S$ is a laminarly-weighted instance with $\text{value}(\mathcal{I}_S) = \text{value}(S)$. It is furthermore an irreducible instance, since $\mathcal{I}$ was irreducible. Hence we can recursively call our algorithm to find a tour $F_S$ of $\mathcal{I}_S$ with $w(F_S) \leqslant 64\beta\,\text{value}(S)$. As mentioned in Section 3, the simplifying assumption is not true in general. However, in the full version we describe a similar approach (where we introduce an operation of *inducing* on $S$) that loses another factor of two, and so $w(F_S) \leqslant 2 \cdot 64\beta\,\text{value}(S) = 128\beta\,\text{value}(S)$ in general. Thus the weight increase in the course of completing the lift into a tour is at most

$$128\beta \cdot \sum_{S \text{ maximal in } \mathcal{L}^*} \text{value}(S) \leqslant 128\beta \cdot 1/4\,\text{value}(\mathcal{I}) = 32\beta\,\text{value}(\mathcal{I})\,,$$

as required. The above inequality follows from the construction of $B$ (see Claim 4.3). Indeed, we crucially used the property $\sum_{S \in \mathcal{L}^*} \text{value}(S) \leqslant 1/4\,\text{value}(\mathcal{I})$ to bound the total value of the subinstances for which we make recursive calls, for which we incur an approximation factor of $2 \cdot 64\beta$. If these comprised, say, at least half of the total LP value, then the weight incurred by the recursive calls would be prohibitively large and the argument would fail. $\square$

# 5 Algorithm for Vertebrate Pairs

Now we are dealing with a vertebrate pair $(\mathcal{I}, B)$. Results in [Sve15] imply that it is enough to solve an easier problem called Local-Connectivity ATSP.

**Local-Connectivity ATSP.** The Local-Connectivity ATSP problem consists in finding "local" subtours that are only required to cross the sets of a given partition $V = V_1 \cup \ldots \cup V_k$ of vertices instead of connecting the entire graph (as in standard ATSP). A "good" solution to Local-Connectivity ATSP has a local requirement: each subtour should not be much more expensive than the lower bound on the cost (weight) of visiting the vertices in the subtour.

That lower bound on the cost of visiting vertices is defined in terms of a lower bound function $\text{lb} : V \to \mathbb{R}_+$. Intuitively, $\text{lb}(v)$ encodes how much we are willing to pay to visit vertex $v$. The lb function needs to be fixed by our algorithm before it is allowed to access the given partition.

More formally, the input to Local-Connectivity ATSP is an instance $\mathcal{I}$ together with a partition $V = V_1 \cup \ldots \cup V_k$ of vertices. (In the case of vertebrate pairs, we are also given a backbone $B$ to help us.) A solution $F \subseteq E$ must be Eulerian and cross every set $V_i$ in the partition. For some parameter $\alpha$, we say that a solution $F \subseteq E$ is *$\alpha$-light with respect to* lb if for every connected component $\tilde{G} = (V(\tilde{G}), E(\tilde{G}))$ of $F$ we have $w(E(\tilde{G})) \leqslant \alpha\,\text{lb}(V(\tilde{G}))$. We also say that an algorithm is $\alpha$-light if for any input partition it returns an $\alpha$-light solution.

**Theorem 5.1** ([Sve15]). *Suppose there is a polynomial-time algorithm for Local-Connectivity ATSP that is $\alpha$-light with respect to a lower bound function* lb *on $\mathcal{I}$. Then a tour of weight at most $10\alpha\,\mathrm{lb}(V)$ can be found in polynomial time.*

To simplify the notation, let $y_u = y_{\{u\}}$ if $\{u\} \in \mathcal{L}$ and let $y_u = 0$ otherwise. We define the lower bound function

$$\mathrm{lb}(v) = \begin{cases} (\mathrm{value}(\mathcal{I}) + w(B)) \,/\, |V(B)| & \text{if } v \in V(B), \\ 2y_v & \text{otherwise.} \end{cases}$$

Clearly $\mathrm{lb}(V) \leqslant 2\,\mathrm{value}(\mathcal{I}) + w(B) \leqslant 2(\mathrm{value}(\mathcal{I}) + w(B))$.[3] We exhibit an $O(1)$-light algorithm for Local-Connectivity ATSP with respect to lb. Theorem 4.2 via Theorem 5.1 then provides a constant-factor approximation algorithm for arbitrary irreducible instances, which in turn implies a constant-factor approximation algorithm for ATSP by Theorems 3.4 and 2.3.

We showcase our main ideas using the special case when the input is the singleton partition: $V = \{v_1\} \cup \{v_2\} \cup \ldots \cup \{v_n\}$. Then, the connectivity requirement is to find an Eulerian edge set $F$ which is adjacent to all vertices – in other words, a cycle cover. For more general partitions, we need to modify the construction by adding auxiliary vertices for each partition class. This can be achieved by extending the approach in [Sve15, Section 4].

For the singleton partition case, we first present the further special case when $\mathcal{L}$ also contains only singletons. This setting corresponds to a node-weighted instance. Then we extend the argument to a general family $\mathcal{L}$.

**Node-weighted instances.** Suppose that $\mathcal{L}$ contains only singletons. Then we have a node-weighted weight function: $w(u, v) = y_u + y_v$ for each $(u, v) \in E$. Further note that $B = \emptyset$ is a valid backbone. In the sequel we assume that $B = \emptyset$, and thus $\mathrm{lb}(u) = 2y_u$ for any $u \in V$. We now find a 1-light edge set $F$ for the singleton partition.

Our approach for this case is similar to the classical algorithm in [FGM82]. Let us solve the minimum-weight integer circulation problem in $G$ with the following constraints: minimize $w^\top z$ over $z \in \mathbb{R}^E_+$, subject to $z(\delta^-(v)) = z(\delta^+(v)) = 1$ for every $v \in V$ with $y_v > 0$, and $z(\delta^-(v)) = z(\delta^+(v)) \geqslant 1$ whenever $y_v = 0$. We observe that the Held-Karp solution $x$ provided in the instance $\mathcal{I}$ is a feasible solution. Using the integrality of the circulation polytope, there must be an integer solution $z \in \mathbb{Z}^E_+$ with $w^\top z \leqslant w^\top x = \mathrm{value}(\mathcal{I})$.

Now, the edge set $F$ defined by including $z(e)$ copies of every edge $e \in E$ satisfies the connectivity requirement. To prove 1-lightness, consider a connected component $\tilde{G}$ of $F$. We have

$$w(E(\tilde{G})) = \sum_{(u,v) \in E(\tilde{G})} y_u + y_v = 2 \sum_{v \in V(\tilde{G})} y_v = \mathrm{lb}(V(\tilde{G})). \tag{5.1}$$

The second equality holds because every $u \in V$ with $y_u > 0$ has exactly one incoming and one outgoing edge.

**General laminar families.** Let us now consider the case when $\mathcal{L}$ can be arbitrary, but the input for Local-Connectivity ATSP is still the singleton partition. We will find a 4-light edge set $F$ with respect to lb, in the form $F = B \cup F'$, where $B$ is the backbone (now non-empty) and $F'$ is another Eulerian edge set.

[3] In the full version we normalize the lb function so that $\mathrm{lb}(V) \leqslant \mathrm{value}(\mathcal{I})$. This is done to further emphasize the dependency between the lightness guarantee and the final approximation guarantee. Here we have preferred to keep the notation as simple as possible.

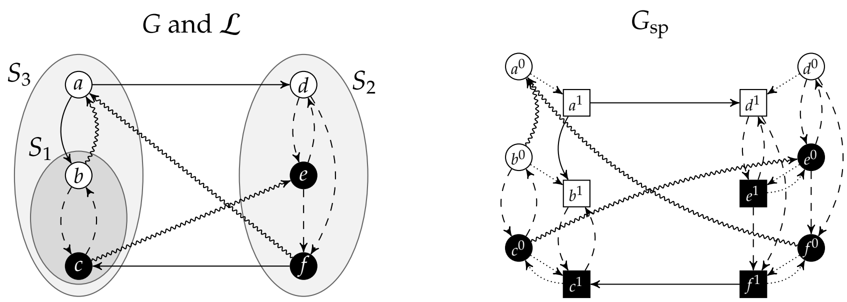


**Figure 5:** An example of the construction of $G_{\text{sp}}$. The vertices of the backbone are depicted in black. The nonsingleton sets in $\mathcal{L}$ are $S_1, S_2$, and $S_3$. Straight, swirly, and dashed edges correspond to various edge types in the construction.

We will obtain $F'$ by solving a minimum-weight circulation problem on a modification of the graph, called a *split graph* $G_{\text{sp}}$. The vertex set of $G_{\text{sp}}$ contains two copies of every original vertex: $V(G_{\text{sp}}) = \{v^0, v^1 : v \in V\}$. We can naturally map the split graph to $G$ by mapping $v^0$ and $v^1$ to $v$. The edge set is defined so that it satisfies the following property; the construction details are given in the full version. See Figure 5 for an illustration.

**Fact 5.2.** *Consider a cycle $C_{\text{sp}}$ in $G_{\text{sp}}$. If the image of $C_{\text{sp}}$ in $G$ (obtained by contracting every pair $v^0, v^1$ of vertices into a single vertex $v$) crosses a non-singleton tight set in $\mathcal{L}$, then it visits a vertex of the backbone.*

Note that the image of $C_{\text{sp}}$ in $G_{\text{sp}}$ will be a subtour. We define the weight function $w_{\text{sp}}$ in the split graph so that $w_{\text{sp}}(u^p, v^q) = w_{\text{sp}}(u, v)$ for $p, q \in \{0, 1\}$, whenever this edge is added. Further, we show that $x$ can also be mapped to an Eulerian vector $x_{\text{sp}} \in \mathbb{R}_+^{E(G_{\text{sp}})}$.

**Lemma 5.3.** *There is a polynomial-time algorithm that finds an Eulerian vector $x_{\text{sp}} \in \mathbb{R}_+^{E(G_{\text{sp}})}$ such that the image of $x_{\text{sp}}$ in $G$ is $x$, and $w_{\text{sp}}^\top x_{\text{sp}} = w^\top x$.*

The lemma can be proved by solving an auxiliary LP, which "channels" flow entering relevant sets in $\mathcal{L}$ to vertices in $V(B)$ inside these sets. This construction is inspired by [STV16].

Given the split graph $G_{\text{sp}}$ and $x_{\text{sp}}$, we can solve a similar minimum-weight circulation problem as in the node-weighted case. For every $v \in V$ either $v^0$ or $v^1$ will have at least $1/2$ units of in-flow in $x_{\text{sp}}$; we set a lower bound 1 on this vertex. Further, if $y_v > 0$, then we also set an upper bound of 2. We observe that $2x_{\text{sp}}$ is a feasible solution to this problem. We find an integer solution $z \in \mathbb{Z}_+^{E(G_{\text{sp}})}$ with $w_{\text{sp}}^\top z \leqslant 2 w_{\text{sp}}^\top x_{\text{sp}} = 2 w^\top x = 2\,\text{value}(\mathcal{I})$.

We obtain the edge set $F'$ by mapping $z$ from the split graph to the original graph $G$, and adding $z(e)$ copies of $e \in E$. Hence $w(F') \leqslant 2\,\text{value}(\mathcal{I})$. Also note that for every $y_v > 0$ we have $|\delta^-(v) \cap F'| \leqslant 4$.

It remains to show that $F = B \cup F'$ is a 4-light edge set with respect to lb. Consider any connected component $\tilde{G}$ of $F$. We distinguish two cases.

First, assume $\tilde{G}$ is the component containing the backbone $B$. We can upper-bound the weight of the component by the total weight of $F$: $w(E(\tilde{G})) \leqslant w(F) = w(B) + w(F') \leqslant w(B) + 2\,\text{value}(\mathcal{I})$. On the other hand, we have $\text{lb}(V(\tilde{G})) \geqslant \text{lb}(V(B)) = w(B) + \text{value}(\mathcal{I})$. This shows that $w(E(\tilde{G})) \leqslant 2\,\text{lb}(V(\tilde{G}))$.

Assume now that $\tilde{G}$ is any other component. Thus $E(\tilde{G}) \subseteq F'$ and $V(\tilde{G}) \cap V(B) = \emptyset$. Therefore $\text{lb}(V(\tilde{G})) = 2\sum_{v \in V(\tilde{G})} y_v$. We will now take advantage of Fact 5.2, the key property of the split graph. It implies that $\tilde{G}$ cannot contain any edge that crosses a non-singleton set in $\mathcal{L}$. Indeed, if there were any such edge, then $V(\tilde{G})$ would intersect $V(B)$. Consequently, $w(u, v) = y_u + y_v$ for every $(u, v) \in E$. Now we can use a similar estimation as in (5.1) to obtain $w(E(\tilde{G})) \leqslant 4\,\text{lb}(V(\tilde{G}))$; we use that $|\delta^-(v) \cap F'| \leqslant 4$ for every $y_v > 0$.

# References


[AG15] Nima Anari and Shayan Oveis Gharan. Effective-resistance-reducing flows, spectrally thin trees, and asymmetric TSP. In *IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 20–39, 2015. 1

[AGM+10] Arash Asadpour, Michel X. Goemans, Aleksander Madry, Shayan Oveis Gharan, and Amin Saberi. An O(log n/ log log n)-approximation algorithm for the asymmetric traveling salesman problem. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010*, pages 379–389, 2010. 1

[Blä08] Markus Bläser. A new approximation algorithm for the asymmetric TSP with triangle inequality. *ACM Transactions on Algorithms*, 4(4), 2008. 1

[CFN85] Gérard Cornuéjols, Jean Fonlupt, and Denis Naddef. The traveling salesman problem on a graph and some related integer polyhedra. *Mathematical Programming*, 33(1):1–27, 1985. 3

[CGK06] Moses Charikar, Michel X. Goemans, and Howard J. Karloff. On the integrality ratio for the asymmetric traveling salesman problem. *Math. Oper. Res.*, 31(2):245–252, 2006. 1, 2

[Chr76] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Graduate School of Industrial Administration, CMU, 1976. 1

[FGM82] Alan M. Frieze, Giulia Galbiati, and Francesco Maffioli. On the worst-case performance of some algorithms for the asymmetric traveling salesman problem. *Networks*, 12(1):23–39, 1982. 1, 9

[FS07] Uriel Feige and Mohit Singh. Improved approximation ratios for traveling salesperson tours and paths in directed graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 10th International Workshop, APPROX 2007, and 11th International Workshop, RANDOM 2007*, pages 104–118, 2007. 1

[GS11] Shayan Oveis Gharan and Amin Saberi. The asymmetric traveling salesman problem on graphs with bounded genus. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011*, pages 967–975, 2011. 1

[GSS11] Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. A randomized rounding approach to the traveling salesman problem. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011*, pages 550–559, 2011. 1

[KLS13] Marek Karpinski, Michael Lampis, and Richard Schmied. New inapproximability bounds for TSP. In *Algorithms and Computation - 24th International Symposium, ISAAC 2013*, pages 568–578, 2013. 2

[KLSS05] Haim Kaplan, Moshe Lewenstein, Nira Shafrir, and Maxim Sviridenko. Approximation algorithms for asymmetric TSP by decomposing directed regular multigraphs. *J. ACM*, 52(4):602–626, 2005. 1

[MS16] Tobias Mömke and Ola Svensson. Removing and adding edges for the traveling salesman problem. *J. ACM*, 63(1):2:1–2:28, 2016. 1

[Muc12] Marcin Mucha. 13/9-approximation for graphic TSP. In *29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012*, pages 30–41, 2012. 1

[STV16] Ola Svensson, Jakub Tarnawski, and László A. Végh. Constant factor approximation for ATSP with two edge weights. In *IPCO*, pages 226–237, 2016. 1, 2, 10

[SV14] 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. 1

[Sve15] Ola Svensson. Approximating ATSP by relaxing connectivity. In *FOCS 2015: Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, 2015. 1, 2, 4, 6, 7, 8, 9