2011 IEEE 52nd Annual Symposium on Foundations of Computer Science

Approximating Graphic TSP by Matchings

Tobias Mömke
KTH Royal Institute of Technology
Stockholm, Sweden
Email: moemke@kth.se

Ola Svensson
EPFL
Lausanne, Switzerland
Email: ola.svensson@epfl.ch

Abstract— We present a framework for approximating the metric TSP based on a novel use of matchings. Traditionally, matchings have been used to add edges in order to make a given graph Eulerian, whereas our approach also allows for the removal of certain edges leading to a decreased cost.

For the TSP on graphic metrics (graph-TSP), the approach yields a 1.461-approximation algorithm with respect to the Held-Karp lower bound. For graph-TSP restricted to a class of graphs that contains degree three bounded and claw-free graphs, we show that the integrality gap of the Held-Karp relaxation matches the conjectured ratio $4/3$. The framework allows for generalizations in a natural way and also leads to a 1.586-approximation algorithm for the traveling salesman path problem on graphic metrics where the start and end vertices are prespecified.

Keywords-approximation; TSP; linear programming

1. Introduction

The traveling salesman problem in metric graphs is one of most fundamental NP-hard optimization problems. In spite of a vast amount of research several important questions remain open. While the problem is known to be APX-hard and NP-hard to approximate with a ratio better than $220/219$ [23], the best upper bound is still the 1.5-approximation algorithm obtained by Christofides [4] more than three decades ago. A promising direction to improve this approximation guarantee has long been to understand the power of a linear program known as the Held-Karp relaxation [14] used by Dantzig, Fulkerson, and Johnson [6] already in 1954. On the one hand, the best lower bound on its integrality gap (for the symmetric case) is $4/3$ and indeed conjectured to be tight [10]. On the other hand, the best known analysis [25], [26] is based on Christofides' algorithm and gives an upper bound on the integrality gap of 1.5.

In the light of this difficulty of even determining the integrality gap of the Held-Karp relaxation, a reasonable way to approach the metric TSP is to restrict the set of feasible inputs. One promising candidate is the *graph-TSP,* that is, the traveling salesman problem where distances between cities are given by any graphic metric: the distance between two cities is the length of the shortest path in a given (unweighted) graph. Equivalently, graph-TSP can be formulated as the problem of finding an Eulerian multigraph within an unweighted input graph so as to minimize the number of edges. In contrast to TSP on Euclidean metrics that admits a PTAS [1], [17], the graph-TSP seems to capture the difficulty of the metric TSP in the sense that, as stated in [12], it is APX-hard and the lower bound 4/3 on the integrality gap of the Held-Karp relaxation is established using a graph-TSP instance.

The TSP on graphic metrics has recently drawn considerable attention. In 2005, Gamarnik et al. [9] showed that for cubic 3-edge-connected graphs, there is an approximation algorithm achieving an approximation ratio of $1.5 - 5/389$. This result was generalized to cubic graphs by Boyd et al. [3], who obtained an improved performance guarantee of 4/3. For subcubic graphs, i. e., graphs of degree at most 3, they also gave a 7/5-approximation algorithm with respect to the Held-Karp lower bound. In a major achievement, Oveis Gharan et al. [22] recently presented an approximation algorithm for graph-TSP with performance guarantee strictly better than 1.5. The approach in [22] is similar to that of Christofides in the sense that they start with a spanning tree and then add a perfect matching of those vertices of odd-degree to make the graph Eulerian. The main difference is that instead of starting with a minimum spanning tree, their approach uses the solution of the Held-Karp relaxation to sample a spanning tree. Although the proposed algorithm in [22] is surprisingly simple, the analysis is technically involved and several novel ideas are needed to obtain the improved performance guarantee $1.5 - \epsilon$ for an ϵ of the order 10^{-12}.

Our Results and Overview of Techniques: We propose an alternative framework for approximating the metric TSP and use it to obtain an improved approximation algorithm for graph-TSP.

Theorem 1.1. *There is a polynomial time approximation algorithm for graph-TSP with performance guarantee* $\frac{14\cdot(\sqrt{2}-1)}{12\cdot\sqrt{2}-13} < 1.461$.

The result implies an upper bound on the integrality gap of the Held-Karp relaxation for graph-TSP that matches the approximation ratio. For the restricted class of graphs, where each block (i. e., each maximally 2-vertex-connected subgraph) is either claw-free or of degree at most 3, we use the framework to construct a polynomial time $4/3$-approximation algorithm showing that the conjectured integrality gap of the Held-Karp relaxation is tight for those graphs. The techniques allow us to prove the tight result

0272-5428/11 $26.00 © 2011 IEEE
DOI 10.1109/FOCS.2011.56

that any 2-vertex-connected graph of degree at most 3 has a spanning Eulerian multigraph with at most $4n/3-2/3$ edges, which settles a conjecture of Boyd et al. [3] affirmatively.

Our framework is based on earlier works by Frederickson & Ja'ja' [8] and Monma et al. [19], who related the cost of an optimal tour to the size of a minimum 2-vertex-connected subgraph. More specifically, Monma et al. showed that a 2-vertex-connected graph $G = (V, E)$ always has a spanning Eulerian multigraph with at most $\frac{4}{3}|E|$ edges, generalizing a previous result of Frederickson & Ja'ja' who obtained the same result for the special case of planar 2-vertex-connected graphs. One interpretation of their approaches is the following. Given a 2-vertex-connected graph $G = (V, E)$, they show how to pick a random subset M of edges satisfying: (i) an edge is in M with probability $1/3$ and (ii) the multigraph H with vertex set V and edge set $E \cup M$ is spanning and Eulerian. From property (i) of M, the expected number of edges in H is $\frac{4}{3}|E|$ yielding their result.

Although the factor $4/3$ is asymptotically tight for some classes of graphs (one example is the family of integrality gap instances for the Held-Karp relaxation described in Section 2), the bound rapidly gets worse for 2-vertex-connected graphs with significantly more than n edges. The novel idea to overcome this issue is the following. Instead of adding all the edges in M to G, some of the edges in M might instead be removed from G to form H. As long as the removal of the edges does not disconnect the graph, this will again result in a spanning Eulerian multigraph H. To specify a subset R of edges that safely may be removed we introduce, in Section 3, the notion of a "removable pairing". The framework is then completed by Theorem 3.2, where we show that a 2-vertex-connected graph $G = (V, E)$ with a set R of removable edges has a spanning Eulerian multigraph with at most $\frac{4}{3}|E| - \frac{2}{3}|R|$ edges.

In order to use the framework, one of the main challenges is to find a sufficiently large set of removable edges. We first give a fairly easy method for finding such a set when considering subcubic graphs. To deal with general graphs, we then, in Section 4, generalize these ideas and show that the problem of finding a large set of removable edges can be reduced to that of finding a min-cost circulation in a certain circulation network. To analyze the circulation network we use (in Section 5) several properties of an extreme point solution to the Held-Karp relaxation to obtain our main algorithmic result.

Finally, we note that the techniques generalize in a natural way. Our results can be adapted to the more general traveling salesman path problem (graph-TSPP) with prespecified start and end vertices to improve on the approximation ratio of $5/3$ by Hoogeveen [15] when considering graphic metrics. More specifically, we obtain the following.

Theorem 1.2. *For any $\varepsilon > 0$, there is a polynomial time approximation algorithm for graph-TSPP with performance guarantee $3 - \sqrt{2} + \varepsilon < 1.586 + \varepsilon$.*

If furthermore each block of the given graph is degree three bounded, there is a polynomial time approximation algorithm for graph-TSPP with performance guarantee $1.5+\varepsilon$, for any $\varepsilon > 0$.

Due to space constraints, the proof of this theorem can be found in the full version [18].

2. PRELIMINARIES

Held-Karp Relaxation: The linear program known as the Held-Karp (or subtour elimination) relaxation is a well studied lower bound on the value of an optimal tour. It has a variable $x_{\{u,v\}}$ for each pair of vertices with the intuitive meaning that $x_{\{u,v\}}$ should take value 1 if the edge $\{u, v\}$ is used in the tour and 0 otherwise. Letting $G = (V, E)$ be the complete graph on the set of vertices and $c_{\{u,v\}}$ be the distance between vertices u and v, the Held-Karp relaxation can then be formulated as the linear program where we wish to minimize $\sum_{e \in E} c_e x_e$ subject to

$$\begin{aligned} x(\delta(v)) &= 2 \quad \text{for } v \in V, \\ x(\delta(S)) &\geq 2 \quad \text{for } \emptyset \neq S \subset V, \text{ and} \\ x &\geq 0, \end{aligned}$$

where $\delta(S)$ denotes the set of edges crossing the cut $(S, \bar{S})$ and $x(F) = \sum_{e \in F} x_e$ for any $F \subseteq E$.

Goemans & Bertsimas [11] proved that for metric distances the above linear program has the same optimal value as the linear program obtained by dropping the equality constraints. Moreover, when considering a graph-TSP instance $G = (V, E)$ we only need to consider the variables $(x_e)_{e \in E}$. Indeed, any solution x to the Held-Karp relaxation without equality constraints such that $x_{\{u,v\}} > 0$ for a pair of vertices $\{u, v\} \notin E$ can be transformed into a solution x' with no worse cost and $x'_{\{u,v\}} = 0$ by setting $x'_e = x_e + x_{\{u,v\}}$ for each edge on the shortest path between u and v, and $x'_e = x_e$ for the other edges. The Held-Karp relaxation for graph-TSP on a graph $G = (V, E)$ can thus be formulated as follows:

$$\begin{aligned} \min \quad & \sum_{e \in E} x_e \\ \text{s.t.} \quad & x(\delta(S)) \geq 2 \quad \text{for } \emptyset \neq S \subset V, \text{ and} \\ & x \geq 0. \end{aligned}$$

We shall refer to this linear program as $LP(G)$ and denote the value of an optimal solution by $OPT_{LP}(G)$. Its integrality gap was previously known to be at most $3/2 - \epsilon$ and at least $4/3$ for graphic instances. The lower bound is obtained by a claw-free graphic instance of degree at most 3 that consists of three paths of equal length with endpoints $(s_1, t_1), (s_2, t_2)$, and (s_3, t_3) that are connected so as $\{s_1, s_2, s_3\}$ and $\{t_1, t_2, t_3\}$ form two triangles (see Figure 1).

We end our discussion of $LP(G)$ with a useful observation. When considering graph-TSP, it is intuitively clear

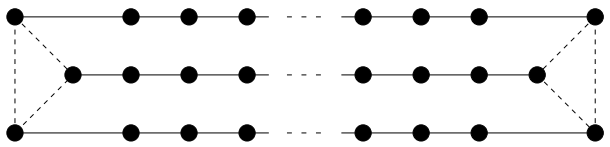

Figure 1. Graph for which the Held-Karp relaxation has an integrality gap tending to $4/3$. Any tour has value approaching $4n/3$ whereas the relaxation has a solution of value n obtained by setting the fractional value of the dashed and solid edges to $1/2$ and 1, respectively.

that we can restrict ourselves to 2-*vertex-connected* graphs, i. e., graphs that stay connected after deleting a single vertex. Indeed, if we consider a graph with a vertex v whose removal results in components $C_1, \ldots, C_\ell$ with $\ell > 1$ then we can recursively solve the graph-TSP problem on the ℓ subgraphs $G_1, G_2, \ldots, G_\ell$ induced by $C_1 \cup \{v\}, C_2 \cup \{v\}, \ldots, C_\ell \cup \{v\}$. The union of these solutions will then provide a solution to the original graph that preserves the approximation guarantee with respect to the linear programming relaxation since one can see that $OPT_{LP}(G) \geq \sum_{i=1}^{\ell} OPT_{LP}(G_i)$. We summarize this observation in the following lemma (see the appendix for a full proof).

Lemma 2.1. *Let G be a connected graph. If there is an r-approximation algorithm for graph-TSP on each 2-vertex-connected subgraph H of G (with respect to $OPT_{LP}(H)$) then there is an r-approximation algorithm for graph-TSP on G (with respect to $OPT_{LP}(G)$).*

Matchings of Cubic 2-Edge-Connected Graphs: Edmonds [7] showed that the following set of equalities and inequalities on the variables $(x_e)_{e \in E}$ determines the perfect matching polytope (i. e., all extreme points of the polytope are integral and correspond to perfect matchings) of a given graph $G = (V, E)$:

$$\begin{aligned} x(\delta(v)) &= 1 \quad \text{for } v \in V, \\ x(\delta(S)) &\geq 1 \quad \text{for } S \subseteq V \text{ with } |S| \text{ odd, and} \\ x &\geq 0. \end{aligned}$$

The linear description is useful for understanding the structure of the perfect matchings. For example, Naddef and Pulleyblank [21] proved that $x_e = 1/3$ defines a feasible solution when G is *cubic* and 2-*edge connected*, i. e., every vertex has degree 3 and the graph stays connected after the removal of an edge. They used that result to deduce that such graphs always have a perfect matching of weight at least $1/3$ of the total weight of the edges.

Standard algorithmic versions of Carathéodory's theorem (see e. g. Theorem 6.5.11 in [13]) say that, in polynomial time, we can decompose a feasible solution to the perfect matching polytope into a convex combination of polynomially many perfect matchings (see also [2] for a combinatorial approach for the matching polytope). Combining these results leads to the following lemma (see [3], [9], [19] for closely related variants that also have been useful for the graph-TSP problem).

Lemma 2.2. *Given a cubic 2-edge-connected graph G, we can in polynomial time find a distribution over polynomially many perfect matchings so that with probability $1/3$ an edge is in a perfect matching picked from this distribution.*

Note that all 2-vertex-connected graphs except the trivial graph on 2 vertices are 2-edge connected. We can therefore apply the above lemma to cubic 2-vertex-connected graphs.

3. Approximation Framework

Lemma 2.1 says that the technical difficulty in approximating the graph-TSP problem lies in approximating those instances that are 2-vertex connected. As alluded to in the introduction, we shall generalize previous results [8], [19] that relate the cost of an optimal tour to the size of a minimum 2-vertex-connected subgraph. The main difference is the use of matchings. Traditionally, matchings have been used to add edges to make a given graph Eulerian whereas our framework offers a structured way to specify a set of edges that safely may be removed leading to a lower cost. To identify the set of edges that may be removed we use the following definition.

Definition 3.1 (Removable pairing of edges)**.** Given a 2-vertex-connected graph G we call a tuple (R, P) consisting of a subset R of removable edges and a subset $P \subseteq R \times R$ of pairs of edges a *removable pairing* if

- an edge is in at most one pair;
- the edges in a pair are incident to a common vertex of degree at least 3;
- any graph obtained by deleting removable edges so that at most one edge in each pair is deleted stays connected.

The following theorem generalizes the corresponding result of [19] (their result follows from the special case of an empty removable pairing).

Theorem 3.2. *Given a 2-vertex-connected graph $G = (V, E)$ with a removable pairing (R, P), there is a polynomial time algorithm that returns a spanning Eulerian multigraph in G with at most $\frac{4}{3} \cdot |E| - \frac{2}{3} \cdot |R|$ edges.*

The proof of the theorem is presented after the following lemma on which it is based.

Lemma 3.3. *Given a 2-vertex-connected graph $G = (V, E)$ with a removable pairing (R, P), we can in polynomial time find a distribution over polynomially many subsets of edges such that a random subset M from this distribution satisfies:*

(a) *each edge of G is in M with probability $1/3$;*
(b) *at most one edge in each pair of P is in M; and*
(c) *each vertex has an even degree in the multigraph with edge set $E \cup M$.*

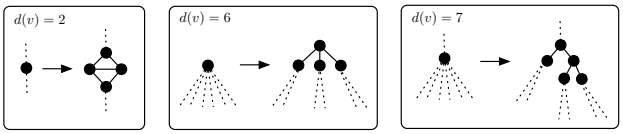

Figure 2. Examples of the used gadgets to obtain a cubic graph.

Proof: We shall use Lemma 2.2 and will therefore need a cubic 2-edge-connected graph. In the spirit of [8], we replace all vertices of G that are not of degree three by gadgets to obtain a cubic graph $G' = (V', E')$ as follows (see also Figure 2):

- A vertex v of degree 2 with neighbors u and w is replaced by a cycle consisting of four vertices v_N, v_W, v_S, v_E with the chord $\{v_W, v_E\}$. The gadget is then connected to the neighbours of v by the edges $\{u, v_N\}$ and $\{v_S, w\}$.
- A vertex v with $d(v) > 3$ is replaced by a tree T_v that has $\lfloor d(v)/2 \rfloor$ leaves, a binary root if $d(v)$ is odd, and otherwise only degree 3 internal vertices. Each leaf is connected to two neighbours of v such that the edges incident to v that form a pair in P are incident to the same leaf. If $d(v)$ is odd, one of the neighbors is left and connected to the binary root.

The above gadgets guarantee that the graph G' is cubic and it is 2-vertex connected since G was assumed to be 2-vertex connected. We can therefore apply Lemma 2.2 in order to obtain a random perfect matching M'. Each edge of G' is in M' with a probability of exactly 1/3. Let M be the set of edges obtained by restricting M' to the edges of G in the obvious way. Now M contains each edge of G with probability 1/3. Note that in general M is not a matching.

We complete the proof by showing that M also satisfies properties (b) and (c). As each pair of edges in P is incident to a vertex of degree at least 3, we have, by the construction of the gadgets, that they are incident to a common vertex in G' and hence at most one edge of each pair is in M. Finally, property (c) follows from that $E' \cup M'$ is clearly a spanning Eulerian multigraph of G' and compressing a set of even-degree vertices results in one vertex of even degree. ∎

Equipped with the above lemma we are now ready to prove the main result of this section.

Proof of Theorem 3.2: Pick a random subset $M \subseteq E$ of edges that satisfies the properties of Lemma 3.3. Let M_R be the set of those edges of M that are removable and let $\bar{M}_R$ be the set of the remaining edges of M.

Consider the multigraph H on vertex set V and edge set $(E \setminus M_R) \cup \bar{M}_R$. Observe that both adding an edge and removing an edge swaps the parity of the degree of an incident vertex. We have thus from property (c) of Lemma 3.3 that the degree of each vertex in H is even. Moreover, as (R, P) is a removable pairing, property (b) of Lemma 3.3 gives that H is connected. Alltogether we have that H is an Eulerian graph, i. e., a graph-TSP solution. We continue to calculate its expected number of edges, which is

$$\mathbb{E}[|E| + |\bar{M}_R| - |M_R|]. \tag{1}$$

Using that each edge is in M with probability $1/3$, we have, by linearity of expectation, that (1) equals

$$|E| + \frac{1}{3}(|E| - |R|) - \frac{1}{3}|R| = \frac{4}{3} \cdot |E| - \frac{2}{3} \cdot |R|.$$

To conclude the proof, we note that the selection of M can be derandomized since there are, by Lemma 3.3, polynomially many edge subsets to choose from; taking the one that minimizes the number of edges of H is sufficient. ∎

With Theorem 3.2, we have already the core ingredient to show our results on bounded degree graphs for which we obtain a tight bound on the integrality gap of the Held-Karp relaxation. We note that the result can also be seen as a corollary (see Section 5.1) of the more involved techniques introduced in Section 4 for the general case. However, the easier and more direct proof that we sketch here gives valuable insights in the concept of removable edges and also motivates the approach developed in Section 4 for general graphs.

Lemma 3.4. *Given a 2-vertex-connected graph G with n vertices of degree at most 3, there is a polynomial time algorithm that computes a spanning Eulerian multigraph H in G with at most $4n/3 - 2/3$ edges.*

Proof Sketch: Theorem 3.2 says that in order to find a short tour it is sufficient to find a large enough removable pairing (R, P). For subcubic graphs, this can be done as follows (see Figure 3(a) for an example):

(a) Obtain a spanning tree T of G by depth-first search (starting from some arbitrary root r). We call the edges in T *tree-edges* and the others *back-edges*. Note that each back-edge connects a vertex to either one of its predecessors or one of its successors in T.
(b) Define the set P by pairing each back-edge $e = \{u, v\}$ with the adjacent tree-edge $e' = \{v, w\}$, where u is a successor of v in T and e' is on the path to u in T (unless v is of degree 2 or the tree-edge is already paired, which can only happen if $v = r$).
(c) Form the set R of removable edges by taking all back-edges and the paired tree-edges.

Using that the graph G is subcubic and that T is a depth-first search tree, one can verify that (R, P) indeed is a removable pairing. Moreover, as all back-edges ($|E| - (n-1)$ many) are removable and each one of them except one (incident to the root) is paired with a tree-edge, we have $|R| = 2(|E| - (n - 1)) - 1$. The claimed result now follows from Theorem 3.2. ∎

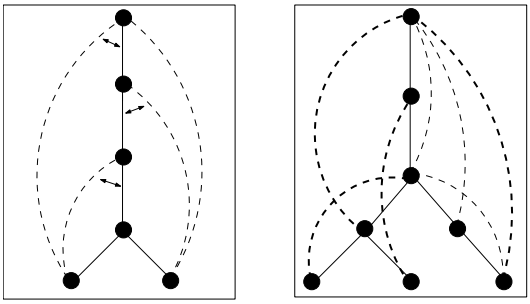

Figure 3. (a): Example of the pairing obtained for subcubic 2-vertex-connected graphs. The pairs are depicted with arrows and the tree-edges and back-edges are solid and dashed, respectively. (b): For graphs of arbitrary degree it is more complex to find a large enough removable pairing since not each back-edge can be paired with a (unique) tree-edge.

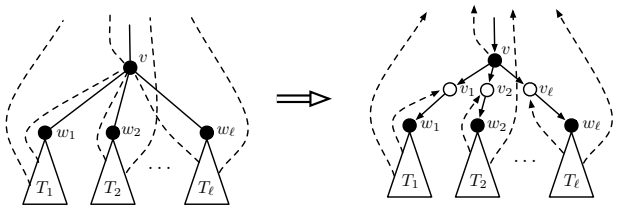

Figure 4. The gadget that, for each child of v, introduces a new vertex (depicted in white) and redirects back-arcs.

Note that the proof of Lemma 3.4 relied on finding a large enough set of removable edges with respect to the total number of edges in order to apply Theorem 3.2. This could be achieved because for subcubic graphs basically each back-edge can be paired with a unique tree-edge. For general graphs, this is not the case as can be seen for example by looking at the root of the graph depicted in Figure 3(b). However, further inspection of that graph reveals that the subgraph consisting of the tree-edges and the fat back-edges is still 2-vertex connected and allows for a similar pairing as in the subcubic case. This motivates the following approach for general graphs: find a depth-first search tree and then find a subset of back-edges so that the resulting graph is 2-vertex connected and the number of back-edges not paired with a tree-edge is minimized. In the next section we show how to find such a subset of back-edges by computing a minimum cost circulation. We then, in Section 5.2, show how the Held-Karp relaxation can both guide us in the selection of the depth-first search tree and help us in analyzing the cost of the circulation flow, which in turn leads to the improved approximation guarantee for general graphs.

4. Finding a Removable Pairing by Minimum Cost Circulation

In order to use our framework, one of the main challenges is to find a removable pairing that is sufficiently large. In the following, we show how to obtain a useful removable pairing based on circulations.

Consider a 2-vertex connected graph G and—as in the proof sketch of Lemma 3.4—let T be a spanning tree of G obtained by depth-first search. Thus T is composed of tree-edges and the remaining edges in G but not in T are back-edges.

We shall now define a circulation network $C(G,T)$. We start by introducing an orientation of G: all tree-edges become tree-arcs directed from the root to the leaves and all back-edges become back-arcs directed towards the root. To distinguish the circulation network and the original graphs, we use the names $\overrightarrow{G}$ and $\overrightarrow{T}$ for the network versions of G and T. In order to ensure connectivity properties of subnetworks obtained from feasible circulations, we replace some of the vertices by gadgets as follows.

For each vertex v except the root that has ℓ children $w_1, w_2, \ldots, w_\ell$ in the tree, we introduce ℓ new vertices $v_1, v_2, \ldots, v_\ell$ and replace the tree-arc (v, w_j) by the tree-arcs (v, v_j) and (v_j, w_j) for $j = 1, 2, \ldots, \ell$. Then we redirect all incoming back-arcs of v from the subtree rooted by w_j to v_j. For an illustration of the gadget see Figure 4. This way, all back-arcs start in old vertices and lead to new vertices or the root. In the following, we call the new vertices and the root *in-vertices* and the remaining (old) ones *out-vertices*. We also let $\mathcal{I}$ be the set of all in-vertices.

We now specify a lower bound (demand) and an upper bound (capacity) on the circulation. For each arc a in $\overrightarrow{T}$, we set the demand of a to 1 and for all other arcs to 0. The capacity is ∞ for any arc. Finally, the cost of a circulation f in $C(G,T)$ is the piecewise linear function $\sum_{v\in\mathcal{I}} \max[f(B(v)) - 1, 0]$, where $B(v)$ is the set of incoming back-arcs of v. One can think of the cost as the total circulation on the back-arcs except that each in-vertex accepts a circulation of 1 for free; the intuition being that we would like to minimize the number of back-arcs that cannot be paired with tree-arcs and only one back-arc in $B(v)$ can be paired with the tree-arc going out from v. Note that algorithmically there is no considerable difference whether we use our cost function or define a linear cost function on the arcs: for each in-vertex v we introduce a new vertex v' and redirect all back-arcs of v to v' while setting their costs to 0. Then we introduce two arcs (v', v), one of cost 0 and capacity 1 and the other of cost 1 and capacity ∞.

The following lemma shows how to use a circulation in $C(G,T)$ to approximate graph-TSP.

Lemma 4.1. *Given a 2-vertex connected graph G and a depth first search tree T of G let C^* be the minimum cost circulation to $C(G,T)$ of cost $c(C^*)$. Then there is a spanning Eulerian multigraph H in G with at most $\frac{4}{3}n + \frac{2}{3}c(C^*) - 2/3$ edges.*

Proof: We first note that, for any arc of $C(G,T)$, the demand and the capacity are integral. Therefore, applying Hoffman's circulation theorem (see [24], Corollary 12.2a), we can assume the circulation C^* to be integral. Let $C^*(G,T)$ be the support of C^* in $C(G,T)$, i. e., the induced subgraph of the arcs with non-zero circulation in C^*, and let H be the subgraph of G obtained from $C^*(G,T)$ by compressing the gadgets of the circulation network in the obvious way.

To prove the lemma, we shall first prove that graph H is 2-vertex connected and then define a removable pairing (R,P) on H in order to apply Theorem 3.2. That H is 2-vertex connected follows from flow conservation, that each arc a in $\overrightarrow{T}$ has demand 1, and the design of the gadgets. Indeed, if H had a cut vertex v with children $w_1, w_2, \ldots, w_\ell$ in T then one of the subtrees, say the one rooted at w_j, would have no back-edges to the ancestors of v which in turn, by flow conservation, would contradict that the tree-arc (v, v_j) in $\overrightarrow{T}$ carries a flow of at least 1. (Recall that the edge $\{v, w_j\}$ in T is replaced by tree-arcs (v, v_j) and (v_j, w_j) in $\overrightarrow{T}$.)

We now determine a removable pairing (R,P) on H. For ease of argumentation we shall first slightly abuse notation and define a removable pairing (R_C, P_C) on $C^*(G,T)$. The set P_C consists of all (e,e') such that $e=(u,v)$ is a back-arc of cost 0 in $C^*(G,T)$, v has at least two incoming arcs, and $e'=(v,w)$ is a tree-arc. Note that each such v is an in-vertex, the number of incoming back-arcs of cost zero is at most one, e' is the unique outgoing tree-arc of v, and the only possible vertex v with only one incoming back-arc and no other incoming arc is the root. The set R_C contains all arcs from P_C and additionally all remaining back-arcs of $C^*(G,T)$. In other words, each arc of $C^*(G,T)$ that is neither in $\overrightarrow{T}$ nor in P_C is a back-arc with integer non-zero cost in the circulation or a back-arc to the root. Hence, $|R_C| - 2|P_C| = c(C^*)$ if the root has more than one incoming back-arc and $|R_C| - 2|P_C| = c(C^*)+1$ otherwise. Note that the minimality of C^* implies that its back-arcs have flows of at most 1 each.

The removable pairing (R,P) on H is now obtained from (R_C, P_C), by merely compressing the gadgets used to form $C(G,T)$ and by dropping the orientations of the arcs. As all edges in R_C are either back-arcs or they are tree-arcs starting from an in-vertex, no arc in R_C is removed by the compression and thus $|R| = |R_C|$ and $|P| = |P_C|$. Moreover, H has $(n-1) + |R| - |P|$ edges and, assuming (R,P) is a valid removable pairing, Theorem 3.2 yields that H (and thus G) has a spanning Eulerian multigraph with at most $\frac{4}{3}((n-1)+|R|-|P|)-\frac{2}{3}|R| = \frac{4}{3}n+\frac{2}{3}(|R|-2|P|)-\frac{4}{3} \leq \frac{4}{3}n + \frac{2}{3}c(C^*) - \frac{2}{3}$ edges. The last inequality follows from that $|R| - 2|P|$ is at most $c(C^*)+1$.

Therefore, we can conclude the proof by showing that (R,P) is a valid removable pairing. It is easy to verify that (R,P) satisfies the first two conditions of Definition 3.1, that is, each edge is contained in at most one pair and the edges in each pair are incident to one common vertex of degree at least three. The third condition follows from that, for any vertex v of H, the vertices in the subtree T_v of T rooted at v form a connected subgraph of H even after removing edges according to (R,P). To see this we do a simple induction on the depth of v. In the base case, v is a leaf and the statement is clearly true. For the inductive step, consider a vertex v with ℓ children $w_1, w_2, \ldots, w_\ell$ in T. By the inductive hypothesis, the vertices in T_{w_j} for $j = 1, 2, \ldots, \ell$ stay connected after the removal of edges according to (R,P). To complete the inductive step it is thus sufficient to verify that v is connected to each T_{w_j} after the removal of edges. If $\{v, w_j\}$ is not in R this clearly holds. Otherwise if $e_j = \{v, w_j\} \in R$ then by the definition of (R,P) there is an edge e such that $(e, e_j) \in P$ and e is incident to v and a vertex in T_{w_j}. Since at most one edge in each pair is removed we have that v also stays connected to T_{w_j} in this case, which completes the inductive step. We have thus proved that (R,P) satisfies the properties of a removable pairing which completes the proof of the statement.

■

5. Improved Approximation Algorithms

We first show how to apply our framework by formally proving Lemma 3.4. We then show how to use our framework to obtain an improved approximation algorithm for general graphs.

5.1. Bounded Degree and Claw-Free Graphs

Lemma 3.4 (Restated) *Given a 2-vertex-connected graph G with n vertices of degree at most 3, there is a polynomial time algorithm that computes a spanning Eulerian multigraph H in G with at most $4n/3 - 2/3$ edges.*

Proof: If G has one or two vertices, we obtain an Eulerian multigraph of zero or two edges. Otherwise, we compute a depth-first search tree T in G and determine the circulation network $C(G,T)$. We now show that this network has a feasible circulation f of cost at most one. Let us assign a circulation of one to each back-arc e in $C(G,T)$ and push it through the path in $\overrightarrow{T}$ that is incident to both the start and end vertex of e. By the construction of $C(G,T)$ and from the assumption that G is 2-vertex connected, each tree-arc is in a directed cycle that contains exactly one back-arc. Therefore, all demand constraints are satisfied. Due to the degree-bounds, no vertex but the root may have more than one incoming back-arc. The cost $\sum_{v \in \mathcal{I}} \max[f(B(v)) - 1, 0]$ of the circulation is therefore at most one and zero if the root has only one back-arc. If the circulation cost is zero, by Lemma 4.1 we obtain a spanning Eulerian multigraph H in G with at most $4n/3 - 2/3$ edges. For those circulations

where the cost is one, the proof of Lemma 4.1 allows to save an additional constant of $2/3$ (since then the root has more than one incoming back-arc) and we obtain the same bound on the number of edges.

■

Note that it is sufficient to find a 2-vertex-connected degree three bounded spanning subgraph (a 3-trestle) and thus, using a result from [16], we can apply Lemma 3.4 also to claw-free graphs. Applying Lemma 2.1, we obtain an upper bound of 4/3 on the integrality gap for the Held-Karp relaxation for the considered class of graphs. In addition, along the lines of the proof of Lemma 2.1, one can see that the above arguments imply that any connected graph G decomposed into k blocks, i. e., maximal 2-connected subgraphs, such that each block is either degree three bounded or claw-free, has a spanning Eulerian multigraph with at most $4n/3 + 2k/3 - 4/3$ edges.

5.2. General Graphs

We now apply our framework to graphs without degree constraints. We start with an algorithm that achieves an approximation ratio better than $3/2$ for graphs for which the linear programming relaxation has a value close to n. Let $G = (V, E)$ be an n-vertex graph. The support $E' = \{e : x^*_e > 0\}$ of an extreme point x^* of $LP(G)$ is known to contain at most $2n - 1$ edges (see Theorem 4.9 in [5]). Moreover, if we let x^* be an optimal solution, then any r-approximate solution to graph $G' = (V, E')$ with respect to $OPT_{LP}(G')$ is an r-approximate solution to G with respect to $OPT_{LP}(G)$, because $E' \subseteq E$ and $OPT_{LP}(G') = OPT_{LP}(G)$. We can thus restrict the analysis to n-vertex graphs with at most $2n - 1$ edges and, by Lemma 2.1, we can further assume the graph to be 2-vertex connected.

Algorithm 1.

Input: A 2-vertex-connected graph G with n vertices and at most $2n - 1$ edges.

1: Obtain an optimal solution x^* to $LP(G)$.
2: Obtain a depth-first-search tree T of G by starting at some root and in each iteration pick, among the possible edges, the edge e with maximum x^*_e.
3: Solve the min cost circulation problem $C(G, T)$ to obtain a circulation C^* with cost $c(C^*)$.
4: Apply Lemma 4.1 to find a spanning Eulerian multigraph with less than $\frac{4}{3}n + \frac{2}{3}c(C^*)$ edges.

To analyze the approximation ratio achieved by Algorithm 1, we bound the cost of the circulation.

Lemma 5.1. *We have*

$$c(C^*) \leq 6(1 - \sqrt{2})n + (4\sqrt{2} - 3)OPT_{LP}(G).$$

Proof: For notational convenience, when considering an arc a in the flow network, we shall slightly abuse notation and use x^*_a to denote the value of the corresponding edge in G according to the optimal LP-solution x^*. We prove the statement by defining a fractional circulation f of cost at most $6(1 - \sqrt{2})n + (4\sqrt{2} - 3)OPT_{LP}(G)$. The circulation f will in turn be the sum of two circulations f' and f''. We obtain the circulation f' as follows: for each back-arc a we push a flow of size $\min[x^*_a, 1]$ along the cycle formed by a and the tree-arcs in $\overrightarrow{T}$. We shall now define the circulation f'' so as to guarantee that f forms a feasible circulation, i. e., one that satisfies the demands $f_a \geq 1$ for each $a \in \overrightarrow{T}$. As out- and in-vertices are alternating in $\overrightarrow{T}$ and in-vertices have only one child in $\overrightarrow{T}$ and no outgoing back-edges, a sufficient condition for f to be feasible can be seen to be $f_a \geq 1$ for each $a \in \overrightarrow{T}$ that is from an out-vertex to an in-vertex. To ensure this, we now define f'' as follows. For each vertex v of G that is replaced by a gadget consisting of an out-vertex v and a set $\mathcal{I}_v$ of in-vertices, we push for each $w \in \mathcal{I}_v$ a flow of size $\max[1 - f'_{(v,w)}, 0]$ along a cycle that includes the arc (v, w) (and one back-arc). Note that such a cycle is guaranteed to exist since G was assumed to be 2-vertex connected. From the definition of f'', we have thus that $f = f' + f''$ defines a feasible circulation.

We proceed by analyzing the cost of f, i. e., $\sum_{v \in \mathcal{I}} \max[f(B(v)) - 1, 0]$, where $\mathcal{I}$ is the set of all in-vertices and $B(v)$ is the set of incoming back-arcs of $v \in \mathcal{I}$. Note that the cost is upper bounded by $\sum_{v \in \mathcal{I}} \max[f'(B(v)) - 1, 0] + \sum_{v \in \mathcal{I}} f''(B(v))$ and we can thus analyze these two terms separately. We start by bounding the second summation and then continue with the first one. If $OPT_{LP}(G) = n$ then one can see that $f'' = 0$. Moreover,

Claim 5.2. *We have*

$$\sum_{v \in \mathcal{I}} f''(B(v)) \leq OPT_{LP}(G) - n.$$

Proof of Claim: When considering a vertex v as done above in the definition of f'', the flow pushed on back-arcs is $\sum_{w \in \mathcal{I}_v} \max[1 - f'_{(v,w)}, 0]$ which equals $\sum_{w \in \mathcal{I}'_v} (1 - f'_{(v,w)})$, where $\mathcal{I}'_v = \{w \in \mathcal{I}_v : f'_{(v,w)} < 1\}$. Letting T_w be the set of vertices of G in the subtree of the undirected tree T rooted at the child of $w \in \mathcal{I}'_v$, we have, by the definition of f',

$$f'_{(v,w)} = \sum_{a \in \delta(T_w) \setminus \delta(v)} \min[x^*_a, 1] = x^*(\delta(T_w) \setminus \delta(v)).$$

The second equality follows from that if $x^*_a > 1$ for some $a \in \delta(T_w) \setminus \delta(v)$ then $f'_{(v,w)} \geq 1$ and hence $w \notin \mathcal{I}'_v$. We have thus $\sum_{w \in \mathcal{I}'_v} (1 - f'_{(v,w)}) = |\mathcal{I}'_v| - \sum_{w \in \mathcal{I}'_v} x^*(\delta(T_w) \setminus \delta(v))$. As we are considering a depth-first-search tree (see Figure 5),

$$2 \sum_{w \in \mathcal{I}'_v} x^*(\delta(T_w) \setminus \delta(v)) =$$

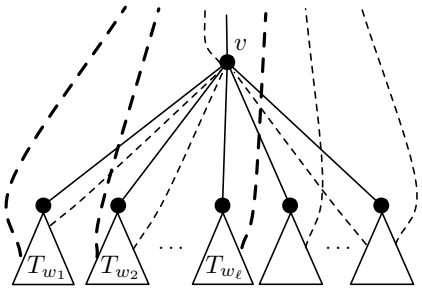

Figure 5. An illustration of Equality (2) with $\mathcal{I}'_v = \{w_1, w_2, \ldots, w_\ell\}$: both the left-hand-side and the right-hand-side of the equality express two times the value of the fat edges.

$$\sum_{w\in\mathcal{I}'_v} x^*(\delta(T_w)) + x^*\left(\delta\left(\bigcup_{w\in\mathcal{I}'_v} T_w \cup \{v\}\right)\right) - x^*(\delta(v)). \tag{2}$$

Since by the feasibility of x^* each of the sets corresponds to a cut of fractional value at least 2, we use $2\cdot(|\mathcal{I}'_v|+1) - x^*(\delta(v))$ as a lower bound on (2).
Summarizing the above calculations yields

$$\begin{aligned}\sum_{w\in\mathcal{I}'_v}\left(1-f'_{(v,w)}\right) &= |\mathcal{I}'_v| - \sum_{w\in\mathcal{I}'_v} x^*(\delta(T_w)\setminus\delta(v)) \\ &\le \frac{x^*(\delta(v))}{2} - 1.\end{aligned}$$

Repeating this argument for each v we have

$$\begin{aligned}\sum_{v\in\mathcal{I}} f''(B(v)) &= \sum_{v\in V}\sum_{w\in\mathcal{I}'_v}\left(1-f'_{(v,w)}\right) \\ &\le \sum_{v\in V}\left(\frac{x^*(\delta(v))}{2}-1\right),\end{aligned}$$

which equals $OPT_{LP}(G) - n$ since $OPT_{LP}(G) = \frac{1}{2}\sum_{v\in V} x^*(\delta(v))$. ∎

We proceed by bounding $\sum_{v\in\mathcal{I}}\max[f'(B(v))-1,0]$ from above.

Claim 5.3. *We have*

$$\sum_{v\in\mathcal{I}}\max[f'(B(v))-1,0] \le \big(7-6\sqrt{2}\big)n + 4\big(\sqrt{2}-1\big)OPT_{LP}(G).$$

Proof of Claim: To analyze this expression we shall use two facts. First G has at most $2n-1$ edges, and therefore the number of back-arcs is at most $2n-1-(n-1)=n$. Second, as the depth-first-search in each iteration chooses (among the available edges) the edge e with maximum x^*_e, we have that $x^*_a \le x^*_{t_v}$ for each $a \in B(v)$ where t_v is the outgoing tree-arc of $v \in \mathcal{I}$. Moreover, as $f'_a = \min[x^*_a, 1]$ for each back-arc, the number of back-arcs in $B(v)$ is at least $\left\lceil \frac{f'(B(v))}{\min[x^*(t_v),1]} \right\rceil$. Combining these two facts gives us that

$$\sum_{v\in\mathcal{I}}\left\lceil\frac{f'(B(v))}{\min[x^*(t_v),1]}\right\rceil \le n. \tag{3}$$

For $v \in \mathcal{I}$, we partition $f'(B(v))$ into $\ell_v = \min[2 - x^*(t_v), f'(B(v))]$ and $u_v = f'(B(v)) - \ell_v$. Furthermore, let $u^* = \sum_{v\in\mathcal{I}} u_v$. With this notation we can upper bound $\sum_{v\in\mathcal{I}}\max[f'(B(v))-1,0]$ by

$$\sum_{v\in\mathcal{I}}\max[\ell_v - 1, 0] + u^* \tag{4}$$

and relax Inequality (3) to

$$\sum_{v\in\mathcal{I}}\frac{\ell_v}{x^*(t_v)} \le n - u^*. \tag{5}$$

The cost (4) (where we ignore u^*) subject to (5) can now be interpreted as a knapsack problem of capacity $n-u^*$ that is packed with an item of profit $\max[\ell_v - 1, 0]$ and size $\ell_v/x^*(t_v)$ for each $v \in \mathcal{I}$. Consequently, we can upper bound (4) by considering the fractional knapsack problem with capacity $n-u^*$ and infinitely many items of a maximized profit to size ratio. Associating a variable L with ℓ_v and T with $x^*(t_v)$ this ratio is $\max_{0\le T\le 1, 0\le L\le 2-T}\frac{L-1}{L}\cdot T$. For any T the ratio is maximized by letting $L = 2-T$ and we can thus restrict our attention to items with profit to size ratio $\max_{0\le T\le 1}\frac{1-T}{2-T}\cdot T$. A simple analysis shows that the maximum is achieved when $T = 2-\sqrt{2}$. Therefore, taking into account u^*, the profit (4) is upper bounded by

$$\frac{\sqrt{2}-1}{\sqrt{2}}\cdot(2-\sqrt{2})\cdot(n-u^*)+u^* = (\sqrt{2}-1)^2\cdot(n-u^*)+u^*.$$

As the fractional degree of a vertex v that is replaced by a gadget with a set $\mathcal{I}_v$ of in-vertices is at least $2+\sum_{w\in\mathcal{I}_v} u_w$, we have $u^* \le 2(OPT_{LP}(G)-n)$. Hence,

$$\begin{aligned}(4) \le\ & (\sqrt{2}-1)^2\cdot(n-2(OPT_{LP}(G)-n)) \\ & +2(OPT_{LP}(G)-n),\end{aligned}$$

which equals $(7-6\sqrt{2})n + 4(\sqrt{2}-1)OPT_{LP}(G)$. ∎

Finally, by summing up the bounds given by Claim 5.2 and Claim 5.3 we bound the cost of f and hence $c(C^*)$ from above by $OPT_{LP}(G) - n + n(7-6\sqrt{2}) + 4(\sqrt{2}-1)OPT_{LP}(G)$, which equals $6(1-\sqrt{2})n + (4\sqrt{2}-3)OPT_{LP}(G)$. ∎

Having analyzed Algorithm 1, we are ready to prove our main algorithmic result.

Theorem 1.1 (Restated) *There is a polynomial time approximation algorithm for graph-TSP with performance guarantee* $\frac{14\cdot(\sqrt{2}-1)}{12\cdot\sqrt{2}-13} < 1.461$.

Proof: By Lemma 2.1 and the discussion before Algorithm 1, we can restrict the analysis to n-vertex graphs that

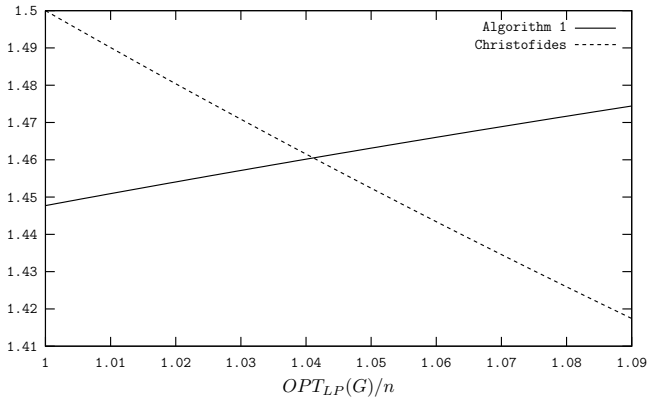

Figure 6. The approximation ratios of Algorithm 1 and Christofides' algorithm depending on the ratio $OPT_{LP}(G)/n$.

are 2-vertex connected and have at most $2n-1$ edges. The statement now follows by using Algorithm 1 if $OPT_{LP}(G)$ is close to n and otherwise by using Christofides' algorithm.

On the one hand, since Christofides' algorithm returns a solution with at most $n-1+OPT_{LP}(G)/2$ edges (see [25] for an analysis of Christofides' algorithm in terms of $OPT_{LP}(G)$), it has an approximation guarantee of at most

$$\frac{n+OPT_{LP}(G)/2}{OPT_{LP}(G)}.$$

On the other hand, by Lemma 5.1, the approximation guarantee of Algorithm 1 is at most

$$\frac{\frac{4}{3}n+\frac{2}{3}\left(6(1-\sqrt{2})n+(4\sqrt{2}-3)OPT_{LP}(G)\right)}{OPT_{LP}(G)}.$$

In particular, the approximation guarantee of Algorithm 1 for a graph G with $OPT_{LP}(G)=n$ is

$$4/3+2/3\cdot(\sqrt{2}-1)^2\approx 1.4477$$

but deteriorates as $OPT_{LP}(G)$ increases. The approximation guarantee of Christofides' algorithm on the other hand is getting better and better as $OPT_{LP}(G)$ increases. Comparing these two ratios, one gets that the worst case happens when $OPT_{LP}(G)=\frac{24\sqrt{2}-26}{16\sqrt{2}-15}n$ (see Figure 6) and, by using simple arithmetics, the approximation guarantee can be seen to be $\frac{14(\sqrt{2}-1)}{12\cdot\sqrt{2}-13}$. ∎

6. Conclusions

We have introduced a framework of removable pairings to find Eulerian multigraphs. This framework proved to be useful to obtain an approximation algorithm for graph-TSP with an approximation ratio smaller than 1.461 and to obtain a tight upper bound on the integrality gap of the Held-Karp relaxation for a restricted class of graphs that contains degree three bounded and claw-free graphs. In particular, we showed that in subcubic 2-vertex-connected graphs we can always find a solution to graph-TSP of at most $4n/3-2/3$ edges, which settles a conjecture from [3] affirmatively.

Our framework is not restricted to graph-TSP. With the same techniques and a more detailed analysis, our result translates to the traveling salesman path problem on graphic metrics with prespecified start and end vertex. In this way, one is guaranteed to obtain an approximation ratio smaller than 1.586 and, for the degree three bounded case, the approximation ratio gets arbitrarily close to 1.5.

An interesting open problem is to improve the analysis of the circulation network, i.e., Lemma 5.1. Recent progress has been made on this by Mucha [20] who used a more involved "knapsack" argument to prove that the presented algorithm achieves a performance guarantee of 1.458 (1.583) for graph-TSP (graph-TSPP) on general graphs. However, similar to ours, his analysis degrades as a function of the value of the linear programming relaxation. In particular, the approximation guarantee of the algorithm for graph-TSP is at most $4/3+1/9$ when the linear programming value equals the number of nodes of the graph. We therefore think that it would be very interesting to investigate whether there is an analysis that does not degrade with an increasing value of the linear programming relaxation.

Finally, we note that the framework of removable pairings is straightforward to generalize to general metrics, but the problem of finding a large enough removable pairing in such graphs in order to improve on Christofides' algorithm remains open.

Acknowledgment

We would like to thank Johan Håstad for useful comments and stimulating discussions. This research was supported by ERC Advanced investigator grant 226203. Research of the second author was also partially funded by the ERC grant 228021-ECCSciEng.

References

[1] S. Arora, "Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems," *Journal of the ACM*, vol. 45, no. 5, pp. 753–782, 1998.

[2] F. Barahona, "Fractional packing of T-joins," *SIAM Journal on Discrete Mathematics*, vol. 17, no. 4, pp. 661–669, 2004.

[3] S. Boyd, R. Sitters, S. van der Ster, and L. Stougie, "TSP on cubic and subcubic graphs," in *Proc. 15th Conference on Integer Programming and Combinatorial Optimization (IPCO 11)*, ser. Lecture Notes in Computer Science, vol. 6655. Springer, 2011, pp. 65–77.

[4] N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem," Graduate School of Industrial Administration, Carnegie-Mellon University, Tech. Rep. 388, 1976.

[5] G. Cornuéjols, J. Fonlupt, and D. Naddef, "The traveling salesman problem on a graph and some related integer polyhedra," *Mathematical Programming*, vol. 33, no. 1, pp. 1–27, 1985.

[6] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Operations Research*, vol. 2, pp. 393–410, 1954.

[7] J. Edmonds, "Maximum matching and a polyhedron with $0,1$ vertices," *Journal of Research of the National Bureau of Standards*, vol. 69, pp. 125–130, 1965.

[8] G. N. Frederickson and J. Ja'ja', "On the relationship between the biconnectivity augmentation and travelling salesman problems," *Theoretical Computer Science*, vol. 19, no. 2, pp. 189–201, 1982.

[9] D. Gamarnik, M. Lewenstein, and M. Sviridenko, "An improved upper bound for the TSP in cubic 3-edge-connected graphs," *Operations Research Letters*, vol. 33, no. 5, pp. 467–474, 2005.

[10] M. X. Goemans, "Worst-case comparison of valid inequalities for the TSP," *Mathematics and Statistics*, vol. 69, no. 1, pp. 335–349, 1995.

[11] M. X. Goemans and D. J. Bertsimas, "On the parsimonious property of connectivity problems," in *Proc. 1st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 90)*, 1990, pp. 388–396.

[12] M. Grigni, E. Koutsoupias, and C. H. Papadimitriou, "An approximation scheme for planar graph TSP," in *Proc. 36th Annual Symposium on Foundations of Computer Science (FOCS 95)*, 1995, pp. 640–645.

[13] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, ser. Algorithms and Combinatorics. Springer, 1988, vol. 2.

[14] M. Held and R. M. Karp, "The traveling-salesman problem and minimum spanning trees," *Operations Research*, vol. 18, pp. 1138–1162, 1970.

[15] J. A. Hoogeveen, "Analysis of Christofides' heuristic: some paths are more difficult than cycles," *Operations Research Letters*, vol. 10, no. 5, pp. 291–295, 1991.

[16] A. Kaneko, A. Kelmans, and T. Nishimura, "On packing 3-vertex paths in a graph," *Journal of Graph Theory*, vol. 36, no. 4, pp. 175–197, 2001.

[17] J. S. B. Mitchell, "Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems," *SIAM Journal on Computing*, vol. 28, no. 4, pp. 1298–1309, 1999.

[18] T. Mömke and O. Svensson, "Approximating graphic TSP by matchings," *arXiv*, 2011, arXiv:1104.3090v1.

[19] C. L. Monma, B. S. Munson, and W. R. Pulleyblank, "Minimum-weight two-connected spanning networks," *Mathematical Programming*, vol. 46, no. 1, pp. 153–171, 1990.

[20] M. Mucha, "Improved analysis for graphic TSP approximation via matchings," *arXiv*, 2011, arXiv:1108.1130v1.

[21] D. Naddef and W. R. Pulleyblank, "Matchings in regular graphs," *Discrete Mathematics*, vol. 34, no. 3, pp. 283–291, 1981.

[22] S. Oveis Gharan, A. Saberi, and M. Singh, "A randomized rounding approach to the traveling salesman problem," in *Proc. 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 11)*, 2011, to appear.

[23] C. H. Papadimitriou and S. Vempala, "On the approximability of the traveling salesman problem," *Combinatorica*, vol. 26, no. 1, pp. 101–120, 2006.

[24] A. Schrijver, *Combinatorial Optimization*. Springer, 2003.

[25] D. B. Shmoys and D. P. Williamson, "Analyzing the Held-Karp TSP bound: a monotonicity property with application," *Information Processing Letters*, vol. 35, no. 6, pp. 281–285, 1990.

[26] L. A. Wolsey, "Heuristic analysis, linear programming and branch and bound," in *Combinatorial Optimization II*, ser. Mathematical Programming Studies. Springer, 1980, vol. 13, pp. 121–134.

APPENDIX

Lemma 2.1 (Restated) *Let G be a connected graph. If there is an r-approximation algorithm for graph-TSP on each 2-vertex-connected subgraph H of G (with respect to $OPT_{LP}(H)$) then there is an r-approximation algorithm for graph-TSP on G (with respect to $OPT_{LP}(G)$).*

Proof: Let $\mathcal{A}$ be an r-approximation algorithm for graph-TSP on each 2-vertex connected subgraph H of G. We shall now define an r-approximation algorithm $\mathcal{A}'$ for G as follows:

1) If G is 2-vertex connected then return the graph-TSP solution obtained by running $\mathcal{A}$ on G.
2) Otherwise, let v be a cut vertex whose removal results in the components $C_1, C_2, \ldots, C_l$ with $l > 1$. Recursively run $\mathcal{A}'$ on the l subgraphs $G_1, \ldots, G_l$ induced by $C_i \cup \{v\}$ and return the union of the obtained Euleriean subgraphs.

From the description of $\mathcal{A}'$ it is clear that it returns a graph-TSP solution, i. e., a conneced Eulerian multigraph. Moreover, as a vertex is selected as cut vertex at most once, $\mathcal{A}'$ terminates in time bounded by a polynomial in the running time of $\mathcal{A}$.

It remains to verify the cost of the solution compared to the Held-Karp lower bound. We do so by induction on the depth of the recursion. In the base case no recursive calls are made so the solution is that returned by $\mathcal{A}$ which by assumption is at most $r \cdot OPT_{LP}(G)$.

Now consider the inductive step when a cut vertex v of G is selected whose removal results in components $C_1, C_2, \ldots, C_l$ with $l > 1$. Let E_i be the multiset of edges of the connected Eulerian multigraph obtained for the graph G_i. With this notation the Eulerian multigraph of G returned by $\mathcal{A}'$ is induced by $\bigcup_{i=1}^{\ell} E_i$ and we need to prove $\sum_{i=1}^{\ell} |E_i| \leq r \cdot OPT_{LP}(G)$. By the induction hypothesis, we have $\sum_{i=1}^{\ell} |E_i| \leq r \cdot \sum_{i=1}^{\ell} OPT_{LP}(G_i)$ and it is thus sufficient to prove $\sum_{i=1}^{\ell} OPT_{LP}(G_i) \leq OPT_{LP}(G)$.

To this end, let x be an optimal solution to $LP(G)$ and let x^i denote its restriction to the subgraph G_i. As each constraint in $LP(G_i)$ has an identical constraint in $LP(G)$ (using that v is a cut vertex), x^i is also a solution to $LP(G_i)$ and hence $OPT_{LP}(G) \geq \sum_{i=1}^{\ell} OPT_{LP}(G_i)$, which completes the inductive step and the proof of the lemma. ∎