Improved Approximations for Cubic Bipartite and Cubic TSP

Anke van Zuylen*
Department of Mathematics
The College of William and Mary, Williamsburg, VA, 23185, USA
anke@wm.edu

Abstract

We show improved approximation guarantees for the traveling salesman problem on cubic bipartite graphs and cubic graphs. For cubic bipartite graphs with n nodes, we improve on recent results of Karp and Ravi by giving a "local improvement" algorithm that finds a tour of length at most $5/4n - 2$. For 2-connected cubic graphs, we show that the techniques of Mömke and Svensson can be combined with the techniques of Correa, Larré and Soto, to obtain a tour of length at most $(4/3 - 1/8754)n$.

1 Introduction

The traveling salesman problem (TSP) is one of the most famous and widely studied combinatorial optimization problems. Given a set of cities and pairwise distances, the goal is to find a tour of minimum length that visits every city exactly once. Even if we require the distances to form a metric, the problem remains NP-hard. The classic Christofides' algorithm [5] finds a tour that has length at most $\frac{3}{2}$ times the length of the optimal tour. Despite much effort in the 35 years following Christofides's result, we do not know any algorithms that improve on this guarantee.

One approach that has often been useful in designing approximation algorithms is the use of linear programming. In this context, a major open question is to determine the integrality gap of the subtour elimination linear program or Held-Karp relaxation [7, 9]; the integrality gap is the worst-case ratio of the length of an optimal tour to the optimal value of the relaxation.

*Supported in part by NSF Prime Award: HRD-1107147, Women in Scientific Education (WISE) and by a grant from the Simons Foundation (#359525, Anke Van Zuylen).

Examples are known in which the length of the optimal tour is $\frac{4}{3}$ times the value of the Held-Karp relaxation, and a major open question is whether this is tight.

Recent years have seen some exciting progress towards answering this question on graph metrics, also called the graph-TSP. In this special case of the metric TSP, we are given an unweighted graph $G = (V, E)$ in which the nodes represent the cities, and the distance between two cities is equal to the shortest path in G between the corresponding nodes. Examples are known in which the ratio between the length of the optimal tour and the Held-Karp relaxation is $\frac{4}{3}$, where the examples are in fact graph-TSP instances with an underlying graph G that is 2-connected and subcubic (every node has degree at most three).

The graph-TSP thus captures many of the obstacles that have prevented us from obtaining improved approximations for general metrics, and much recent research has focused on finding improved algorithms for the graph-TSP. The first improvement for graph-TSP metrics is due to Gamarnik, Lewenstein and Sviridenko [8], who show an approximation guarantee strictly less than $\frac{3}{2}$ for cubic, 3-connected graphs. Aggarwal, Garg and Gupta [1] give a $\frac{4}{3}$-approximation algorithm for this case. Boyd, Sitters, Van der Ster, and Stougie [3] show that there is a $\frac{4}{3}$-approximation algorithm for any cubic graph, and Mömke and Svensson [11] show this holds also for subcubic graphs.

Mömke and Svensson also show a 1.461-approximation algorithm if we make no assumptions on the underlying graph G. Mucha [12] improves their analysis to show an approximation guarantee of $\frac{13}{9}$. Sebő and Vygen [13] combine the techniques of Mömke and Svensson with a clever use of ear decompositions, which yields an approximation ratio of 1.4.

As mentioned previously, for subcubic graphs, examples exist that show that we cannot obtain better approximation guarantees than $\frac{4}{3}$ unless we use a stronger lower bound on the optimum than the Held-Karp relaxation or subtour elimination linear program. Correa, Larré and Soto [6] show that this is not the case for cubic graphs. They refine the techniques of Boyd et al. [3] and show how to find a tour of length at most $\left(\frac{4}{3} - \frac{1}{61236}\right) n$ for the graph-TSP on a 2-connected cubic graph G, where n is the number of nodes. Correa, Larré and Soto also consider the graph-TSP on planar cubic bipartite 3-connected graphs, and give a $(\frac{4}{3} - \frac{1}{18})$-approximation algorithm. Planar cubic bipartite 3-connected graphs are known as Barnette graphs, and a long-standing conjecture in graph theory by Barnette [2] states that all planar cubic bipartite 3-connected graphs are Hamiltonian. Recently, Karp and Ravi [10] gave a $\frac{9}{7}$-approximation algorithm for the graph-TSP on a superset of Barnette graphs, cubic bipartite graphs.

In this paper, we give two results that improve on the results for cubic graph-TSP. For the graph-TSP on (non-bipartite) cubic graphs, we show that the techniques of Mömke and Svensson [11] can be combined with those of Correa et al. [6] to find an approximation algorithm with guarantee $(\frac{4}{3} - \frac{1}{8754})$. We note that independent of our work, Candráková and Lukot'ka [4] showed very recently, using different techniques, how to obtain a 1.3-approximation algorithm for the graph-TSP on cubic graphs. For connected bipartite cubic graphs, we give an algorithm that finds a tour of length at most $\frac{5}{4}n - 2$, where n is the number of nodes. The idea behind our algorithm is the same as that of many previous papers, namely to find a cycle cover (or 2-factor) of the graph with a small number of cycles. Our algorithm is basically a simple "local improvement" algorithm. The key idea for the analysis is to assign the size of each cycle to the nodes contained in it in a clever way; this allows us to give a very simple proof that the algorithm returns a 2-factor with at most $n/8$ components. We also give an example that shows that the analysis is tight, even if we relax a certain condition in the algorithm that restricts the cases when we allow the algorithm to move to a new solution.

The remainder of this paper is organized as follows. In Section 2, we describe and analyze our algorithm for the graph-TSP on cubic bipartite graphs, and in Section 3, we give our improved result for non-bipartite cubic graphs.

2 The Graph-TSP on Cubic Bipartite Graphs

In the graph-TSP, we are given a graph $G = (V, E)$, and for any $u, v \in V$, we let the distance between u and v be the number of edges in the shortest path between u and v in G. The goal is to find a tour of the nodes in V that has minimum total length. A 2-factor of G is a subset of edges $F \subseteq E$, such that each node in V is incident to exactly two edges in F. Note that if F is a 2-factor, then each (connected) component of (V, F) is a simple cycle. If C is a component of (V, F), then we will use $V(C)$ to denote the nodes in C and $E(C)$ to denote the edges in C. The size of a cycle C is defined to be $|E(C)|$ (which is of course equal to $|V(C)|$). Sometimes, we consider a component of $(V, F \setminus E')$ for some $E' \subset F$. A component of such a graph is either a cycle C or a path P. We define the length of a path P to be the number of edges in P.

The main idea behind our algorithm for the graph-TSP in cubic bipartite graphs (and behind many algorithms for variants of the graph-TSP given in the literature) is to find a 2-factor F in G such that (V, F) has a a small number of cycles, say k. We can then contract each cycle of the 2-factor, find a spanning tree on the contracted graph, and add two copies of the corresponding edges to the 2-factor. This yields a spanning Eulerian (multi)graph containing $n + 2(k - 1)$ edges. By finding a Eulerian walk in this graph and shortcutting, we get a tour of length at most $n + 2k - 2$. In order to get a good algorithm for the graph-TSP, we thus need to show how to find a 2-factor with few cycles, or, equivalently, for which the average size of the cycles is large.

In Section 2.2, we give an algorithm for which we prove in Lemma 5 that, given a cubic bipartite graph $G = (V, E)$, it returns a 2-factor with average cycle size at least 8. By the arguments given above, this implies the following result.

Theorem 1. *There exists a $\frac{5}{4}$-approximation algorithm for the graph-TSP on cubic bipartite graphs.*

Before we give the ideas behind our algorithm and its analysis in Section 2.1, we begin with the observation that we may assume without loss of generality that the graph has no "potential 4-cycles": a set of 4 nodes S will be called a potential 4-cycle if there exists a 2-factor in G that contains a cycle with node set exactly S. The fact that we can modify the graph so that G has no potential 4-cycles was also used by Karp and Ravi [10].

Lemma 1. *To show that every simple cubic bipartite graph $G = (V, E)$ has a 2-factor with at most $|V|/8$ components, it suffices to show that every simple cubic bipartite graph $G' = (V', E')$ with no potential 4-cycles has a 2-factor with at most $|V'|/8$ components.*

Proof. We show how to contract a potential 4-cycle S in G to get a simple cubic bipartite graph G' with fewer nodes than the original graph, and how, given a 2-factor with average component size 8 in G', we can uncontract S to get a 2-factor in G without increasing the number of components.

Let $S = \{v_1, v_2, v_3, v_4\}$ be a potential 4-cycle in G, i.e., $E[S]$ contains 4 edges, say $\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_1, v_4\}$, and there exists no node $v_5 \notin S$ that is incident to two nodes in $\{v_1, v_2, v_3, v_4\}$ (since in that case a 2-factor containing a cycle with node set S would have v_5 as an isolated node, and this cannot be a 2-factor since v_5 must have degree 2 in a 2-factor).

We contract S, by identifying v_1, v_3 to a new node v_{odd}, and identifying v_2, v_4 to a new node v_{even}. We keep a single copy of the edge $\{v_{\text{odd}}, v_{\text{even}}\}$.

The new graph G' is simple, cubic and bipartite, and $|V'| = |V| - 2$. See Figure 1 for an illustration.

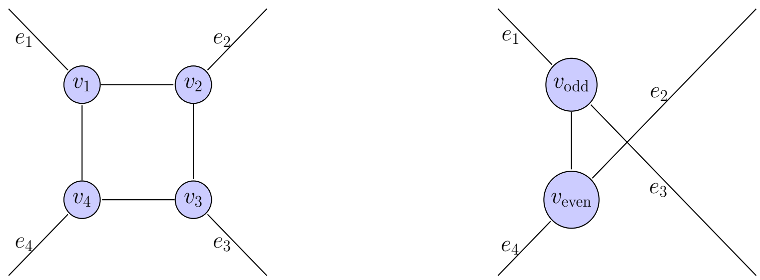

Figure 1: The 4-cycle on the left is contracted, by identifying v_1, v_3 to a new node v_{odd}, and identifying v_2, v_4 to a new node v_{even}, and keeping a single copy of the edge $\{v_{\text{odd}}, v_{\text{even}}\}$, to obtain the simple cubic bipartite graph on the right.

Given any 2-factor in G', we can "uncontract" S and find a 2-factor in G with at most as many components as the 2-factor in G': If the 2-factor on G' does not contain $\{v_{\text{odd}}, v_{\text{even}}\}$ then it must contain the other 4 edges incident to v_{odd} and v_{even}. When uncontracting S, this gives one edge incident to each $v_i, i = 1, \ldots, 4$. Since all other node degrees are even, the graph consists of even cycles and two paths with endpoints in $\{v_1, v_2, v_3, v_4\}$. We can choose to add the edges $\{v_1, v_2\}, \{v_3, v_4\}$, or the edges $\{v_2, v_3\}, \{v_1, v_4\}$; both of these choices give a 2-factor, and at least one of the two options must give a 2-factor in which all 4 nodes in S are in the same cycle. If the 2-factor on G' does contain $\{v_{\text{odd}}, v_{\text{even}}\}$ then it must contain one other edge incident to v_{odd} and to v_{even}. When uncontracting S, this gives one edge incident to v_1 or v_3, and one edge incident to v_2 or v_4. Suppose without loss of generality the edges are incident to v_1 and v_2. Then, we add edges $\{v_2, v_3\}, \{v_3, v_4\}$, and $\{v_1, v_4\}$ to get a 2-factor. Note that it is again the case that all 4 nodes in S are in the same cycle in the resulting 2-factor. □

2.1 A Local Improvement Heuristic

A cubic bipartite graph has a perfect matching (in fact, it is the case that the edge set can be decomposed into three perfect matchings), and given a cubic bipartite graph $G = (V, E)$, we can obtain a 2-factor F by simply finding a perfect matching M and letting $F = E \backslash M$. Conversely, if F is a 2-factor for G, then $E \backslash F$ is a perfect matching in G. Now, given an arbitrary 2-factor

F_1, we can use these observations to build a second 2-factor F_2 such that most nodes that are in a small cycle in (V, F_1) are in a long cycle in (V, F_2): The 2-factor F_2 is constructed by taking the matching $E \backslash F_1$ and adding half of the edges from each cycle in (V, F_1). Since each cycle is even, its edges can be decomposed into two perfect matchings, and we may choose either one of them to add to F_2. We will say 2-factor F_2 is *locally optimal with respect to* F_1 if F_2 contains all edges in $E \setminus F_1$ and for each cycle C in (V, F_1), replacing F_2 by $F_2 \triangle E(C)$ does not reduce the number of components of (V, F_2) (where $\triangle$ denotes the symmetric difference operator).

The essence of our algorithm is to start with an arbitrary 2-factor F_1, find a 2-factor F_2 that is locally optimal with respect to F_1, and return the 2-factor among F_1, F_2 with the smallest number of components.

If we consider a 6-cycle C in (V, F_1), and a 2-factor F_2 that is locally optimal with respect to F_1, then it is not hard to see that at least two edges of C will be part of the same cycle, say D, in (V, F_2). Moreover, the fact that the graph G has no potential 4-cycles can be shown to imply that D has size at least 10. This observation motivates the condition in Lemma 2 below that for any C in (V, F_1), there should exist D in (V, F_2) of size at least 10, such that $|V(C) \cap V(D)| \geq 4$.

In Lemma 2 we show that this condition suffices to guarantee that either (V, F_1) or (V, F_2) has at most $|V|/8$ cycles. In Lemma 3 we show the condition holds for F_2 that is locally optimal for F_1, provided that all cycles in F_1 are *chordless*: An edge $\{x, y\}$ is a *chord* for cycle C in (V, F_1), if $x, y \in C$, and $\{x, y\} \in E \setminus F_1$; a cycle will be referred to as chorded if it has at least one chord, and chordless otherwise.

A few more details are needed to deal with the general case when F_1 is not necessarily chordless; these are postponed to Section 2.2.

Lemma 2. *Let $G = (V, E)$ be a simple cubic bipartite graph that has no potential 4-cycles, let F_1 and F_2 be 2-factors in G, such that for any cycle C in (V, F_1), there exists a cycle D in (V, F_2) of size at least 10 such that $|V(C) \cap V(D)| \geq 4$. Then either (V, F_1) or (V, F_2) has at most $|V|/8$ components.*

Proof. Let K_i be the number of components of (V, F_i) for $i = 1, 2$. Note that it suffices to show that $\gamma K_1 + (1 - \gamma) K_2 \leq |V|/8$ for some $0 \leq \gamma \leq 1$.

In order to do this, we introduce a value $\alpha(v)$ for each node. This value is set based on the size of the cycle containing v in the second 2-factor, and they will satisfy $\sum_{v \in D} \alpha(v) = 1$ for every cycle D in the second 2-factor (V, F_2). Hence, we have that $\sum_{v \in V} \alpha(v)$ is equal to the number of cycles in

(V, F_2). We will then show that the condition of the lemma guarantees that for a cycle C in (V, F_1),

$$\sum_{v \in C} \alpha(v) \leq \frac{1}{6}|V(C)| - \frac{1}{3}. \tag{*}$$

This suffices to prove what we want: we have $K_2 = \sum_{v \in V} \alpha(v) \leq \frac{1}{6}|V| - \frac{1}{3}K_1$, which is the same as $\frac{1}{4}K_1 + \frac{3}{4}K_2 \leq \frac{1}{8}|V|$.

The basic idea to setting the α-values is that if v is in a cycle D in (V, F_2) of size k, then we have $\alpha(v) = \frac{1}{k}$. The only exception to this rule is when D has size 10; in this case we set $\alpha(v)$ for $v \in D$ to either $\frac{1}{6}$ or $\frac{1}{12}$. There will be exactly 8 nodes with $\alpha(v) = \frac{1}{12}$ and 2 nodes with $\alpha(v) = \frac{1}{6}$. The nodes v in D with $\alpha(v) = \frac{1}{12}$ are chosen in such a way that, if there is a cycle C in (V, F_1) containing at least 4 nodes in D, then at least 4 of the nodes in $V(C) \cap V(D)$ will have $\alpha(v) = \frac{1}{12}$. It is possible to achieve this, since the fact that D has 10 nodes implies that (V, F_1) can contain at most two cycles that intersect D in 4 or more nodes.

It is easy to see that (*) holds: by the condition in the lemma, any cycle C contains at least 4 nodes v such that $\alpha(v) \leq \frac{1}{12}$. Since we assumed in addition that G has no potential 4-cycles, we also know that $\alpha(v) \leq \frac{1}{6}$ for all other $v \in V(C)$. Hence $\sum_{v \in C} \alpha(v) \leq \frac{1}{6}(|V(C)| - 4) + 4 \cdot \frac{1}{12} = \frac{1}{6}|V(C)| - \frac{1}{3}$. $\square$

By Lemma 2, it is enough to find a 2-factor F_2 that satisfies that every cycle in the first 2-factor, F_1, has at least 4 nodes in some "long" cycle of F_2 (where "long" is taken to be size 10 or more). The following lemma states that a locally optimal F_2 satisfies this condition, provided that F_1 is chordless.

Lemma 3. *Let $G = (V, E)$ be a simple cubic bipartite graph that has no potential 4-cycles, let F_1 be a chordless 2-factor in G, and let F_2 be a 2-factor that is locally optimal with respect to F_1. Then for any cycle C in (V, F_1), there exists a cycle D in (V, F_2) of size at least 10 such that $|V(C) \cap V(D)| \geq 4$.*

Proof. Suppose F_2 is locally optimal with respect to F_1, and assume by contradiction that there is some cycle C in (V, F_1) such that (V, F_2) contains no cycle of size at least 10 that intersects C in at least 4 nodes. Let $F_2' = F_2 \triangle E(C)$; we will show that (V, F_2') has fewer components than (V, F_2), contradicting the fact that F_2 is locally optimal with respect to F_1.

Consider an arbitrary cycle D in (V, F_2) that intersects C. We will first show that any node v in D will be in a cycle in (V, F_2') that is at least as large as D. This shows that the number of cycles of (V, F_2') is at most the number of cycles of (V, F_2). We then show that it is not possible that for every node, its cycle in (V, F_2') is the same size as the cycle containing it in (V, F_2).

If D contains exactly one edge, say e, in C, then in (V, F_2'), the edge e in D is replaced by an odd-length path. Hence, in this case the nodes in D will be contained in a cycle in (V, F_2') that is strictly larger than D.

If D contains $k > 1$ edges in C, then D has size at most 8, since otherwise D contradicts our assumption that no cycle exists in (V, F_2) of size at least 10 that intersects C in at least 4 nodes. We now show this implies that D has size exactly 8 and $k = 2$: The size of D is either 6 or 8 since G has no potential 4-cycles. Note that D alternates edges in C and odd-length paths in $(V, F_2 \setminus E(C))$. Since C is chordless, the paths cannot have length 1 and must thus have length at least 3. We thus have that D must consist of exactly two edges from C, say e_1, e_2, separated by two paths of length 3, say P_1, P_2.

Since P_1 and P_2 do not contain edges in C, (V, F_2') also contains the edges in P_1 and P_2. Hence, to show that all nodes in D are in cycles of size at least 8 in (V, F_2'), it now suffices to show that the cycles containing P_1 and P_2 in (V, F_2') have size at least 8. Consider the cycle in (V, F_2') containing P_1; besides P_1, the cycle contains another path, say P_1', connecting the endpoints of P_1, and this path must have odd length ≥ 3 since G is bipartite and has no potential 4-cycles. Furthermore, P_1' starts and ends with an edge in C, by definition of $F_2' = F_2 \triangle E(C)$. Note that P_1' thus cannot have length 3, as this would imply that the middle edge in P_1' is a chord for C. So P_1' has length at least 5, and the cycle in (V, F_2') containing P_1 thus has size at least 8. Similarly, the cycle containing P_2 in (V, F_2') has size at least 8.

We have thus shown that all nodes in D are in cycles of size at least $|V(D)|$ in (V, F_2'), and hence, (V, F_2') has at most as many cycles as (V, F_2). Furthermore, it follows from the argument given above that the number of cycles in (V, F_2) and (V, F_2') is the same only if all nodes in C are in cycles of size 8 in both (V, F_2) and (V, F_2') and each such cycle consists of two edges from $E(C)$ and two paths of length 3 in $(V, F_2 \setminus E(C))$.

We now show by contradiction that the latter is impossible. Suppose C is such that both in (V, F_2) and in (V, F_2') every node in $V(C)$ is contained in a cycle containing two edges from C. Then $|V(C)|$ must be a multiple of 4, say $|V(C)| = 4k$. Let the nodes of C be labeled $1, 2, \ldots, 4k \equiv 0$, such that $\{2i + 1, 2i + 2\} \in F_2$ and $\{2i, 2i + 1\} \in F_2'$ for $i = 0, \ldots, 2k - 1$. We also define a mapping $p(i)$ for every $i = 1, \ldots, 4k$, such that $(V, F_2 \setminus E(C))$ contains a path (which, by our assumption has length 3) from i to $p(i)$ for

$i = 1, \ldots, 4k$. Observe that by the definition of the mapping, $p(p(i))$ must be equal to $i \pmod{4k}$ for $i = 1, \ldots, 4k$.

Let $p(1) = \ell$, then the fact that edge $\{1, 2\}$ is in a cycle with one other edge from C in (V, F_2) implies that either $\{\ell, \ell + 1\} \in F_2$ or $\{\ell, \ell - 1\} \in F_2$, and that either $p(2) = \ell + 1$ or $p(2) = \ell - 1$. In the first case, edge $\{2, 3\}$ must be in a cycle with $\{\ell+1, \ell+2\}$ in the second 2-factor (V, F_2'), and thus $p(3) = \ell+2$. In the second case, $\{2, 3\}$ must be in a cycle with $\{\ell-1, \ell-2\}$ in (V, F_2'), and thus $p(3) = \ell-2$. Repeating the argument shows that either $p(i) \equiv \ell + (i-1) \pmod{4k}$ for $i = 1, \ldots, 4k$, or $p(i) \equiv \ell - (i-1) \pmod{4k}$ for $i = 1, \ldots, 4k$.

The first case gives a contradiction to the fact that G is bipartite: note that $p(p(i)) \equiv 2\ell + i - 2 \pmod{4k}$ and this must be equal to i. Hence, $\ell \equiv 1 \pmod{2k}$; in other words, ℓ is odd, which cannot be the case since $(V, F_2 \setminus E(C))$ has a path of length 3 from 1 to ℓ (since $p(1) = \ell$) and if ℓ were odd, then C would have a path of even length from 1 to ℓ, and thus G would contain an odd cycle.

Now suppose that $p(i) \equiv \ell - (i-1) \pmod{4k}$ for $i = 1, \ldots, 4k$. From the previous argument we know that ℓ must be even, since otherwise G is not bipartite. But then $p(\ell/2) = \ell - (\ell/2 - 1) = \ell/2 + 1$. In other words, node $\ell/2$ is connected to node $\ell/2+1$ in $(V, F_2 \setminus E(C))$. But since the edge $\{\ell/2, \ell/2+1\}$ is either in F_2 or in F_2', $\{\ell/2, \ell/2+1\}$ is the only edge from C in its cycle in either (V, F_2) or (V, F_2'), contradicting the assumption on C. □

It may be the case that Lemma 3 also holds for cycles in (V, F_1) that do have chords, but we have not been able to prove this. Instead, there is a simple alternative operation that ensures that a cycle in (V, F_1) with a chord intersects at least one "long" cycle of size at least 10 in (V, F_2) in 4 or more nodes. The algorithm described next will add this operation, and for technical reasons it will only modify F_2 with respect to a cycle C in (V, F_1) if the cycle C does not yet intersect a long cycle in (V, F_2) in 4 or more nodes.

2.2 A 2-Factor with Average Cycle Size 8

We give our algorithm in Algorithm 1. The algorithm fixes a 2-factor F_1 and initializes F_2 to be a 2-factor that contains all edges in $E \setminus F_1$. The algorithm then proceeds to modify F_2; note that F_1 is not changed. Figure 2 illustrates the modification to F_2 in the case of a chorded cycle C_i.

We need to prove that the set of edges F_2 remains a 2-factor throughout

Let $G = (V, E)$ be a bipartite cubic graph, with potential 4-cycles contracted using Lemma 1.
Let F_1 be an arbitrary 2-factor in G, and let $C_1, \ldots, C_k$ be the cycles in (V, F_1).
For each cycle C_i in (V, F_1), let $M(C_i) \subseteq E(C_i)$ be a perfect matching on $V(C_i)$.
Initialize $F_2 = (E \backslash F_1) \cup \bigcup_{i=1}^{k} M(C_i)$.
while *there exists a cycle* C_i *such that* $|V(C_i) \cap V(D)| < 4$ *for all cycles* D *in* (V, F_2) *of size at least 10* **do**
    if C_i *is a chordless cycle* **then**
        $F_2 \leftarrow F_2 \triangle E(C_i)$.
    else
        Let $\{x, y\}$ be a chord for C_i, let P_1, P_2 be the edge disjoint paths in C_i from x to y.
        Relabel P_1 and P_2 if necessary so that P_1 starts and ends with an edge in $F_1 \setminus F_2$.
        $F_2 \leftarrow (F_2 \triangle E(P_1)) \setminus \{x, y\}$.
    end if
end while
Uncontract the 4-cycles in (V, F_j) for $j = 1, 2$ using Lemma 1.
Return the 2-factor among F_1, F_2 with the smaller number of components.

Algorithm 1: Approximation Algorithm for Cubic Bipartite TSP

the course of the algorithm, that the algorithm terminates, and that upon termination, either (V, F_1) or (V, F_2) has at most $|V|/8$ components. The latter is clear: *if* the algorithm terminates, then the condition of Lemma 2 is satisfied, and therefore one of the two 2-factors has at most $|V|/8$ components.

To show that F_2 is a 2-factor and that the algorithm terminates is a little more subtle. In order to show this, it will be helpful to know that each cycle in (V, F_i) alternates edges in $F_i \cap F_{i+1}$ and edges in $F_i \setminus F_{i+1}$ for $i = 1, 2$ (where subscripts are modulo 2, so $F_3 \equiv F_1$). This is true initially, however, it is not the case that this property continues to hold for all cycles. We will show that it does hold in certain cases, which turn out to be exactly the cases "when we need it". In the following, we will say a cycle or path in (V, F_i) is *alternating* (for F_1 and F_2) if it alternates edges in $F_i \cap F_{i+1}$ and $F_i \setminus F_{i+1}$. We will say that a cycle C in (V, F_1) is *violated* if there exists no

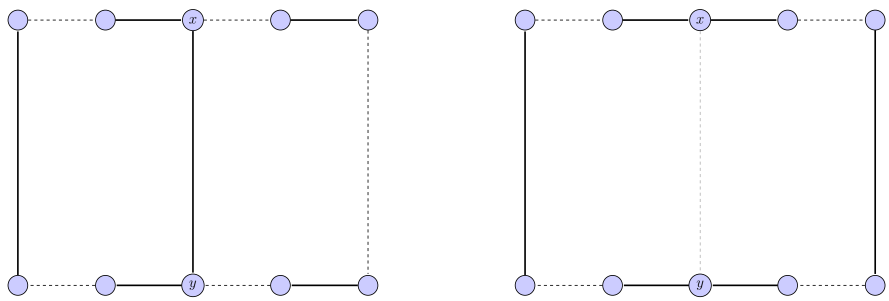

Figure 2: The figure on the left shows a chorded cycle C in F_1 of size 10 and all edges in G that have both endpoints in C. The dashed edges are in $F_1 \setminus F_2$, and non-dashed edges are in F_2 (where not all edges in F_2 that are incident on the nodes are shown). The figure on the right shows how Algorithm 1 would update F_2.

D of size at least 10 in (V, F_2) such that $|V(C) \cap V(D)| \geq 4$.

Lemma 4. *Algorithm 1 maintains that F_2 is a 2-factor that satisfies the following properties:*

(1) if C in (V, F_1) is violated, then C is alternating for F_1 and F_2;

(2) if D in (V, F_2) is not alternating for F_1 and F_2, then D has size at least 10.

Proof. We prove the lemma by induction on the algorithm. Initially, F_2 consists of $E \setminus F_1$ and $\bigcup_{i=1}^{k} M(C_i)$, which are two edge-disjoint perfect matchings on V. Hence, F_2 is a 2-factor, and the two properties hold for all cycles in (V, F_1) and (V, F_2).

Suppose the lemma holds and we modify F_2 by considering some violated cycle C. The two properties of the lemma imply the following:

Claim 1. *If C is violated, then $(V, F_2 \setminus E(C))$ consists of even cycles and odd-length paths, where paths that are not alternating for F_1 and F_2 have length at least 9.*

Proof of Claim: For each path in the graph $(V, F_2 \setminus E(C))$ there exists some cycle D in (V, F_2) such that the path results when removing $E(C) \cap E(D)$ from D. If D is alternating for F_1 and F_2, then the path must have the same property, and it must start and end with an edge in $F_2 \setminus F_1$. Hence, the path must have odd length if D is alternating. If D is not alternating then D has size at least 10 by Property (2), so C can have at most one edge in common with D, since otherwise C is not violated. Hence, the path obtained by removing the unique edge in $E(C) \cap E(D)$ has length at least 9, and its length must be odd, since D is an even cycle. $\diamond$

If C is chordless, then we modify F_2 to $F_2' = F_2 \triangle E(C)$. Clearly, F_2' is again a 2-factor, and Property (1) remains satisfied. Furthermore, any cycle in (V, F_2') that is not alternating for F_1 and F_2' either also existed in (V, F_2) and hence it has size at least 10, since Property (2) holds for F_2, or the cycle contains a path in $(V, F_2 \setminus E(C))$ that is not alternating for F_1 and F_2, and this path has length at least 9 by the claim. So Property (2) holds for F_2'.

Now consider the modification of F_2 when considering a chorded cycle C in (V, F_1). Let P_1 be as defined in the algorithm. First consider $F_2' = F_2 \triangle E(P_1)$; every node is incident to two edges in F_2', except for x and y, which are incident to three edges in F_2', namely two edges in $E(C)$ plus the edge $\{x, y\}$. Hence, removing $\{x, y\}$ will give a new 2-factor, say F_2''. The modification from F_2 to F_2'' is exactly the modification made to F_2 by the algorithm.

We now show that the two properties are satisfied. Clearly, C is not alternating for F_1 and F_2'', so in order to maintain Property (1), we need to show that C is no longer violated. To do this, we show that (V, F_2'') contains a cycle of size at least 10 that contains x, y and their 4 neighbors in C. First, suppose by contradiction that after removing $\{x, y\}$, x and y are not in the same cycle. Consider the component of (V, F_2'') containing x: starting from x, it alternates edges in $E(C)$ and paths in $F_2 \setminus E(C)$, starting and ending with an edge in $E(C)$. By Claim 1 the paths in $(V, F_2 \setminus E(C))$ have odd length, and hence the component containing x must be an odd cycle, contradicting the fact that G is bipartite. So, x and y must be in the same cycle in (V, F_2''). This cycle must thus consist of two odd-length paths from x to y, each starting and ending with an edge in $F_2'' \cap F_1$. These paths cannot have length 3, because this would imply that the path plus the edge $\{x, y\}$ would form a potential 4-cycle. Hence, the cycle in (V, F_2'') containing x and y has size at least 10.

For Property (2), note that any cycle D in (V, F_2'') that is not alternating for F_1 and F_2'' either (i) existed in (V, F_2) and therefore has size at least 10, or (ii) contains x and y and we showed above that this cycle has size at least 10, or (iii) it contains a path in $(V, F_2 \setminus E(C))$ that is not alternating for F_1 and F_2, and by the claim this path has length at least 9. Hence, Property (2) is satisfied by F_2''. $\square$

Lemma 5. *Given a cubic bipartite graph $G = (V, E)$, Algorithm 1 returns a 2-factor in G with at most $|V|/8$ components.*

Proof. By Lemma 1 it suffices to show that the current lemma holds if G has no potential 4-cycles. By the termination condition of Algorithm 1 and Lemma 2, the 2-factor returned by the algorithm does indeed have at most $|V|/8$ components, so it remains to show that the algorithm always returns a 2-factor.

By Lemma 4, the algorithm maintains two 2-factors F_1 and F_2. Observe that if a cycle C is not violated, then this continues to hold throughout the remainder of the algorithm: Let D be a cycle of size at least 10 in (V, F_2) such that $|V(C) \cap V(D)| \geq 4$. The only possible changes to D will be caused by a violated cycle C', which necessarily contains at most one edge in D: by Lemma 4 C' is alternating for F_1 and F_2, so if C' contains more than one edge in D, C' cannot be violated. The modification of F_2 with respect to $E(C')$ can therefore only cause the cycle D to become a larger cycle D' where $V(D') \supseteq V(D)$. So D' will have size at least 10, and $|V(C) \cap V(D')| \geq 4$.

It remains to show that if we modify F_2 with respect to some violated cycle C, then C is not violated for the new 2-factor F_2'. If C is not chordless, then this holds because C is not alternating for F_1 and the new 2-factor F_2', so by Lemma 4, C is not violated. If C is chordless and violated, then by Claim 1, $(V, F_2 \setminus E(C))$ consists of even cycles and odd-length paths. The proof of Lemma 3 then shows that taking the symmetric difference of F_2 with $E(C)$ (strictly) reduces the number of components. This implies that for $F_2' = F_2 \triangle E(C)$, cycle C is not violated: otherwise, we could apply the same arguments to show that $(V, F_2 \triangle E(C) \triangle E(C))$ has strictly fewer cycles than (V, F_2), but this is a contradiction since $F_2 \triangle E(C) \triangle E(C) = F_2$. $\square$

In the appendix, we give an example on 48 nodes that shows our analysis of Algorithm 1 is tight. In fact, the example is also tight for the local improvement heuristic from Section 2.1 and for the local improvement heuristic we obtain if we allow Algorithm 1 to modify F_2 for cycles C that are chorded and/or do have at least two edges in a cycle of size 10 or more in (V, F_2).

3 Cubic graphs

We now consider cubic graphs, in other words, we drop the requirement that the graph is bipartite. We assume the graph is 2-connected. The best known approximation result for graph-TSP on a 2-connected cubic graphs

$G = (V, E)$ is due to Correa, Larré and Soto [6] who show how to find in polynomial time a tour of length at most $\left(\frac{4}{3} - \frac{1}{61236}\right)|V|$.

One obstacle for their techniques are chorded 4-cycles, i.e., a set of 4 nodes (v_1, v_2, v_3, v_4) such that the subgraph of G induced by $\{v_1, v_2, v_3, v_4\}$ contains edges $\{v_1, v_2\}, \{v_2, v_3\}$, $\{v_3, v_4\}, \{v_4, v_1\}$ and the "chord" $\{v_2, v_4\}$. In fact, Correa et al. prove the following.

Lemma 6 (Correa, Larré and Soto [6])**.** *Consider a graph-TSP instance on a 2-connected cubic graph $G = (V, E)$, and let B be the set of nodes in G contained in a chorded 4-cycle. Then, we can find in polynomial time a tour of length at most $\frac{4}{3}|B| + (\frac{4}{3} - \frac{1}{8748})(|V \backslash B|) + 2$.*

The proof of this lemma is contained in the proof of Theorem 2 in [6]. More precisely, it is shown that there exists a distribution over tours of expected size $\sum_{v \in V} z(v) + 2$, where $z(v)$ is the "average contribution" of v. In the proof of the Theorem 2 in [6] it is shown that $\sum_{v \in V} z(v) \leq \frac{4}{3}|B| + (\frac{4}{3} - \frac{1}{8748})(|V \backslash B|)$.

On the other hand, chorded 4-cycles are "beneficial" for the analysis of the Mömke-Svensson [11] algorithm for the graph-TSP on 2-connected subcubic graphs, as we will show next.

Lemma 7. *Consider a graph-TSP instance on a 2-connected subcubic graph $G = (V, E)$, and let B be the set of nodes in G contained in a chorded 4-cycle. Then, we can find in polynomial time a tour of length at most $\frac{4}{3}|V| - \frac{1}{6}|B| - \frac{2}{3}$.*

Proof. For convenience, we will use the word "tour" to refer to a connected spanning Eulerian multigraph obtained by doubling and deleting some edges of G. Note that the graph-TSP instance on G indeed has a tour (obtained by shortcutting a Eulerian walk on this multigraph) of length at most the number of edges in the multigraph. Mömke and Svensson [11] show that there exists a probability distribution over tours, such that each edge $e \in E$ appears an even number of times (zero or twice) with probability $\frac{1}{3}$, and the expected number of edges in the tour is $\frac{4}{3}|V| - \frac{2}{3}$.

Now, we can simply contract chorded 4-cycles in a cubic graph to obtain a subcubic graph $\tilde{G}$, on which we can apply the Mömke-Svensson algorithm to find a distribution over tours with $\frac{4}{3}(|V| - \frac{3}{4}|B|) - \frac{2}{3} = \frac{4}{3}|V| - |B| - \frac{2}{3}$ edges in expectation. Next, we uncontract the chorded 4-cycles in each of the tours T: For a chorded 4-cycle (v_1, v_2, v_3, v_4) that is contracted to a node v, note that T contains an even number of edges incident to v. Hence, either both edges incident to v in $\tilde{G}$ appear an odd number of times in T, or both appear an even number of times. In the first case, we uncontract the chorded 4-cycle by adding the edges $\{v_1, v_2\}, \{v_2, v_4\}$ and $\{v_3, v_4\}$. In the second case, we uncontract the chorded 4-cycle by adding the edges $\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_4, v_1\}$.

Since each edge appears an even number of times with probability $\frac{1}{3}$, the second case happens with probability $\frac{1}{3}$, and the first case thus happens with probability $\frac{2}{3}$. Hence, the expected number of edges in T increases by $\frac{10}{3}\frac{|B|}{4}$, giving a total number of edges of $\frac{4}{3}|V| - \frac{1}{6}|B| - \frac{2}{3}$ in expectation. $\square$

Note that the bound in Lemma 6 is increasing in $|B|$ and the bound in Lemma 7 is decreasing in $|B|$. Setting the bound in Lemma 6 equal to the bound in Lemma 7 gives $|B| = \frac{1}{1459}|V|$ and thus shows that there exists a polynomial time algorithm for finding a tour of length at most $\left(\frac{4}{3} - \frac{1}{8754}\right)|V|$ for a graph-TSP instance on a 2-connected cubic graph $G = (V, E)$. By an observation of Mömke and Svensson [11], this also implies a $\left(\frac{4}{3} - \frac{1}{8754}\right)$-approximation algorithm for cubic graph-TSP, i.e., the graph G does not have to be 2-connected. We have thus shown the following result.

Theorem 2. *There exists a* $\left(\frac{4}{3} - \frac{1}{8754}\right)$*-approximation algorithm for Graph-TSP on cubic graphs.*

Acknowledgements

The author would like to thank Marcin Mucha for careful reading and pointing out an omission in a previous version, Frans Schalekamp for helpful discussions, and an anonymous reviewer for suggesting the simplified proof for the result in Section 3 for cubic non-bipartite graphs. Other anonymous reviewers are acknowledged for helpful feedback on the presentation of the algorithm for bipartite cubic graphs.

References

[1] Nishita Aggarwal, Naveen Garg, and Swati Gupta. A 4/3-approximation for TSP on cubic 3-edge-connected graphs. Available at `http://arxiv.org/abs/1101.5586`, 2011.

[2] David W. Barnette. Conjecture 5. *Recent progress in combinatorics*, 1969.

[3] Sylvia Boyd, René Sitters, Suzanne van der Ster, and Leen Stougie. The traveling salesman problem on cubic and subcubic graphs. *Math. Program.*, 144(1-2):227–245, 2014.

[4] Barbora Candráková and Robert Lukotka. Cubic TSP - a 1.3-approximation. *CoRR*, abs/1506.06369, 2015.

[5] Nicos Christofides. Worst case analysis of a new heuristic for the traveling salesman problem. Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA, 1976.

[6] José R. Correa, Omar Larré, and José A. Soto. TSP tours in cubic graphs: Beyond 4/3. *SIAM J. Discrete Math.*, 29(2):915–939, 2015. Preliminary version appeared in *ESA 2012*: 790-801.

[7] George B. Dantzig, Delbert R. Fulkerson, and Selmer M. Johnson. Solution of a large-scale traveling-salesman problem. *Operations Research*, 2:393–410, 1954.

[8] David Gamarnik, Moshe Lewenstein, and Maxim Sviridenko. An improved upper bound for the TSP in cubic 3-edge-connected graphs. *Oper. Res. Lett.*, 33(5):467–474, 2005.

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

[10] Jeremy Karp and R. Ravi. A 9/7 -approximation algorithm for graphic TSP in cubic bipartite graphs. In *(APPROX-RANDOM 2014)*, volume 28 of *LIPIcs*, pages 284–296, 2014.

[11] Tobias Mömke and Ola Svensson. Removing and Adding Edges for the Traveling Salesman Problem. *J. ACM* 63(1): 2 (2016). Preliminary version appeared in *FOCS 2011*: pages 560–569, 2011.

[12] Marcin Mucha. 13/9 -approximation for graphic TSP. *Theory Comput. Syst.*, 55(4):640–657, 2014.

[13] 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.

4 Tightness of the Analysis of Algorithm 1

We give an example of a cubic bipartite graph $G = (V, E)$ for which both the 2-factors F_1 and F_2 that result from Algorithm 1 have $|V|/8$ components.

The instance has 48 nodes, numbered 1 through 48, and (V, F_1) contains six cycles, four cycles of size 6, one cycle of size 10 and one cycle of size 14.

For brevity, we denote the cycles by only giving an ordered listing of their nodes; an edge between consecutive nodes and between the last and first node is implicit. The cycles in (V, F_1) are:

$$\begin{aligned}
C_1 &= (1, 2, 3, 4, 5, 6),\\
C_2 &= (7, 8, 9, 10, 11, 12),\\
C_3 &= (13, 14, 15, 16, 17, 18),\\
C_4 &= (19, 20, 21, 22, 23, 24),\\
C_5 &= (25, 26, 27, 28, 29, 30, 31, 32, 33, 34),\\
C_6 &= (35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48).
\end{aligned}$$

The second 2-factor (V, F_2) has six cycles as well, namely, three cycles of size 6 and three cycles of size 10. We again denote the cycles by giving an ordered listing of the nodes, but now a semicolon between subsequent nodes indicates that the nodes are connected by an edge in $F_2 \setminus F_1$, and a comma denotes that they are connected by an edge in $F_2 \cap F_1$. The cycles in (V, F_2) are

$$\begin{aligned}
D_1 &= (6, 1; 30, 31; 42, 43; 28, 29; 40, 41;), & D_4 &= (18, 13; 38, 39; 44, 45;),\\
D_2 &= (48, 35; 10, 11; 4, 5; 8, 9; 2, 3;), & D_5 &= (24, 19; 26, 27; 32, 33;),\\
D_3 &= (34, 25; 16, 17; 22, 23; 14, 15; 20, 21;), & D_6 &= (12, 7; 36, 37; 46, 47;).
\end{aligned}$$

It is straightforward to verify that every node occurs in exactly one cycle in (V, F_1) and exactly one cycle in (V, F_2), and that each cycle in (V, F_1) alternates edges in $F_1 \setminus F_2$ and edges in $F_1 \cap F_2$, and that each cycle in (V, F_2) alternates edges in $F_2 \cap F_1$ and edges in $F_2 \setminus F_1$. Furthermore, each cycle C_i in (V, F_1) has exactly two edges in a cycle D in (V, F_2) of size exactly 10.

Local Optimum

Figure 3 depicts each of the six cycles C_i in (V, F_1), together with the cycles D_j in (V, F_2) that intersect the given cycle C_i. For any of the cycles C_i, replacing F_2 by $F_2 \triangle E(C_i)$ does not decrease the number of components of (V, F_2) for any cycle C_i, nor does the modification of F_2 for a chorded cycle described in Algorithm 1.

Hence, if, rather than following Algorithm 1, we would execute one of the two possible modifications of F_2 suggested by the algorithm, as long as this reduced the number of components of (V, F_2), then the 2-factor F_2 is in fact a local optimum with respect to this process since none of the possible modifications reduces the number of components.

We note that the instance does have a Hamilton cycle. This cycle contains subpaths of more than two adjacent edges in $E \setminus F_1$, so it can never be found from F_1 and F_2 using the "moves" we defined, even if we allow "moves" that do not reduce the number of components of (V, F_2). We give the Hamilton cycle by again giving an ordered listing of the nodes, where a semicolon between subsequent nodes indicates that the nodes are connected by an edge in $E \setminus F_1$, and a comma denotes that they are connected by an edge in F_1.

$(1; 30, 29; 40, 39, 38; 13, 14, 15, 16; 25, 26; 19, 20, 21; 34, 33; 24, 23, 22; 17, 18; 45, 44, 43; 28, 27; 32, 31; 42, 41; 6, 5; 8, 7, 12, 11; 4, 3; 48, 47, 46; 37, 36, 35; 10, 9; 2, 1)$.

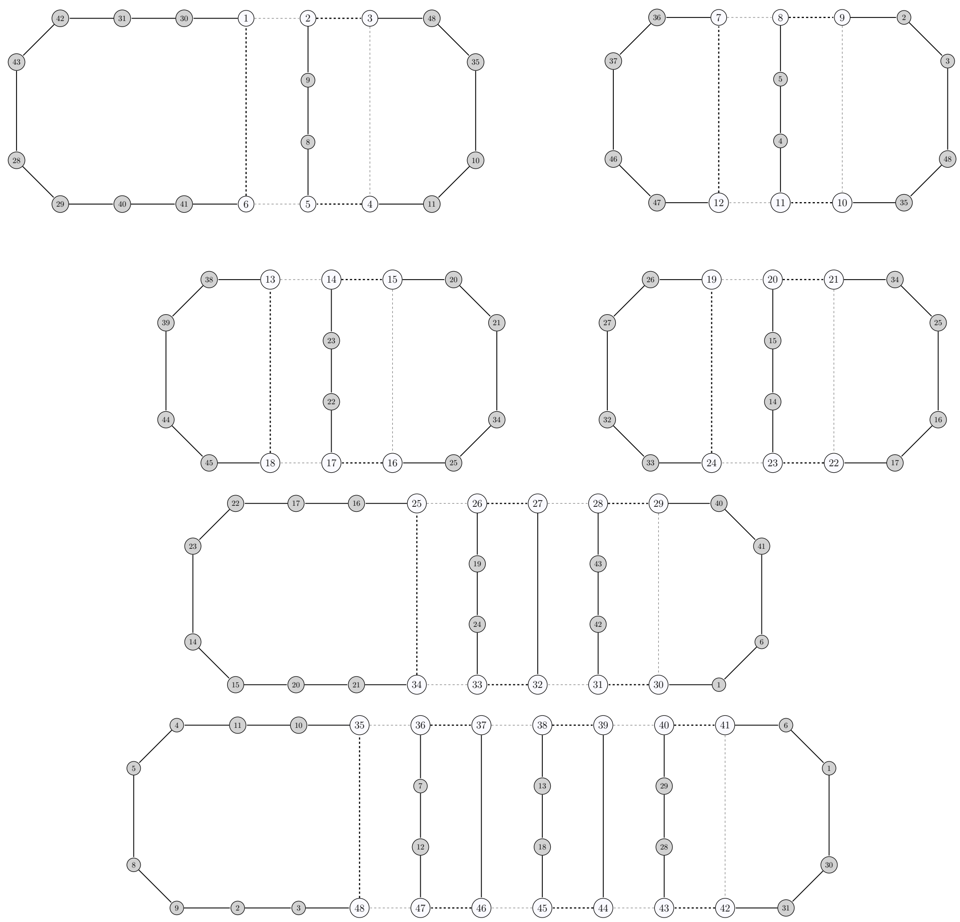

Figure 3: Depicted are six figures, one for each cycle C_i in (V, F_1) for $i = 1, \ldots, 6$. The white nodes are the nodes in the cycle C_i, and the dashed edges are the edges in $F_1 \setminus E(C_i)$. The non-dashed and thick dashed edges are edges in F_2. For each C_i, the number of components of (V, F_2) does not decrease by replacing F_2 by $F_2 \triangle E(C_i)$.