Insights from the Two-Stripe Circulant TSP: New Easily-Solvable Cases and Improved Approximations

Austin Beal, Yacine Boudabida, Samuel C. Gutekunst[1], Asta Rustad

Abstract

DRAFT PREPRINT

Keywords: Traveling Salesman Problem, Circulant, Complexity

1. Introduction and Circulant TSP

The Symmetric Traveling Salesman Problem (TSP) is a fundamental problem in combinatorial optimization and a canonical NP-hard problem. An input consists of a set of n vertices $[n] := \{1, 2, ..., n\}$ and edge costs $c_{ij} = c_{ji}$ (for $1 \leq i, j \leq n$), indicating the costs of travelling between vertices i and j. The TSP is then to find a minimum-cost Hamiltonian cycle, visiting each city exactly once.

With just this set-up, the TSP is well known to be NP-hard. An algorithm that could approximate TSP solutions in polynomial time to within any factor $\alpha > 1$ would imply P=NP (see, e.g., Theorem 2.9 in Williamson and Shmoys [31]). Thus it is common to consider special cases that restrict the edge costs. For instance, requiring costs to be *metric* (so that $c_{ij} + c_{jk} \geq c_{ik}$ for all $i, j, k \in [n]$), to correspond to distances in an underlying graph on $[n]$, to correspond to euclidean distances between n points in $\mathbb{R}^2$, to restricting $c_{ij} \in \{1, 2\}$ for all i, j (the $(1, 2)-$*TSP*). See, e.g., [2, 3, 8, 17, 18, 22, 23, 24, 25, 26, 28, 29] amoung many others.

One special case that is particularly intriguing, but where relatively little is known, is the *circulant TSP*. Circulant TSP instances are those whose edge costs can be described by a *circulant matrix*, which imposes substantial symmetry: the cost of edge $\{i, j\}$ only depends on $(i - j) \bmod n$. We implicitly assume that the edge costs are symmetric, so that circulant TSP instances can be described by a (symmetric, circulant) cost matrix with $\lfloor \frac{n}{2} \rfloor$ parameters $c_1, c_2, ..., c_{\lfloor \frac{n}{2} \rfloor}$:

$$C := (c_{i,j})_{i,j=1}^{n} = \begin{pmatrix} 0 & c_1 & c_2 & c_3 & \cdots & c_1 \\ c_1 & 0 & c_1 & c_2 & \cdots & c_2 \\ c_2 & c_1 & 0 & c_1 & \ddots & c_3 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & c_3 & c_4 & \cdots & 0 \end{pmatrix}, \quad (1)$$

with $c_0 = 0$ and $c_i = c_{n-i}$ for $i = 1, ..., \lfloor \frac{n}{2} \rfloor$. That is, the cost of traveling between vertices i and j is

$$c_{i,j} = c_{\min\{(i-j) \mod n, (j-i) \mod n\}}.$$

We interpret $\min\{(i - j) \mod n, (j - i) \mod n\}$ as the **length** of the the edge $\{i, j\}$. For instance, edges $\{1, 2\}$, $\{3, 2\}$, and $\{n, 1\}$ all have the same length 1, and thus the same cost c_1; see Figure 1. Importantly, in circulant TSP we do not make the prototypical assumption that edge costs are metric.

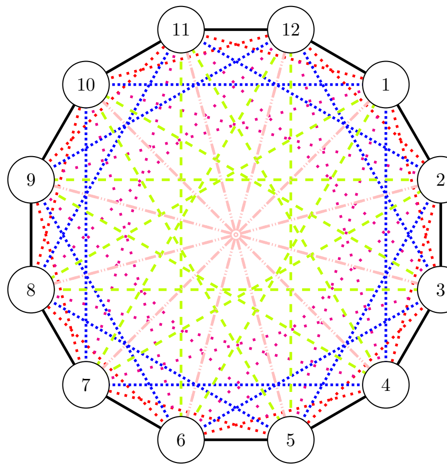

Figure 1: Circulant symmetry. Edges of a fixed length are indistinguishable and have the same cost. E.g. all edges of the form $\{v, v+1\}$ (where $v + 1$ is taken mod n) have length 1, and thus cost c_1.

Circulant TSP was first studied in the 70's, and was motivated by waste minimization (Garfinkel [10]) and reconfigurable network design (Medova [21]). Intriguingly, in the 70's Garfinkel [10] shows that circulant TSP can be easily and efficiently solved whenever the number of vertices n is prime (see Section 3). More generally, circulant symmetry imposes just enough structure to sometimes – but by no means always – make a formally hard problem tractable. It is not known if this is the case for circu-

[1]Corresponding author; sg040@bucknell.edu

lant TSP, and circulant TSP's complexity has been often cited as a significant open problem (e.g., Burkhard [6], Burkhard, Deĭneko, Van Dal, Van der Veen, and Woeginger [7], and Lawler, Lenstra, Rinnooy Kan, and Shmoys [19]).

Since the 70s, most work on circulant TSP's complexity stems has been on the simplest non-trivial special case of circulant TSP: the *two stripe symmetric circulant TSP*. This is the special case where exactly two of the edge costs $c_1, c_2, ..., c_{\lfloor \frac{n}{2} \rfloor}$ are finite. Greco and Gerace [13] and Gerace and Greco [11] made progress on this case and recently, Gutekunst, Jin, and Williamson [14] resolved it and showed that the two stripe symmetric circulant TSP problem was solvable in polynomial time. In parallel, substantial number theoretic work has gone into understanding what collections of edge lengths can constitute a Hamiltonian cycle and/or path (see, e.g., Buratti and Merola [4], Horak and Rosa, Pasotti and Pellegrini [27], Costa, Morini, Pasotti, and Pellegrini [9], and McKay and Peters [20]).

In this paper, we show three new results that build off of the work of Gutekunst, Jin, and Williamson [14]. In Section 2, we begin by briskly introducing background about circulant graphs and their Hamiltonicity, which we will repeatedly make use of. We present our first result in Section 3, which fleshes out connections between circulant TSP and number theory and shows that circulant TSP is also efficiently solvable any time the number of vertices n is a prime-squared; this is the first general complexity result for circulant TSP based on the factorization of n since Garfinkel [10] 70's result that circulant TSP could be efficiently solved when n was prime. Then, in Section 4, we study the *two-class circulant TSP*, which specializes the $(1, 2)-$TSP to circulant instances: this is the circulant TSP when the edge costs $c_1, c_2, ..., c_{\lfloor \frac{n}{2} \rfloor}$ take on exactly two values. Perhaps counter-intuitively, it turns out that the two-class problem is considerably easier than the two-stripe circulant TSP. Finally, in Section 5, we present a 10/9-approximation algorithm for finding a minimum-cost Eulerian tour on two-stripe instances (or equivalently, finding a minimum-cost Hamiltonian cycle on the metric completion of a two-stripe instance).

2. Preliminaries: Circulant Graphs and Hamiltonicity

We first define *circulant graphs* in terms of a set S of edge lengths:

Definition 2.1. *Let $S \subset \{1, ..., \lfloor \frac{n}{2} \rfloor\}$. The* **circulant graph** $C\langle S\rangle$ *is the (simple, undirected, unweighted) graph including exactly the edges whose lengths are in S. I.e., the graph with adjacency matrix $A = (a_{ij})_{i,j=1}^n$, where*

$$a_{ij} = \begin{cases} 1, & (i-j) \bmod n \in S \textit{ or } (j-i) \bmod n \in S \\ 0, & \textit{else.} \end{cases}$$

Burkard and Sandholzer [5] studied Hamiltonicity and bottleneck TSP in symmetric circulant graphs, and and used the following result:

Proposition 2.2 (Burkard and Sandholzer [5])**.** *Let $\{a_1, ..., a_t\} \subset \left[\lfloor \frac{n}{2} \rfloor\right]$ and let $\mathcal{G} = \gcd(n, a_1, ..., a_t)$. The circulant graph $C\langle\{a_1, ..., a_t\}\rangle$ has $\mathcal{G}$ components. The ith component, for $0 \le i \le \mathcal{G} - 1$, consists of $n/\mathcal{G}$ nodes*

$$\left\{ i + \lambda\mathcal{G} \bmod n : \ 0 \le \lambda \le \frac{n}{\mathcal{G}} - 1 \right\}.$$

$C\langle\{a_1, ..., a_t\}\rangle$ is Hamiltonian if and only if $\mathcal{G} = 1$.

A complete proof can be found in Burkard and Sandholzer [5], showing how to recursively construct Hamiltonian cycles whenever $\mathcal{G} = 1$ and giving rise to an $O(n \log(n))$ algorithm for finding Hamiltonian cycles whenever $\mathcal{G} = 1$. Because many of our results will lean on this proposition, we sketch the idea below. We also adopt two notational conventions: First, all vertex labels are implicitly taken modulo n (e.g. $v + a_1$ is shorthand for $(v + a_1) \mod n$, so that $\{v, v + a_1\}$ is a length-a_1 edge). Second, we use $\equiv_n$ to denote congruence modulo n.

Proof (sketch). The proof of Proposition 2.2 proceeds recursively. First, consider the graph $C\langle\{a_1\}\rangle$, consisting of exactly the length-a_1 edges. These a_1 edges form a cycle cover, consisting of $\gcd(n, a_1)$ cycles. For example, starting at vertex 1 and following length-a_1 edges yields a cycle $1, 1 + a_1, 1 + 2a_1, ..., 1 + \left(\frac{n}{\gcd(n,a_1)} - 1\right) a_1, 1$ consisting of all vertices congruent to 1 mod $\gcd(n, a_1)$. Note that

$$1 + \left(\frac{n}{\gcd(n, a_1)} - 1\right) a_1 + a_1 = 1 + \frac{n}{\gcd(n, a_1)} a_1 \equiv_n 1,$$

since $\gcd(n, a_1)$ divides a_1.

Now suppose that $\gcd(n, a_1, a_2, ..., a_{t-1}) > \gcd(n, a_1, a_2, ..., a_{t-1}, a_t)$ so that $C\langle\{a_1, a_2, ..., a_{t-1}\}\rangle$ has strictly more components than $C\langle\{a_1, a_2, ..., a_t\}\rangle$. Burkard and Sandholzer appeal to circulant symmetry and show that length-a_t edges can be used to merge the $\gcd(n, a_1, ..., a_{t-1})$ Hamiltonian cycles on components of $C\langle\{a_1, a_2, ..., a_{t-1}\}\rangle$ into $\gcd(n, a_1, ..., a_t)$ Hamiltonian cycles on the components of $C\langle\{a_1, a_2, ..., a_t\}\rangle$; once $\gcd(n, a_1, ..., a_t) = 1$, this process terminates in a Hamiltonian cycle on all n vertices.

Consider some Hamiltonian cycle $v_1, v_2, ..., v_{n/\gcd(n,a_1,...,a_{t-1})}, v_1$ on the component of $C\langle\{a_1, a_2, ..., a_{t-1}\}\rangle$ containing vertex 1. Burkard and Sandholzer show how to merge together the $\gcd(n, a_1, ..., a_{t-1})/\gcd(n, a_1, ..., a_t)$ components of $C\langle\{a_1, a_2, ..., a_{t-1}\}\rangle$ consisting of vertices congruent to 1 mod $\gcd(n, a_1, ..., a_t)$. Circulant symmetry allows us to add a multiple of a_t and translate our cycle to the

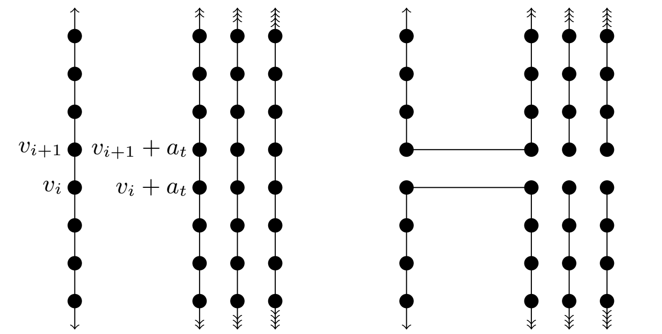

Figure 2: The recursive approach to building Hamiltonian cycles on circulant graphs. Edges with arrows "wrap around" vertically to the matching edge with the same number of arrowheads.

components of $C\langle\{a_1, a_2, ..., a_{t-1}\}\rangle$ that we will merge. More formally, for $k = 0, 1, ..., \frac{\gcd(n,a_1,...,a_{t-1})}{\gcd(n,a_1,...,a_t)}$, we have cycles

$$v_1 + ka_t, v_2 + ka_t, ..., v_{n/\gcd(n,a_1,...,a_{t-1})} + ka_t, v_1 + ka_t.$$

Circulant symmetry then allows us to iteratively merge these cycles. If $\{v_i, v_{i+1}\}$ is an edge in the cycle $v_1, v_2, ..., v_{n/\gcd(n,a_1,...,a_{t-1})}, v_1$, we merge in the cycle $v_1 + a_t, v_2 + a_t, ..., v_{n/\gcd(n,a_1,...,a_{t-1})} + a_t, v_1 + a_t$ as follows: delete $\{v_i, v_{i+1}\}$ and $\{v_i + a_t, v_{i+1} + a_t\}$, and add in edges $\{v_i, v_i + a_t\}$ and $\{v_{i+1}, v_{i+1} + a_t\}$. See Figure 2. Once we have merged the first two components, we proceed similarly. For instance, we next pick some edge $\{v_s + a_t, v_{s+1} + a_t\}$ in our merged cycle; $\{v_s + 2a_t, v_{s+1} + 2a_t\}$ is an edge in $v_1 + 2a_t, v_2 + 2a_t, ..., v_{n/\gcd(n,a_1,...,a_{t-1})} + 2a_t, v_1 + 2a_t$. Deleting these two edges $\{v_s + a_t, v_{s+1} + a_t\}$ and $\{v_s + 2a_t, v_{s+1} + 2a_t\}$, and adding the edges $\{v_s + a_t, v_s + 2a_t\}$ and $\{v_{s+1} + a_t, v_{s+1} + 2a_t\}$ will have merged together the three cycles in

$$v_1 + ka_t, v_2 + ka_t, ..., v_{n/\gcd(n,a_1,...,a_{t-1})} + ka_t, v_1 + ka_t,$$

for $k = 0, 1, 2$. $\square$

3. Circulant TSP and Primes

That circulant TSP can be solved in polynomial-time whenever n is prime follows from Proposition 2.2 immediately: let ℓ denote the length of a cheapest edge in an input to circulant TSP (i.e. $c_\ell = \min\{c_1, c_2, ..., c_{\lfloor \frac{n}{2} \rfloor}\}$). Then by Proposition 2.2, the graph $C\langle\{\ell\}\rangle$ is Hamiltonian, and starting at vertex 1 and following edges of length ℓ until you return to vertex 1 yields a minimum-cost Hamiltonian cycle of cost $n \times c_\ell$. This result was first shown in the 70's in Garfinkel [10], but since then, no general results relating the complexity of circulant TSP to the factorization of n have been shown. More generally, the same logic implies that any circulant TSP instance can be easily solved if the cheapest edge-length ℓ is relatively prime to n, and an optimal solution will have cost $n \times c_\ell$.

Our first result extends these connections and shows that, when n is a prime-squared, it is also easy to determine the cost of circulant TSP solution. In this case, the cost of an optimal solution depends on up to two edge-lengths: the cheapest edge-length ℓ, and (if ℓ is not relatively prime to n), the cheapest edge-length that's relatively prime to n. Note that, if n is a prime-squared, the only edge-lengths that are not relatively prime to n are those that are multiples of p. I.e.

$$\gcd(i, n) = \begin{cases} p, & i \text{ is a multiple of } p \\ 1, & \text{else.} \end{cases}$$

Theorem 3.1. *Let $n = p^2$ where $p \geq 3$ is a prime. Let ℓ denote the length of a cheapest edge in an input to circulant TSP (i.e. $c_\ell = \min\{c_1, c_2, ..., c_{\lfloor \frac{n}{2} \rfloor}\}$), and let s denote the cheapest edge-length that's relatively prime to n (i.e. $c_s = \min\{S\}$, where $S = \{c_i : 1 \leq i \leq \lfloor \frac{n}{2} \rfloor, \gcd(n, i) = 1\}$ is the set of edge-lengths relatively prime to n). If $\gcd(\ell, n) = 1$, then a minimum-cost Hamiltonian cycle costs $n \times c_\ell$. Otherwise, a minimum-cost Hamiltonian cycle costs $(n - p) \times c_\ell + p \times c_s$.*

If $\gcd(\ell, n) = 1$, then the circulant graph $C\langle\{\ell\}\rangle$ is Hamiltonian. Otherwise, by Proposition 2.2, $C\langle\{\ell\}\rangle$ has $n/\gcd(n, p) = n/p = p$ components, each of which has p vertices (and all vertices in a component will be congruent mod p). We will adopt a convention for plotting $C\langle\{\ell, s\}\rangle$ in terms of these components, shown in Figure 3: we start with the first component of $C\langle\{\ell\}\rangle$, consisting of all vertices congruent to 1 mod p, connected by length-ℓ edges in a cycle that "wraps around" vertically. Then we effectively translate this component by s, plotting all vertices congruent to $1 + s$ mod p in the next column. We repeat this process forming a grid until we reach the component consisting of all vertices congruent to $(1 + (p-1)s) \mod p$ in the rightmost column. These vertices in the last column are then connected back to vertices in the first column (i.e. $1 + (p-1)s + s \equiv_p 1$, so a length-$s$ edge from a vertex in the last column wraps back around to a vertex in the first column). However, these length-s edges between the last and first column do not necessarily wrap around to the same row. See Figure 3, for instance, and see Gutekunst and Williamson [15] for more general results on the structure of circulant graphs.

Before proving Theorem 3.1, we need one basic fact about linear congruences. See, e.g., Theorem 57 of Hardy and Wright [16].

Proposition 3.2. *The linear congruence*

$$ax \equiv_n b$$

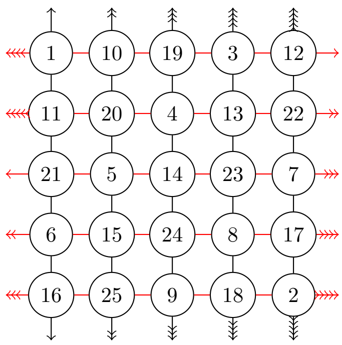

Figure 3: Convention for plotting $C\langle\{\ell, s\}\rangle$ when $n = 25$, $\ell = 10$, and $s = 9$. Length-ℓ edges are in black and length-s edges are in red. Edges on the border "wrap around" to the matching edge with the same number of arrowheads.

has a solution if and only if $\gcd(a, n)$ *divides* b*. Moreover, there are exactly* $\gcd(a, n)$ *solutions which take the form*

$$x_0 + \lambda \frac{n}{\gcd(a, n)},\ \lambda = 0, 1, ..., \gcd(a, n) - 1$$

for some $0 \leq x_0 < \frac{n}{\gcd(a,n)}$.

Proof (of Theorem 3.1). If $\gcd(\ell, n) = 1$, then the circulant graph $C\langle\{\ell\}\rangle$ is Hamiltonian, so following edges of length ℓ yields a Hamiltonian cycle of cost $n \times c_\ell$ (and since any Hamiltonian cycle must use n edges all of which cost at least c_ℓ, this is optimal).

Otherwise, any Hamiltonian cycle must cost at least $(n - p) \times c_\ell + p \times c_s$: Such a cycle must use at least p edges of cost at least c_s, as all edges cheaper than c_s stay within components of $C\langle\{\ell\}\rangle$ and at least p other edges are needed to connect these components in a cycle; the remaining $n - p$ edges must cost at least c_ℓ. Thus it suffices to show that a Hamiltonian cycle using $(n - p)$ length-ℓ edges and p length-s edges exists. Note that such a cycle will use exactly one length-s edge between each pair of adjacent columns (following the convention of Figure 3), and then one final length-s edge wrapping from the last column to the first. We construct a Hamiltonian path starting at vertex 1, using $(p - 1)$ edges of length ℓ to traverse that column, and then taking a length-s edge to the next column; we'll repeat this logic, entering a column, using $(p - 1)$ edges of length-ℓ to traverse that column, and then using a length-s edge to move to the next column. In each column, the only choice is whether we start by traversing "up" (i.e. from vertex 1 to $(1 - \ell) \mod n$) or "down" (i.e. from vertex 1 to $(1 + \ell) \mod n$). We need to show that we can choose a sequence of "up" and "down" columns so that the last vertex visited in the final column is $1 - s$.

Our intuition for doing so is shown in Figure 4, which first shows a sequence where we choose to go down in the first four columns and then up in the last three. Note that, each time we go down in a column, the final vertex visited in that column is one row above the first vertex in that column (modulo the number of rows); each time we go up, the final vertex visited in that column is one row below the first vertex in that column (modulo the number of rows). The red vertices trace out the last vertex visited in each column. On the right of Figure 4, we trace out (in red) all vertices we can end up at by any a sequence of ups and downs: there are two red vertices in the first column (based on whether we go up or down), then three red vertices in the second column (corresponding to going down twice, going down once and up once, or going up twice), and so on. By the penultimate column, we can choose a sequence of ups and downs to reach every vertex in that column.

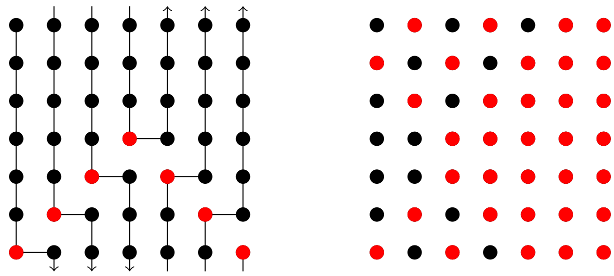

Figure 4: One possible sequence of up- and down-moves with the final vertex visited in each column marked in red (left). All possible final vertices marked in each column when $n = 7^2$ (right).

More formally, suppose that vertex $1 - s$ is in row r, with $0 \leq r < p$ (and the top row as row 0). We need to show that, regardless of r, we can choose a sequence of ups and downs to end our Hamiltonian path at the vertex in the row r of the last column (i.e., at $1 - s$). If we choose to go down k times and up $p - k$ times, we end in row $((-k) + (p - k)) \mod p$. It thus suffices to show that we can choose k, with $0 \leq k \leq p$, such that $p - 2k \equiv_p r$. That is, $2k \equiv_p p - r$. By Proposition , there is a unique solution $0 \leq k < p$. This value of k gives rise to a Hamiltonian path from 1 to $1 - s$ using exactly $p - 1$ edges of length-s; taking one final length-s edge from $1 - s$ to 1 yields the desired Hamiltonian cycle. □

Algorithmically, note that we can find the row r of $1 - s$ by solving

$$1 - s \equiv_n (p - 1) \times s + r \times \ell.$$

We can then find the desired value of k, both which can be done using the extended Euclidean algorithm in $O(\log^2(n))$ time. See, for example, Theorem 4.4 in [30].

4. The Two-Class TSP

In the $(1,2)-TSP$, all edges have cost 1 or 2. This well-studied problem is NP-hard and is a special case of the more general metric TSP, but better approximation algorithms are known for the $(1,2)-$TSP than for the metric TSP (see, e.g., Papadimitriou and Yannakakis [26], Berman and Karpinski [3], Karpinski and Schmied [18], and Adamaszek, Mnich, and Paluch [1]).

In this section, we consider the $(1,2)-$TSP specialized to circulant instances, the *two-class circulant TSP*: Here, the c_i can take on exactly two distinct values; without loss of generality, these values are 1 ("cheap") or 2 ("expensive"). That is, $c_i \in \{1,2\}$ for all $1 \leq i \leq \lfloor \frac{n}{2} \rfloor$.

Theorem 4.1. *Consider an instance of the two-class circulant TSP where $S := \{i : c_i = 1\}$ denotes the set of cheap edges, and let $g := \gcd(n, S)$ denote the GCD of n and all edge-lengths in S. Then the optimal solution to this instance has cost:*

$$\begin{cases} n, & g = 1 \\ n+g, & g > 1. \end{cases}$$

Proof. First, suppose that $g = 1$. Then by Proposition 2.2, the graph $C\langle S \rangle$ is Hamiltonian, so there is a Hamiltonian cycle just using the cheap edges in S.

Otherwise, $g > 1$, so $1 \notin S$ and the graph $C\langle S \rangle$ has g components. Any Hamiltonian cycle must use at least g expensive edges and thus costs at least $n+g$. To construct such a Hamiltonian cycle, we start by building a Hamiltonian path $v_1 = 1, v_2, ...v_{n/g}$ starting at vertex 1 on the component of $C\langle S \rangle$ including vertex 1; this can be done using the algorithm in Proposition 2.2 (obtaining a Hamiltonian cycle, say, and deleting an arbitrary edge incident to vertex 1). By Proposition 2.2, all vertices in this component are congruent to 1 mod g. We translate this Hamiltonian path to the the other components, obtaining Hamiltonian paths

$$v_1 + k, v_2 + k, ...v_{n/g} + k, k = 0, 1, ..., (g-1)$$

on each component. We join these paths with length-1 edges as in Figure 5, adding the edges $\{v_{n/g} + k, v_{n/g} + k+1\}$ and $\{k, k+1\}$ for k even. This yields a Hamiltonian path from vertex 1, and ending in the gth component of $C\langle S \rangle$ (where vertices are all congruent to 0 mod g).

This Hamiltonian path will either end at $v_1 + (g-1) = g$ (if g is even) or $v_{n/g} + (g-1)$ (if g is odd). Note, however, that both are adjacent to 1 by a cost-2 edge: otherwise, they would be in the same component of $C\langle S \rangle$. Thus, we can extend this Hamiltonian path to a Hamiltonian cycle, and we used exactly g cost-2 edges. □

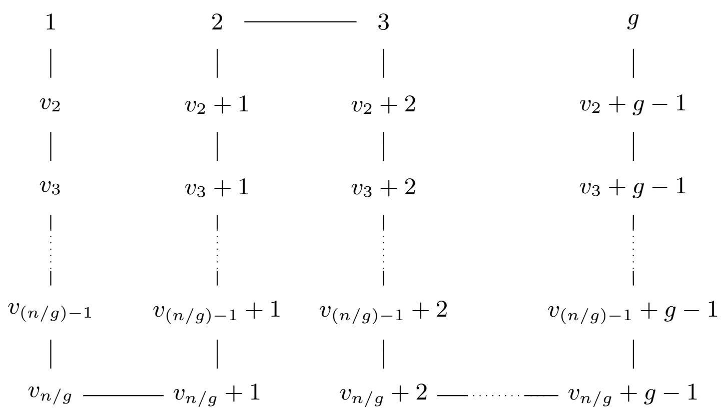

Figure 5: Extending a Hamiltonian path on one component of $C\langle S \rangle$ to a Hamiltonian path on all vertices in the proof of Theorem 4.1 when g is even. When g is odd, the Hamiltonian path ends at $v_{n/g} + g - 1$.

5. Minimum-Cost Eulerian Tours

In our final section, we consider one more problem related to the two-stripe circulant TSP based on a question posed by Jens Vygen: finding a minimum-cost Eulerian tour on a two-stripe circulant instance (or equivalently, finding a Hamiltonian tour on a the metric completion of a two-stripe circulant TSP instance). Gerace and Irving [12] give a (4/3)-approximation algorithm for general circulant TSP instances that are also metric (and therefore for finding minimum-cost Eulerian Tours on any circulant instance). Here, we show that that ratio can be improved to (10/9) when considering a two-stripe instance.

More specifically, consider a circulant instance with two finite edge costs $0 \leq c_i \leq c_j < \infty$. We assume that $\gcd(n, i, j) = 1$: otherwise, by Proposition 2.2, the graph $C\langle\{i,j\}\rangle$ will not be connected, and thus it will not admit an Eulerian tour. Similarly, let $g = \gcd(n, i)$. If $g = 1$, then the length-i edges form a Hamiltonian cycle, and form an optimal Eulerian tour of cost $n \times c_i$. Otherwise, $C\langle\{i\}\rangle$ consists of g components, and an Eulerian tour costs at least $(n-g) \times c_i + g \times c_j$, since only the length-$j$ edges can cross between components of $C\langle\{i\}\rangle$.

We now consider two extremal cases. First, if $c_i = c_j$, then any Hamiltonian cycle is optimal. Figure 6 sketches such a Hamiltonian cycle, based on the parity of g, and following our convention of plotting components of $C\langle\{i\}\rangle$ as columns, so that (cheap) length-i edges are vertical and (expensive) length-j edges are horizontal. Conversely, if $c_i = 0$, we can frivolously use length-i (vertical) edges. We consider an Eulerian tour as shown in Figure 7: There will be some vertex $(1-j) \mod n$ in the last column (highlighted in blue) connected to vertex 1 by a length-j edge. We follow a zig-zagging Hamiltonian path ending at either the bottom or top vertex in that column, which has n/g

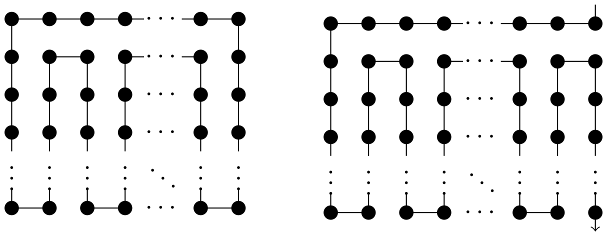

Figure 6: A feasible Hamiltonian cycles when g is even (left) and odd (right), which are optimal Eulerian tours when $c_i = c_j$.

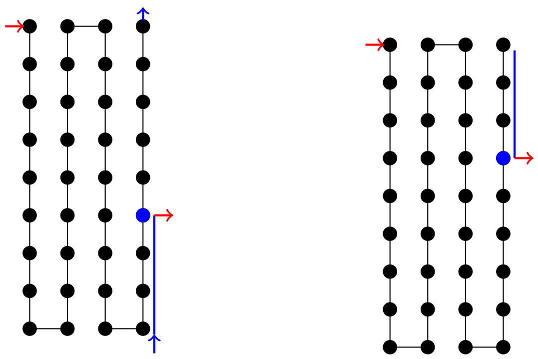

Figure 7: Optimal Eulerian tour when $c_i = 0$, depending on the row of the vertex $(1-j) \mod n$ within the last column.

vertices in it. We then take extra length-i edges to reach $(1-j) \mod n$: either "wrapping around" (as in the left of Figure 7) or turning within the same column (as in the right), depending on whichever uses fewer length-i edges. Since there are n/g length-i edges in this column, one direction will use at most $n/(2g)$ extra length-i edges; any tour will cost at least $g \times c_j$, and (when $c_i = 0$) these tours cost exactly $g \times c_j$

Our final result, which gives rise to a 10/9-approximation algorithm, is that at least one of these extremal tours will always be within 10/9 of the minimum-cost Eulerian tour. To make the analysis more clean, we scale all edge costs by $1/c_j$ (and if $c_j = 0$, then $c_i = c_j = 0$ and our tour from Figure 6 is optimal) and define $c := \frac{c_i}{c_j}$. Then our cheaper edges cost $0 \le c \le 1$, and our expensive edges cost 1. Thus our tours from Figure 6 cost

$$(n - 2(g-1))\, c + 2(g-1).$$

Our tours from Figure 7 use $(n/g) - 1$ length-i edges in each of the g columns in the original Hamiltonian path, at most $n/(2g)$ additional length-i edges in the last column, and exactly g length-j edges. All together, they thus cost

$$g\left(\frac{n}{g} - 1\right) c + \frac{n}{2g} c + g = \left(n - g + \frac{n}{2g}\right) c + g.$$

Theorem 5.1. *Consider a two-stripe circulant input where* $\gcd(n,i,j) = 1$ *and* $g := \gcd(n,i) > 1$. *At least one of the tours shown in Figures 6 and 7 has cost within (10/9) of the minimum-cost Eulerian tour.*

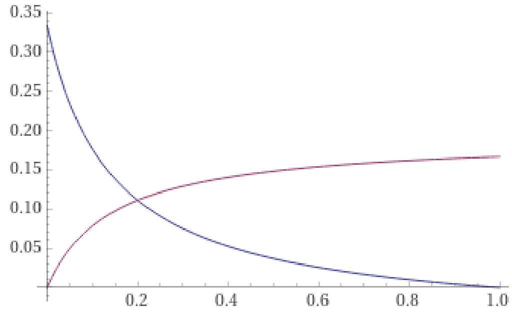

Figure 8: Example of $\frac{(g-2)(1-c)}{(n-g)c+g}$ (blue) and $\frac{\frac{n}{2g}c}{(n-g)c+g}$ (red) when $n = 24$ and $g = 3$.

Proof. First we note that any Eulerian tour must use at least n edges, and at least g of these must be length-j to fully connect the components of $C\langle\{i\}\rangle$ and return to vertex 1. Thus, any Eulerian tour costs at least $(n-g)c+g$. Note also that this implies that, if $g = 2$, the Eulerian tours from Figure 6 are optimal.

Hence, we assume that $g \ge 3$. We will benchmark each tour against the lowerbound, and thus want to show that

$$\min\{\frac{(n-2(g-1))\,c + 2(g-1)}{(n-g)c+g}, \frac{\left(n-g+\frac{n}{2g}\right)c+g}{(n-g)c+g}\} \le \frac{10}{9}.$$

Equivalently, that

$$\min\{1 + \frac{(g-2)(1-c)}{(n-g)c+g}, 1 + \frac{\frac{n}{2g}c}{(n-g)c+g}\} \le \frac{10}{9}.$$

or that

$$\min\{\frac{(g-2)(1-c)}{(n-g)c+g}, \frac{\frac{n}{2g}c}{(n-g)c+g}\} \le \frac{1}{9}.$$

Straightforward calculus shows that $\frac{(g-2)(1-c)}{(n-g)c+g}$ is decreasing in c for $c \in [0,1]$ (and decreases from $\frac{g-2}{g}$ to 0), while $\frac{\frac{n}{2g}c}{(n-g)c+g}$ is increasing in c for $c \in [0,1]$ (and increases from 0 to $1/(2g)$). Thus, $\min\{\frac{(g-2)(1-c)}{(n-g)c+g}, \frac{\frac{n}{2g}c}{(n-g)c+g}\}$ occurs when both terms are equal: when

$$(g-2)(1-c) = \frac{n}{2g}c.$$

See, for instance, Figure 8.

Solving for c, we find

$$c = \frac{2g(g-2)}{n + 2g(g-2)}.$$

Plugging in for c, we find that

$$\begin{aligned}\frac{\frac{n}{2g}c}{(n-g)c+g} &= \frac{\frac{n}{2g}\frac{2g(g-2)}{n+2g(g-2)}}{(n-g)\frac{2g(g-2)}{n+2g(g-2)}+g}\\ &= \frac{\frac{n(g-2)}{n+2g(g-2)}}{(n-g)\frac{2g(g-2)}{n+2g(g-2)}+g}\\ &= \frac{n(g-2)}{(n-g)2g(g-2)+g(n+2g(g-2))}\\ &= \frac{n(g-2)}{2ng(g-2)-2g^2(g-2)+gn+2g^2(g-2)}\\ &= \frac{g-2}{2g^2-3g}.\end{aligned}$$

Hence,

$$\min\{\frac{(g-2)(1-c)}{(n-g)c+g}, \frac{\frac{n}{2g}c}{(n-g)c+g}\} \leq \frac{g-2}{2g^2-3g}.$$

Finally, we note that $\frac{g-2}{2g^2-3g}$ is decreasing in g for $g \geq 3$, and at $g=3$, $\frac{g-2}{2g^2-3g} = 1/9$. This completes our proof. □

Acknowledgments

The material in this poster is based in work supported by the National Science Foundation, Grant No. CCF-2153331.

References

[1] A. Adamaszek, M. Mnich, and K. Paluch. New approximation algorithms for (1, 2)-tsp. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[2] S. Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.

[3] P. Berman and M. Karpinski. 8/7-approximation algorithm for (1, 2)-TSP. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*, pages 641–648, 2006.

[4] M. Buratti and F. Merola. Dihedral Hamiltonian cycle systems of the cocktail party graph. *Journal of Combinatorial Designs*, 21(1):1–23, 2013.

[5] R. Burkard and W. Sandholzer. Efficiently solvable special cases of bottleneck travelling salesman problems. *Discrete Applied Mathematics*, 32(1):61–76, 1991.

[6] R. E. Burkard. Efficiently solvable special cases of hard combinatorial optimization problems. *Mathematical Programming*, 79(1):55–69, 1997.

[7] R. E. Burkard, V. G. Deĭneko, R. van Dal, J. A. A. van der Veen, and G. J. Woeginger. Well-solvable special cases of the traveling salesman problem: A survey. *SIAM Review*, 40(3):496–546, 1998.

[8] N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA, 1976.

[9] S. Costa, F. Morini, A. Pasotti, and M. A. Pellegrini. A problem on partial sums in Abelian groups. *Discrete Mathematics*, 341(3):705–712, 2018.

[10] R. S. Garfinkel. Minimizing wallpaper waste, part 1: A class of traveling salesman problems. *Operations Research*, 25(5):741–751, 1977.

[11] I. Gerace and F. Greco. The travelling salesman problem in symmetric circulant matrices with two stripes. *Mathematical Structures in Computer Science*, 18(1):165–175, 2008.

[12] I. Gerace and R. Irving. The traveling salesman problem in circulant graphs. Technical report, TR-1998-15, University of Glasgow, Department of Computing Science, 1998.

[13] F. Greco and I. Gerace. The traveling salesman problem in circulant weighted graphs with two stripes. *Electronic Notes in Theoretical Computer Science*, 169:99–109, 2007.

[14] S. C. Gutekunst, B. Jin, and D. P. Williamson. The two-stripe symmetric circulant tsp is in p. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 319–332. Springer, 2022.

[15] S. C. Gutekunst and D. P. Williamson. Characterizing the integrality gap of the subtour LP for the circulant traveling salesman problem. *SIAM Journal on Discrete Mathematics*, 33(4):2452–2478, 2019.

[16] G. Hardy and E. Wright. *An Introduction to the Theory of Numbers*. Oxford University Press, 6th edition, 2008.

[17] A. R. Karlin, N. Klein, and S. O. Gharan. A (slightly) improved approximation algorithm for metric tsp. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 32–45, 2021.

[18] M. Karpinski and R. Schmied. On approximation lower bounds for TSP with bounded metrics. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:8, 2012.

[19] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. John Wiley and Sons, 1985.

[20] B. D. McKay and T. Peters. Paths through equally spaced points on a circle. *arXiv preprint arXiv:2205.06004*, 2022.

[21] E. Medova. Using QAP bounds for the circulant TSP to design reconfigurable networks. In *Quadratic Assignment and Related Problems, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, May 20-21, 1993*, pages 275–292, 1993.

[22] 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*, 28(4):1298–1309, 1999.

[23] T. Mömke and O. Svensson. Removing and adding edges for the traveling salesman problem. *Journal of the ACM*, 63(1):2:1–2:28, 2016.

[24] M. Mucha. $\frac{13}{9}$-approximation for graphic TSP. *Theory of Computing Systems*, 55(4):640–657, 2014.

[25] S. Oveis Gharan, A. Saberi, and M. Singh. A randomized rounding approach to the traveling salesman problem. In *Proceedings of the 52nd Annual IEEE Symposium on the Foundations of Computer Science*, pages 550–559, 2011.

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

[27] A. Pasotti and M. A. Pellegrini. A new result on the problem of Buratti, Horak and Rosa. *Discrete Mathematics*, 319:1 – 14, 2014.

[28] A. Sebő and J. 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.

[29] A. Serdyukov. On some extremal walks in graphs. *Upravlyaemye Sistemy*, 17:76–79, 1978.

[30] V. Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, 2009.

[31] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, 2011.