

Structural Properties of Hard Metric TSP Inputs^{*} (Extended Abstract)

Tobias Mömke

KTH — Royal Institute of Technology, Stockholm, Sweden^{**}
moemke@kth.se

Abstract. The metric traveling salesman problem is one of the most prominent APX-complete optimization problems. An important particularity of this problem is that there is a large gap between the known upper bound and lower bound on the approximability (assuming $P \neq NP$). In fact, despite more than 30 years of research, no one could find a better approximation algorithm than the 1.5-approximation provided by Christofides. The situation is similar for a related problem, the metric Hamiltonian path problem, where the start and the end of the path are prespecified: the best approximation ratio up to date is $5/3$ by an algorithm presented by Hoogeveen almost 20 years ago.

In this paper, we provide a tight analysis of the combined outcome of both algorithms. This analysis reveals that the sets of the hardest input instances of both problems are disjoint in the sense that any input is guaranteed to allow at least one of the two algorithms to achieve a significantly improved approximation ratio. In particular, we show that any input instance that leads to a $5/3$ -approximation with Hoogeveen's algorithm enables us to find an optimal solution for the traveling salesman problem. This way, we determine a set S of possible pairs of approximation ratios. Furthermore, for any input we can identify one pair of approximation ratios within S that forms an upper bound on the achieved approximation ratios.

1 Introduction

While being one of the hardest problems with respect to approximability in its general formulation [21], the metric traveling salesman problem (Δ TSP) is well known to be APX-complete. Unless $P = NP$, it does not permit an approximation ratio that is better than $220/219$ [19]. The best algorithm available is a 1.5-approximation algorithm by Christofides [10]. The situation is very similar for the metric Hamiltonian path problem with prespecified start and end vertex (Δ HPP₂): the lower bound is closely related to that of the Δ TSP and the

^{*} The research was partially funded by SNF grant 200021-109252/1 and ERC Advanced investigator grant 226203.

^{**} The results of this paper were obtained at ETH Zurich, Switzerland.

5/3-approximative algorithm by Hoogeveen [16] was not improved so far. An alternative proof for the same result was given by Guttmann-Beck et al. [15]. For both problems, the upper bounds on the approximability have resisted all attempts of improvement for many years.

The two problems are closely related, since both of them take a complete metric graph as input and the goal of both problems is to visit each of the vertices. The Δ TSP is basically the Δ HPP₂, where the start vertex and the end vertex are the same.

In this paper, we significantly improve our former result from [8]. We characterize hard input instances for both Christofides' and Hoogeveen's algorithm and show that the sets of worst-case instances for these algorithms are disjoint in the sense that a hard instance for one problem allows a significantly improved approximation for the other one. We determine the set S of all possible pairs of approximation ratios that are achieved this way (depicted in Figure 1). This includes the guarantee that a worst-case input for Hoogeveen's algorithm, for which we can only compute a 5/3-approximative solution, enables us to compute an optimal solution for Δ TSP on that input. We are guaranteed that the cost of this optimal solution is exactly 4/3 times as high as that of an optimal solution for Δ HPP₂. In addition to the results on the structure of hard inputs, we show that for each input, we can determine a pair of approximation ratios from S that forms an upper bound on the achieved ratios.

To show that our analysis is tight, we present a class of hard input instances for each of the possible pairs from S that forms an upper bound.

Our detailed analysis of these algorithms provides deep insight of the core of the hardness involved in classes of input instances for which we cannot provide an improved approximation. We show for instance that in any worst case instance for Hoogeveen's algorithm, the minimum spanning tree involved in the algorithm contains a path between the end vertices of cost exactly 1/3 of the cost of an optimal solution and a gradual relaxation of this bound for inputs that do not cause worst-case behavior. This generalizes some of the results from [15]. The properties revealed in this work restrict the types of inputs that a possible improved algorithm for the Δ HPP₂ has to cope with. This might be helpful for creating improved algorithms for the Δ HPP₂ or the Δ TSP.

1.1 Related Known Results

The result of this paper is a win/win strategy for approximation algorithms. The concept of win/win strategies is to specify a parameter of the input instance and to guarantee — for any value of the parameter — that we can compute an improved solution for one of two problems according to some complexity measure. Here, the parameter is the computed bound on the approximation ratio for the Hamiltonian path problem and the complexity measure is the approximation ratio.

Win/win strategies fit well into the framework of parameterized complexity [11, 18] as well as stability of approximation [5, 14, 7], because all of these approaches are based on studying the “hardness” of their problem instances.

In parameterized algorithms, win/win strategies are a tool used for kernelization [1], which is a technique used in order to reduce the size of the problem instance and the parameter. Prieto and Sloper presented such a kernelization of the k -internal spanning tree problem by using a win/win strategy that relates the k -internal spanning tree problem and the vertex cover problem [20]. An overview on the use of win/win strategies in parameterized algorithms can be found in [13].

The concept of win/win strategies relates to the design of hybrid algorithms as proposed by Vassilevska et al. [22]. They presented algorithms that allow either an improved approximation ratio or an improved (but still exponential) runtime for computing exact solutions.

Win/win strategies for approximation were independently introduced in our paper [8] and by Eppstein [12]. Eppstein uses the name *paired approximation* for this concept. He was able to use win/win as an upper bound technique and he showed for some pairs of problems that they do not have such a relation.

The win/win result from [8] is related to the result of this paper, but the achieved pairs of approximation ratios of this paper are significantly improved (see Figure 1). The result from [8] is existential in the sense that it does not provide the possibility to identify pairs of approximation ratios within the given boundaries.

Our results open an interesting connection to another field of algorithmics called *reoptimization*. In reoptimization, one is given an optimal or almost optimal solution for some input instance. Now the problem is to find a solution for a different input instance that is closely related to the given one. Some approximation results on reoptimization can be found in [2–4, 6]. In [9], one can find an overview on reoptimization. Let us consider the reoptimization problem of ΔHPP_2 , where the modification is to change one of the end vertices. If there is an approximation algorithm for this reoptimization problem that is better than $5/3$ -approximative, then we can use this algorithm for ΔHPP_2 : for a given worst-case instance I of Hoogeveen’s algorithm, we determine an optimal ΔTSP solution and use this as input for the reoptimization problem by declaring the start vertex of I to be the start vertex as well as the end vertex. Then I is the modified instance that is to be solved by the reoptimization algorithm. However, to improve the approximation ratio for ΔHPP_2 by a constant factor, we depend on the ability of the reoptimization algorithm to handle a broader range of input instances: instead of requiring an optimal solution for the given input graph, it has to be able to accept solutions that deviate by a (small) constant factor from an optimal solution.

1.2 Organization of the Paper

Section 2 fixes the notation used in the paper. The core of this paper is located in Section 3, where we show the combined upper bounds on the approximation ratios achieved by Christofides’ and Hoogeveen’s algorithm. Section 4 then provides a more detailed analysis of hard instances. Finally, Section 5 shows that the analyses of all upper bounds shown in this paper are tight.

2 Preliminaries

For graphs, we use a notation similar to [23]. In a graph $G = (V, E)$, the edges are sets of two vertices. A *trail* from u to v is a sequence of adjacent edges leading from u to v , where no edge may be used more than once. A trail is uniquely defined by a list of vertices $uw_1w_2 \dots w_iv$, where consecutive vertices describe the edges of the trail. We say that $w_1 \dots w_i$ are the *inner vertices*. The *length of a trail* is the number of its edges. A trail, where each vertex is used at most once, is a *path*. A closed trail, i. e., a trail that starts and ends with the same vertex, is a *circuit*. A circuit, where each inner vertex is visited only once, is a *cycle*. In a graph $G = (V, E)$, a *Hamiltonian path* from u to v is a path of length $|V| - 1$ from u to v and a *Hamiltonian tour* is a cycle of length $|V|$. Let $[n]$ denote the set $\{1, 2, \dots, n\}$, where n is an integer.

We call a complete graph $G = (V, E)$ with cost function $c : E \rightarrow \mathbb{Q}^+$ *metric*, if the edge costs satisfy the triangle inequality $c(\{u, v\}) \leq c(\{u, w\}) + c(\{w, v\})$ for any pairwise distinct vertices $u, v, w \in V$.

The *metric traveling salesman problem*, ΔTSP , is the problem of finding a minimum-cost Hamiltonian tour in a complete metric graph. The *metric minimum-cost Hamiltonian path problem* in complete graphs, where the two end vertices are fixed, is called ΔHPP_2 .

Given a graph $G = (V, E)$ and two vertices u and v in G , then we define $G + \{u, v\}$ as $(V, E \cup \{\{u, v\}\})$. In a graph, a vertex is *odd* or *even*, if its degree is odd or even.

3 A Win/Win Strategy for ΔTSP and ΔHPP_2

In this section, we provide an improved analysis of a simple algorithm that combines the two well-known algorithms from [10] and [16]. The algorithm is exactly that from [8]. For completeness, we state this algorithm here.

Algorithm 1 Path and Cycle [8]

Input: A complete graph $G = (V, E)$, a metric cost function $c : E \rightarrow \mathbb{Q}^+$, and two vertices s and t .

- 1: Compute a minimum spanning tree T in G .
- 2: Compute a minimum perfect matching M_C on the odd vertices of T in G .
- 3: Compute a minimum perfect matching M_P on the odd vertices of the multigraph $T + \{s, t\}$ in G .
- 4: Compute an Eulerian tour Eul_C in the multigraph $T \cup M_C$ and an Eulerian path Eul_P in the multigraph $T \cup M_P$.
- 5: Shorten Eul_C and Eul_P to a Hamiltonian tour H_C and a Hamiltonian path H_P , respectively.

Output: H_C and H_P .

We first bound the costs of the matchings involved in the algorithm. Let Opt_P and Opt_C be optimal solutions for the ΔHPP_2 and the ΔTSP , respectively.

Lemma 1.

$$c(M_P) + c(M_C) \leq \min\{c(\text{Opt}_P), c(\text{Opt}_C)\}$$

Proof. First we show that $c(M_P) + c(M_C) \leq c(\text{Opt}_P)$ holds. Let P be an optimal Hamiltonian path from s to t in G and let v_1, v_2, \dots, v_k be the odd vertices of T . Let us assume without loss of generality that they are in the order as they appear in Opt_P . It is clear that k is even. Then we define the matching M'_C as the set of edges $\{v_i, v_{i+1}\}$, where i is odd. Analogously, M'_P is the matching containing the edges $\{v_j, v_{j+1}\}$, where j is even. Additionally, M'_P contains $\{s, v_1\}$ if $v_1 \neq s$ and $\{v_k, t\}$ if $v_k \neq t$. Observe that M'_C is a perfect matching on the odd vertices of T and M'_P is a perfect matching on the odd vertices of $T + \{s, t\}$. Since M'_C and M'_P are disjoint, due to the triangle inequality $c(M'_P) + c(M'_C) \leq c(\text{Opt}_P)$. Since M_P and M_C are minimal, $c(M_P) \leq c(M'_P)$ and $c(M_C) \leq c(M'_C)$.

Now we show that $c(M_P) + c(M_C) \leq c(\text{Opt}_C)$. Note that M_C is a minimum-cost perfect matching of v_1, v_2, \dots, v_k . By Christofides' analysis, we have $c(M_C) \leq c(\text{Opt}_C)/2$: due to the triangle inequality, the cycle formed by v_1, v_2, \dots, v_k in the order as these vertices appear in an optimal Hamiltonian tour Opt_C is not more expensive than Opt_C itself. Since this cycle has two disjoint perfect matchings, the cheaper one has a cost of at most half of the cycle's cost. An analogous analysis shows

$$c(M_P) \leq c(\text{Opt}_C)/2. \quad (1)$$

The only difference is the set of vertices that forms the minimum cost perfect matching which is composed of the odd vertices from $T + \{s, t\}$. \square

Let $\alpha := c(H_P)/c(\text{Opt}_P)$ be the approximation ratio for the computed Hamiltonian path and let $\beta := c(H_C)/c(\text{Opt}_C)$ be the approximation ratio for the computed Hamiltonian tour for a given input G, c, s, t , where H_P and H_C are the solutions of Algorithm 1. Furthermore, we determine a value p from the costs of intermediate graphs in Algorithm 1 as

$$p := \max\{c(T), c(M_P) + c(M_C), 1.5c(M_P)\}.$$

We will show in the following, that p forms a lower bound on the cost of an optimal solution for ΔHPP_2 .

Let α' be the value such that $c(H_P) = \alpha'p$. Thus, we can determine the value of α' , whereas we do not know the value α . We will use α' as a parameter that determines a guarantee for the achieved approximation ratios for ΔTSP and ΔHPP_2 .

Lemma 2. *For any input of Algorithm 1,*

$$(2\alpha - 2)c(\text{Opt}_P) \leq (2\alpha' - 2)p \leq c(\text{Opt}_C) \leq (3 - \alpha')p \leq (3 - \alpha)c(\text{Opt}_P).$$

Proof. We first show that $p \leq c(\text{Opt}_P)$ holds. Since Opt_P is a spanning tree, $c(T) \leq c(\text{Opt}_P)$. Due to Lemma 1, also $c(M_P) + c(M_C) \leq c(\text{Opt}_P)$. Due to the analysis of Hoogeveen's algorithm in [16], $c(M_P) \leq 2c(\text{Opt}_P)/3$ and thus

$3c(M_P)/2 \leq c(\text{Opt}_P)$. Therefore, also the maximum of the three values is at most $c(\text{Opt}_P)$.

Since p is a lower bound on $c(\text{Opt}_P)$,

$$(2\alpha - 2)c(\text{Opt}_P) = 2c(H_P) - 2c(\text{Opt}_P) \leq 2c(H_P) - 2p = (2\alpha' - 2)p,$$

which shows the first inequality of the lemma.

We continue the proof by showing $(2\alpha' - 2)p \leq c(\text{Opt}_C)$. Since $c(H_P) = \alpha'p \leq c(T) + c(M_P)$,

$$(2\alpha' - 2)p \leq 2(c(T) + c(M_P)) - 2p \leq 2(c(T) + c(M_P)) - 2c(T) = 2c(M_P).$$

Now the second inequality follows because of (1).

For the third inequality $c(\text{Opt}_C) \leq (3 - \alpha')p$, we note that $c(H_C) \leq c(T) + c(M_C)$. Due to the definition of p , $c(M_C) + c(M_P) \leq p$ and $c(T) \leq p$. Therefore,

$$c(T) + c(M_C) \leq 2p - c(M_P).$$

Since $c(M_P) \geq c(H_P) - c(T)$ holds,

$$c(M_P) \geq \alpha'p - c(T) = \left(\alpha' - \frac{c(T)}{p} \right) p \geq (\alpha' - 1)p.$$

Therefore we get

$$c(H_C) \leq 2p - (\alpha' - 1)p = (3 - \alpha')p. \quad (2)$$

The last inequality follows, since

$$(3 - \alpha')p = 3p - \alpha'p = 3p - c(H_P) \leq 3c(\text{Opt}_P) - \alpha c(\text{Opt}_P) = (3 - \alpha)c(\text{Opt}_P).$$

□

As a first consequence of Lemma 2, we can relate the actual approximation ratio α that A achieved for the given input for ΔHPP_2 to that for ΔTSP , β .

Theorem 1. *For any input of A ,*

$$\beta \leq \min \left\{ 1.5, \frac{1}{\alpha - 1} - \frac{1}{2} \right\} \quad \text{and} \quad \alpha \leq \min \left\{ \frac{5}{3}, \frac{1}{\beta + 1/2} + 1 \right\}.$$

Proof. Due to the analysis of Christofides and Hoogeveen, we can bound β and α from above by 1.5 and 5/3. For the second bound of β we note that, due to Lemma 2 and (2), $c(H_C) \leq (3 - \alpha)c(\text{Opt}_P)$. Since Lemma 2 also states that $c(\text{Opt}_C) \geq (2\alpha - 2)c(\text{Opt}_P)$, we get

$$\beta = \frac{c(H_C)}{c(\text{Opt}_C)} \leq \frac{(3 - \alpha)c(\text{Opt}_P)}{(2\alpha - 2)c(\text{Opt}_P)} = \frac{3 - \alpha}{2\alpha - 2} = \frac{2 - (\alpha - 1)}{2(\alpha - 1)} = \frac{1}{\alpha - 1} - \frac{1}{2}. \quad (3)$$

The remaining statement of the theorem now follows immediately, since $\beta \leq 1/(\alpha - 1) - 1/2$ implies $\alpha \leq 1/(\beta + 1/2) + 1$. □

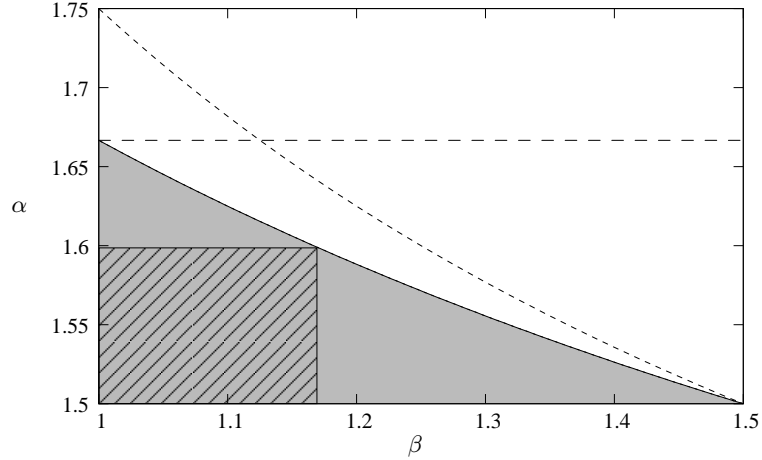


Fig. 1. The gray area describes the set S of all combinations of the approximation ratios α and β for the ΔHPP_2 and the ΔTSP achieved by Algorithm 1. The solid line describes the upper bound on the approximation ratios achieved by that algorithm. The dashed lines represent the upper bounds on the approximation ratios proven in [8]. The hatched area contains the set of possible pairs of solutions given that α' coincides with the highest value of α within the area.

Figure 1 gives a graphical representation of the theorem. The set S of all valid combinations of approximation ratios achieved by Algorithm 1 is represented as the gray area.

The following corollary follows from Theorem 1 by setting $\alpha = 5/3$.

Corollary 1. *Any worst-case instance for Hoogeveen's algorithm for ΔHPP_2 allows us to compute an optimal Hamiltonian cycle in G .*

Theorem 1 described properties that belong to the core of the relation between the two problems. Now we will change the focus and describe how to use the revealed relations in order to guarantee improved approximations according to parameters that we can measure, namely α' . More precisely, we determine the approximation ratios according to the spanning tree T and the matchings M_P and M_C . Let $\delta = p - c(T)$. Note that $\delta \geq 0$ and $\alpha \leq \alpha'$ holds.

Theorem 2. *For any input of A ,*

$$\beta \leq \min\{1.5, 1/(\alpha' - 1) - 1/2\} - \delta/c(\text{Opt}_C).$$

Proof. Analogous to the proof of Theorem 1, $c(H_C) \leq c(\text{Opt}_C) \cdot (1/(\alpha' - 1) - 1/2)$ and $c(H_C) \leq c(\text{Opt}_C) \cdot 1.5$. In this analysis, however, we estimated $c(T)$ by $c(\text{Opt}_P)$. The cost of the actual solution H_C is at least δ cheaper than we estimated previously. Therefore, the claim of the theorem follows. \square

The effect of Theorem 2 is depicted in the hatched area in Figure 1.

4 Implications of the Win/Win strategy

In this section, we classify hard input instances for the ΔHPP_2 . To this end, similar to [15] we combine Algorithm 1 and a variant of the well-known tree-doubling algorithm for the ΔTSP , namely Algorithm 2, which enables us to restrict the class of hard input instances.

Algorithm 2 Tree Doubling

Input: A complete graph $G = (V, E)$, a metric cost function $c : E \rightarrow \mathbb{Q}^+$, and two vertices s and t .

- 1: Compute a minimum spanning tree T in G .
- 2: Let P_{st} be the unique path in T that connects s and t .
- 3: Find an Eulerian tour Eul_P in the multi-graph $T + (T - P_{st})$.
- 4: Shorten Eul_P to a Hamiltonian path H_P .

Output: H_P .

In particular, we focus on the distance of s and t in G . Let A be the algorithm that runs both Algorithm 1 and Algorithm 2. The output of A is the cycle H_C from Algorithm 1 and the path H_P that is the smaller one of the two computed Hamiltonian paths. We introduce $\tilde{\alpha}$ and $\tilde{\alpha}'$ similar to α and α' , but with a slightly extended meaning: these values are based on A instead of Algorithm 1. For simplicity, we assume both algorithms involved in A to use the same spanning tree T .

Theorem 3. *For any input of A ,*

$$\begin{aligned} (2\tilde{\alpha} - 3)c(\text{Opt}_P) &\leq (2\tilde{\alpha}' - 2)p - c(\text{Opt}_P) \leq c(\{s, t\}) \leq c(P_{st}) \\ &\leq (2 - \tilde{\alpha}')p \leq (2 - \tilde{\alpha})c(\text{Opt}_P). \end{aligned}$$

Proof. The first and the last inequality hold, similar to Lemma 2, since $c(H_P) = \tilde{\alpha}c(\text{Opt}_P) = \tilde{\alpha}'p$ and $p \leq c(\text{Opt}_P)$.

For the second inequality, note that any Hamiltonian path from s to t can be made a Hamiltonian cycle by adding the edge $\{s, t\}$. Therefore, $c(\text{Opt}_C) \leq c(\text{Opt}_P) + c(\{s, t\})$ and thus, applying Lemma 2, $(2\tilde{\alpha}' - 2)p \leq c(\text{Opt}_P) + c(\{s, t\})$, which implies the second inequality.

The third inequality $c(\{s, t\}) \leq c(P_{st})$ trivially holds due to the triangle inequality.

For the fourth inequality, we first have to analyze Algorithm 2. The algorithm is correct because in $T + (T - P_{st})$, all vertices but s and t have an even degree, which ensures the existence of Eul_P . Due to the triangle inequality, $c(H_P) \leq c(\text{Eul}_P)$. Therefore, $c(H_P) \leq c(T + (T - P_{st}))$.

By the definitions of $\tilde{\alpha}'$ and p , we have $\tilde{\alpha}'p \leq 2c(T) - c(P_{st})$ and thus the fifth inequality follows:

$$c(P_{st}) \leq 2c(T) - \tilde{\alpha}'p \leq 2p - \tilde{\alpha}'p.$$

□

Theorem 3 reveals several properties of hard input instances. For instance, by setting $\tilde{\alpha}$ to $5/3$ in Theorem 3, we conclude that in each worst-case instance for ΔHPP_2 , $c(P_{st}) = c(\{s, t\}) = c(\text{Opt}_P)/3$ holds. This means that according to Theorem 1, adding the edge $\{s, t\}$ to an optimal Hamiltonian path from s to t yields an optimal Hamiltonian tour in the same graph.

Furthermore, since P_{st} is a part of T , we can take into account the number of edges in P_{st} . Let γ be the value such that $c(P_{st}) = \gamma c(\text{Opt}_P)$. In other words γ is the fraction of $c(\text{Opt}_P)$ that is formed by P_{st} .

Theorem 4. *Suppose that there are k or fewer edges in P_{st} . Then there exists an algorithm that achieves an approximation ratio of*

$$(3 - \tilde{\alpha}) \left(\frac{1}{\tilde{\alpha} - 1} - \frac{1}{2} \right) + \left(1 - \frac{2}{k} \right) \gamma$$

for ΔHPP_2 .

Proof. Let $e = \{u, v\}$ be the edge of maximal cost in P_{st} such that the four vertices are in the order s, u, v, t within P_{st} . Given the Eulerian cycle Eul_C from Algorithm 1, we remove e from Eul_C and add the two edges $\{s, u\}$ and $\{v, t\}$. The resulting graph has an Eulerian path from s to t . Let H'_P be that tour shortened to a Hamiltonian path. Then the cost of H'_P is at most

$$c(T) + c(M_C) - c(e) + (c(P_{st}) - c(e)).$$

Since there are at most k edges in P_{st} , $c(e) \geq c(P_{st})/k$. The value of β is based on the cost of H_C , which is $c(T) + c(M_C)$. Therefore, Theorem 1 implicitly states that $c(T) + c(M_C) \leq (1/(\tilde{\alpha} - 1) - 1/2)c(\text{Opt}_C)$ holds and we can bound the cost of H'_P from above by

$$\begin{aligned} c(H'_P) &\leq \left(\frac{1}{\tilde{\alpha} - 1} - 1/2 \right) c(\text{Opt}_C) + \left(1 - \frac{2}{k} \right) c(P_{st}) \\ &\leq \left(\frac{1}{\tilde{\alpha} - 1} - 1/2 \right) (3 - \tilde{\alpha})c(\text{Opt}_P) + \left(1 - \frac{2}{k} \right) \gamma c(\text{Opt}_P). \end{aligned}$$

The last inequality holds because of Lemma 2. Dividing this value by $c(\text{Opt}_P)$ yields the claimed approximation ratio. \square

Note that in the special case that $\tilde{\alpha} = 5/3$, Theorem 4 together with Theorem 3 implies that the cost of the computed Hamiltonian path is at most

$$\left(\frac{5}{3} - \frac{2}{3k} \right) \cdot c(\text{Opt}_P).$$

5 Combined Hard Input Instances

In this section, we show that the analysis of Theorem 1 is tight. To this end, we construct a class of graphs that can be adapted to any choice of $1.5 < \tilde{\alpha} < 5/3$,

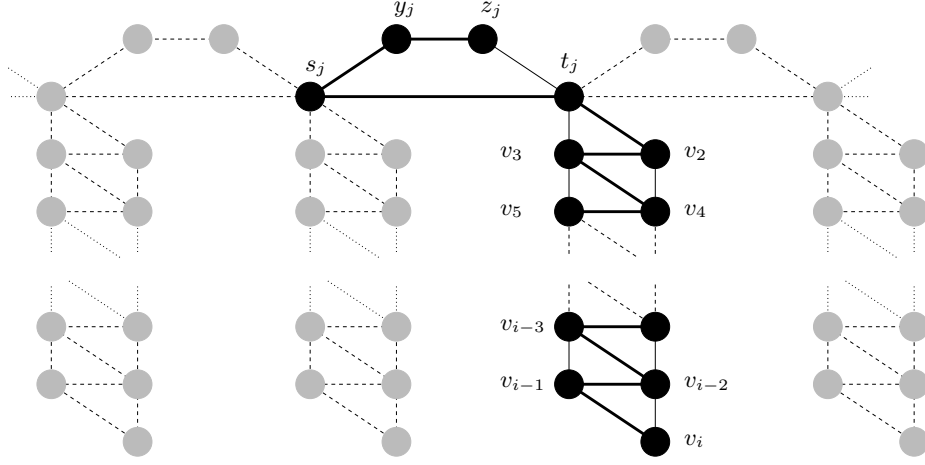


Fig. 2. The graph $G_{\hat{\alpha}, i, k}$. The bold lines form a part of the spanning tree T .

where $\hat{\alpha}$ is the aimed-for lower bound on the approximation ratio achieved for ΔHPP_2 . We define $\hat{\beta} := \max\{1.5, 1/(\hat{\alpha} - 1) - 1/2\}$, the guaranteed upper bound for ΔTSP according to Theorem 1. Hence, we aim for a lower bound $\hat{\beta}$ on the achieved approximation ratio for ΔTSP .

The basic building blocks of the graphs are the well known hard input instances for Christofides' algorithm from [10] (also described, e. g., in the textbook [17]) and for Hoogeveen's algorithm from [16].

Let $i \geq 4$ be an even number. Then we construct the graph $H_{i, \rho} = (V_i, E_{i, \rho})$, where $V_i = \{v_1, v_2, \dots, v_i\}$ and ρ is a value that depends on the specific pair of bounds that we aim for. We specify $E_{i, \rho}$ by determining the edges of cost ρ/i . All remaining edges have the cost of the shortest path between the corresponding vertices. We say that two vertices are connected, if they are connected by an edge of cost ρ/i . For any $j \in [i - 1]$, v_j and v_{j+1} are connected. Furthermore, for any $j \in [i - 2]$, v_j and v_{j+2} are connected.

Now, for $k \in \mathbb{N}$, we construct a graph $G_{\hat{\alpha}, i, k}$ with $n = 1 + (i + 2)k$ vertices (k copies of $H_{i, \rho}$ and $2k + 1$ additional vertices) as depicted in Figure 2.

For each $j = 1, 2, \dots, k$, we create a cycle $s_j y_j z_j t_j s_j$ such that each edge of the cycle has cost 1. The remaining two edges between these vertices are of cost 2. To each vertex t_j , we attach a copy of $H_{i, \rho}$ with $\rho = \frac{5-3\hat{\alpha}}{\hat{\alpha}-3/2}$ such that $t_j = v_1$. Now we join all k components such that for $j \in [k - 1]$, $t_j = s_{j+1}$. Again, all remaining edges of the resulting graph cost as much as the shortest path between the corresponding vertices. The end vertices are $s = s_1$ and $t = t_k$.

Theorem 5. *For each $3/2 < \hat{\alpha} < 5/3$ and each $\varepsilon > 0$, there are integers i and k such the combined result of Algorithm 1, Algorithm 2, and Theorem 4 is not $(\hat{\alpha} - \varepsilon)$ -approximative for ΔHPP_2 and not $(\hat{\beta} - \varepsilon)$ -approximative for ΔTSP for the input $G_{\hat{\alpha}, i, k}$.*

6 Conclusion

We have shown a tight bound of the combined approximation ratio of Hoogeveen's algorithm and Christofides' algorithm and for any input, we provided a pair of approximation ratios that is guaranteed to be achieved. We revealed a strong relation between the two problems and characterized properties of hard instances. These properties might be helpful in order to find an improved algorithm for ΔHPP_2 or ΔTSP . Since the described properties of hard input instances are very specific, the results of this paper show that for most of the practical input instances, we can guarantee better approximation ratios than in the worst-case.

Acknowledgement

The author would like to thank Hans-Joachim Böckenhauer for helping to improve the presentation.

References

1. Abu-Khzam, F.N., Collins, R.L., Fellows, M.R., Langston, M.A., Suters, W.H., Symons, C.T.: Kernelization algorithms for the vertex cover problem: Theory and experiments. In: Arge, L., Italiano, G.F., Sedgewick, R. (eds.) Proc. of the 6th Workshop on Algorithmic Engineering and Experiments (ALENEX 2004). pp. 62–69. Society for Industrial and Applied Mathematics (2004)
2. Archetti, C., Bertazzi, L., Speranza, M.G.: Reoptimizing the traveling salesman problem. *Networks* 42(3), 154–159 (2003)
3. Ausiello, G., Escoffier, B., Monnot, J., Paschos, V.T.: Reoptimization of minimum and maximum traveling salesman's tours. In: Arge, L., Freivalds, R.V. (eds.) Proc. of the 10th Scandinavian Workshop on Algorithm Theory (SWAT 2006). LNCS, vol. 4059, pp. 196–207. Springer, Berlin (2006)
4. Böckenhauer, H.J., Forlizzi, L., Hromkovič, J., Kneis, J., Kupke, J., Proietti, G., Widmayer, P.: On the approximability of TSP on local modifications of optimally solved instances. *Algorithmic Oper. Res.* 2(2), 83–93 (2007)
5. Böckenhauer, H.J., Hromkovič, J., Klasing, R., Seibert, S., Unger, W.: Towards the notion of stability of approximation for hard optimization tasks and the traveling salesman problem. *Theor. Comput. Sci.* 285(1), 3–24 (2002)
6. Böckenhauer, H.J., Hromkovič, J., Mömke, T., Widmayer, P.: On the hardness of reoptimization. In: Geffert, V., Karhumäki, J., Bertoni, A., Preneel, B., Návrát, P., Bieliková, M. (eds.) Proc. of the 34th Int. Conf. on Current Trends in Theory and Practice of Computer Science (SOFSEM 2008). LNCS, vol. 4910, pp. 50–65. Springer, Berlin (2008)
7. Böckenhauer, H.J., Hromkovič, J., Seibert, S.: Stability of approximation. In: Gonzalez, T.F. (ed.) *Handbook of Approximation Algorithms and Metaheuristics*, Chapter 31. Chapman & Hall/CRC Computer and Information Science Series, Chapman & Hall/CRC (2007)
8. Böckenhauer, H.J., Klasing, R., Mömke, T., Steinová, M.: Improved approximations for TSP with simple precedence constraints. In: Calamoneri, T., Díaz, J. (eds.) Proc. of the 7th Int. Conf. on Algorithms and Complexity (CIAC 2010). LNCS, vol. 6078, pp. 61–72. Springer (2010)

9. Böckenhauer, H.J., Komm, D.: Reoptimization of the metric deadline TSP. *Journal of Discrete Algorithms* 8(1), 87–100 (2010)
10. Christofides, N.: Worst-case analysis of a new heuristic for the travelling salesman problem. Tech. Rep. 388, Graduate School of Industrial Administration, Carnegie-Mellon University (1976)
11. Downey, R.G., Fellows, M.R.: *Parameterized Complexity*. Monographs in Computer Science, Springer, New York (1999)
12. Eppstein, D.: Paired approximation problems and incompatible inapproximabilities. In: Charikar, M. (ed.) *Proc. of the 21st Annual ACM-SIAM Symp. on Discrete Algorithms (SODA 2010)*. pp. 1076–1086. Society for Industrial and Applied Mathematics (2010)
13. Fellows, M.R.: Blow-ups, win/win's, and crown rules: Some new directions in FPT. In: Bodlaender, H.L. (ed.) *Proc. of the 29th Int. Workshop on Graph-Theoretic Concepts in Computer Science (WG 2003)*. LNCS, vol. 2880, pp. 1–12. Springer, Berlin (2003)
14. Forlizzi, L., Hromkovič, J., Proietti, G., Seibert, S.: On the stability of approximation for Hamiltonian path problems. *Algorithmic Oper. Res.* 1(1), 31–45 (2006)
15. Guttmann-Beck, N., Hassin, R., Khuller, S., Raghavachari, B.: Approximation algorithms with bounded performance guarantees for the clustered traveling salesman problem. *Algorithmica* 28(4), 422–437 (2000)
16. Hoogeveen, J.A.: Analysis of Christofides' heuristic: some paths are more difficult than cycles. *Oper. Res. Lett.* 10(5), 291–295 (1991)
17. Hromkovič, J.: *Algorithmics for Hard Problems*. Introduction to Combinatorial Optimization, Randomization, Approximation, and Heuristics. Texts in Theoretical Computer Science. An EATCS Series, Springer, Berlin (2003)
18. Niedermeier, R.: *Invitation to Fixed Parameter Algorithms* (Oxford Lecture Series in Mathematics and Its Applications). Oxford University Press, USA (March 2006)
19. Papadimitriou, C.H., Vempala, S.: On the approximability of the traveling salesman problem. *Combinatorica* 26(1), 101–120 (2006)
20. Prieto, E., Sloper, C.: Either/or: using vertex cover structure in designing FPT-algorithms—the case of k -Internal Spanning Tree. In: Dehne, F.K.H.A., Sack, J.R., Smid, M.H.M. (eds.) *Proc. of the 8th Int. Workshop on Algorithms and Data Structures (WADS 2003)*. LNCS, vol. 2748, pp. 474–483. Springer, Berlin (2003)
21. Sahni, S., Gonzalez, T.F.: P-complete approximation problems. *J. ACM* 23(3), 555–565 (1976)
22. Vassilevska, V., Williams, R., Woo, S.L.M.: Confronting hardness using a hybrid approach. In: *Proc. of the 17th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA 2006)*. pp. 1–10. Society for Industrial and Applied Mathematics, New York (2006)
23. West, D.B.: *Introduction to Graph Theory*. Prentice Hall Inc., Upper Saddle River, NJ, second edn. (2000)