Electronic Colloquium on Computational Complexity, Report No. 76 (2009)

Improved and Derandomized Approximations for Two-Criteria Metric Traveling Salesman

Christian Glaßer Christian Reitwießner Maximilian Witek
University of Würzburg, Germany

19th August 2009

Abstract

We improve and derandomize the best known approximation algorithm for the two-criteria metric traveling salesman problem (2-TSP). More precisely, we construct a deterministic 2-approximation which answers an open question by Manthey.

Moreover, we show that 2-TSP is randomized $(3/2 + \varepsilon, 2)$-approximable, and we give the first randomized approximations for the two-criteria traveling salesman path problems 2-TSPP, 2-$\mathrm{TSPP_s}$, and 2-$\mathrm{TSPP_{st}}$. We further provide arguments that indicate the hardness of improving our randomized approximation algorithms in the sense that such improvements force us to improve the best known approximations for TSP, $\mathrm{TSPP_s}$, and $\mathrm{TSPP_{st}}$ (Christofides 1976, Hoogeveen 1991).

A particular interesting situation emerges for 2-TSP: Because of possible trade-offs between the approximation ratios in the first and in the second component, there could exist randomized approximation algorithms that are incomparable to our algorithms. For these we can narrow down the approximation ratios that could be within reach, i.e., that will not force us to improve the well-studied approximations by Christofides and Hoogeveen. This leads to the question of whether 2-TSP has an (α, β)-approximation where $5/3 \leq \alpha, \beta < 2$.

1 Introduction

The traveling salesman problem is one of the oldest combinatorial optimization problems. For a given set of cities, one has to find a shortest cycle that visits each city exactly once. This problem was first mentioned in 1831 as a problem of a traveling salesman who wants to cover as many locations as possible without visiting locations twice [Voi31]. The first reference as a mathematical optimization problem goes back to Karl Menger, who gave a definition in a colloquium in Vienna in 1930. In the 1950s and 1960s the traveling salesman problem became increasingly popular in mathematics and computer science.

The substantial majority of the variants of the traveling salesman problem encountered in practice (including all geometric versions) has metric distance functions [JP85]. In fact, the problem becomes metric, if we allow each city to be visited more than once. Therefore, with a minimum loss of generality, one often assumes a metric distance function and studies the *metric traveling salesman problem* (TSP). In our paper we follow this perspective and assume all distance functions to be metric. A special case of TSP is the *Euclidean* variant, where each

ISSN 1433-8092

city is located at some point in the plane, and the distance function of any two cities is given by their Euclidean distance.

In 1972, a breakthrough was achieved by Karp [Kar72] who proved the difficulty of TSP by showing its NP-completeness. This shows that the search for a polynomial-time algorithm for TSP is an extremely challenging endeavor and is as difficult as solving any NP-complete problem. However, within all NP-complete problems, there are great differences in the difficulty of computing *approximate solutions*. So the NP-completeness of TSP raises the question for good approximation algorithms.

For a long time, the best known approximation algorithm for TSP and Euclidean TSP was the simple tree-doubling method. Given the facts that there are spanning trees with costs less than an optimal Hamiltonian cycle (the cycle itself contains spanning trees as subgraphs), and that minimum spanning trees can be determined in polynomial time, we can use each edge of a minimum spanning tree twice and obtain a Hamiltonian cycle with approximation ratio 2. In 1976, Christofides [Chr76] improved these results significantly by showing that a combination of a minimum spanning tree with a minimum matching yields a Hamiltonian cycle with approximation ratio $3/2$. The latest breakthrough in this line of research was achieved by Arora [Aro98] who found a polynomial-time approximation scheme (PTAS) for the Euclidean TSP. However, after 30 years of research, Christofides' basic algorithm is still the best known approximation for the general TSP.

Regarding lower bounds, Papadimitriou and Vempala [PV06] showed that TSP cannot be approximated with a ratio better than $220/219$, unless $P = NP$. Another variant of TSP is studied by Papadimitriou and Yannakakis [PY93] who construct a $7/6$-approximation algorithm for TSP(1,2), which is the restriction of TSP where all distances are either 1 or 2. Furthermore, TSP motivates several path problems where for given cities, one has to find a shortest path that visits each city exactly once and that starts and ends in specified (resp., arbitrary) cities. To this end, Hoogeveen [Hoo91] introduced the problems TSPP, $\mathrm{TSPP_s}$, and $\mathrm{TSPP_{st}}$, which are the metric traveling salesman path problems with 0, 1, and 2 specified vertices. He showed $3/2$-approximations for TSPP and $\mathrm{TSPP_s}$, and a $5/3$-approximation for $\mathrm{TSPP_{st}}$.

Two-Criteria TSP: In this paper we study the traveling salesman problem in the presence of *two* metric cost functions (2-TSP). For instance, consider the distance of the route as the first and its customs duties as the second cost function. We are interested in finding a tour that minimizes both, the distance and the customs duties. Since these objectives are conflicting, we cannot hope for a single optimal solution, but there will be a trade-off between the objectives. The *Pareto curve* captures the notion of optimality in settings with multiple objectives. It consists of all solutions that are optimal in the sense that there is no solution that is better with respect to all objectives. So for a given situation (i.e., cities with distances and customs duties), the Pareto curve shows us all optimal decisions (i.e., optimal Hamiltonian tours).

For a general introduction to multi-criteria optimization we refer to the survey by Ehrgott and Gandibleux [EG00] and the textbook by Ehrgott [Ehr05]. Regarding the approximability of Pareto curves, Papadimitriou and Yannakakis [PY00] show the following important result: Every Pareto curve has a $(1+\varepsilon)$-approximation of size polynomial in the instance and $1/\varepsilon$. This means that even though a Pareto curve might be an exponentially large object, there always exists a polynomial-size approximation. This result clears the way for a general investigation of the approximability of Pareto curves of multi-criteria optimization problems.

The multi-criteria TSP and in particular 2-TSP was first studied by Gupta and Warburton [GW86]. Angel, Bampis, and Gourvès [ABG04] give a $3/2$-approximation for the two-criteria variant of TSP(1,2). Furthermore, Angel et al. [ABGM05] investigate the non-approximability of this problem. Ehrgott [Ehr00] studies 2-TSP with the normalization approach where a norm is used to aggregate both cost functions into one. This approach uses a definition of approximation ratios that differs from the two-component approximation ratios that we consider in our paper. Manthey and Ram [MR09] give a randomized $(2+\varepsilon)$-approximation algorithm for 2-TSP, which is the first approximation of the Pareto curve for the general 2-TSP.

Our Results: Manthey [Man09] notes that most approximation algorithms for multi-criteria TSP use randomness for computing approximate Pareto curves of cycle covers (resp., matchings), and he raises the question of whether these algorithms can be improved and derandomized:

Are there algorithms for multi-criteria TSP *that are faster, deterministic, and achieve better approximation ratios?*

We give a positive answer to this question in form of a deterministic approximation algorithm for 2-TSP whose approximation ratio is slightly better than the ratio of the currently best known randomized approximation. More precisely, Manthey and Ram [MR09] showed that 2-TSP is randomized $(2+\varepsilon)$-approximable, while in the present paper we improve this to a deterministic 2-approximation. Furthermore, our algorithm is faster than the algorithm by Manthey and Ram, since the expensive approximation of the Pareto-minimal matchings is replaced by an easy graph algorithm (cf. the algorithm `match` at page 9).

This improvement is based on an easy observation: Though deterministic approximations of Pareto-minimal matchings are not known (and computing these matchings exactly is NP-hard [PY00]), in the metric case it is easy to compute a matching in a spanning tree such that the matching has at most the same costs as the tree. Of course, such a matching is by far not optimal. Nevertheless, it suffices to improve the approximation, since at present, the bottleneck of approximations for 2-TSP is not the method of *finding* a good matching, but is the argument that a good matching *exists*.

By this observation we can approximate a Pareto-minimal Hamiltonian tour as follows: Start with a fairly accurate approximation of all Pareto-minimal spanning trees which is possible using a known FPTAS [PY00]. This approximation contains trees T_1 and T_2 such that T_1's costs in the first component (resp., T_2's costs in the second component) are clearly smaller than the corresponding costs of a Pareto-optimal Hamiltonian tour. Let U be the vertices in T_1 with odd degree. Deterministically extract from T_2 a perfect matching for U and combine it with T_1. This results in an Eulerian graph whose Eulerian tour is a 2-approximation for the Hamiltonian tour we started with.

Although our approximation is faster, deterministic, and has a better approximation ratio, it seems likely that the approximation by Manthey and Ram [MR09] shows a better performance in practice. The reason is that the randomized approximation of the Pareto-minimal matchings will most likely find matchings that are less expensive than the one we construct deterministically. If a particular application can cope with a performance loss by a factor of at most 2, then one can run both algorithms (or rather an optimized combination of them) in parallel. This results in an algorithm that guarantees the approximation ratio 2 and that with high probability yields a much better ratio (i.e., at least $(3/2+\varepsilon, 2)$ as shown in Theorem 3.5).

Besides the deterministic 2-approximation of 2-TSP we show that this problem is randomized $(3/2+\varepsilon, 2)$-approximable and randomized $(3/2, 2+\varepsilon)$-approximable, which is achieved basically by a more precise analysis of the algorithm by Manthey and Ram [MR09]. Note that the first component exactly meets the approximation ratio of Christofides' approximation [Chr76], which is still the best known approximation for TSP. Further we construct the first approximation algorithms for the two-criteria traveling salesman path problems 2-TSPP, 2-TSPP_s, and TSPP_st. Table 1 summarizes the ratios of the approximation algorithms that we provide in this paper.

Problem	Deterministic Approximation	Randomized Approximation	Ref.
2-TSP	$(2, 2)$	$(3/2 + \varepsilon, 2)$, $(3/2, 2 + \varepsilon)$	3.2, 3.4, 3.5
2-TSPP	$(2 + \varepsilon, 2 + \varepsilon)$	$(3/2 + \varepsilon, 5/3 + \varepsilon)$	4.6, 4.1
2-TSPP_s	$(2 + \varepsilon, 2 + \varepsilon)$	$(3/2 + \varepsilon, 2 + \varepsilon)$	4.6, 4.3
2-TSPP_st	$(2 + \varepsilon, 2 + \varepsilon)$	$(2 + \varepsilon, 2 + \varepsilon)$	4.5

Table 1: Summary of the approximation ratios obtained in this paper where $\varepsilon > 0$.

In the second part of the paper we give several arguments that indicate the hardness of improving our approximation algorithms. For this we demonstrate several approximation preserving reductions that allow us to translate well-studied problems like TSP or TSPP_st to the two-criteria optimization problems 2-TSP, 2-TSPP, and 2-TSPP_s. From this we obtain that certain improvements of our approximation algorithms for 2-TSP, 2-TSPP, and 2-TSPP_s force us to improve the best known approximation algorithms for TSP, TSPP_s, and TSPP_st [Chr76, Hoo91]. Considerable improvements of the latter, well-studied approximations seem very difficult to obtain, not least because Christofides' approximation for TSP and Hoogeveen's approximation for TSPP_st are unbeaten for decades. So we can conclude that the improvements of our algorithms are difficult as well. Table 2 summarizes the arguments that indicate the hardness of improving our approximation algorithms.

As a consequence of our results, we obtain a particular interesting situation for 2-TSP (cf. Figure 1): We know that 2-TSP is randomized $(3/2, 2+\varepsilon)$-approximable and randomized $(3/2+\varepsilon, 2)$-approximable. It is difficult to improve these approximations with respect to any component, and it is also difficult to obtain a $(5/3 - \varepsilon, 2 - \varepsilon)$-approximation. However, we have no evidence in favor of or against an (α, β)-approximation where $5/3 \leq \alpha, \beta < 2$. The search for such an algorithm remains a challenging open problem.

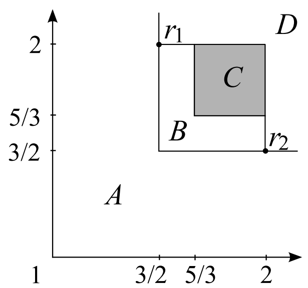

Figure 1: Approximation ratio results for 2-TSP. An approximation ratio inside A would immediately improve Christofides' approximation. We show evidence against an approximation ratio inside B, and further prove approximation ratio r_1 and r_2, hence area D is of no further interest. However, evidence against approximation algorithms within C remains an open question.

The paper is organized as follows. The preliminaries in section 2 give some basics on the concept of multi-criteria optimization and define the problems studied here. Section 3 contains

Problem	Randomized ratio proved in this paper	An improvement to ratio ...	... yields ratio of ...	... for problem	Ref.
2-TSP	$(3/2, 2+\varepsilon)$	$(3/2-\varepsilon, \alpha)$	$(3/2-\varepsilon)$	TSP	5.1
	$(3/2+\varepsilon, 2)$	$(5/3-\varepsilon, 2-\varepsilon)$	$(5/3-\varepsilon)$	$\mathrm{TSPP_{st}}$	5.7
2-TSPP	$(3/2+\varepsilon, 5/3+\varepsilon)$	$(3/2-\varepsilon, \alpha)$	$(3/2-\varepsilon)$	TSPP	5.8
		$(3/2-\varepsilon, 5/3-\varepsilon)$	$(5/3-\varepsilon)$	$\mathrm{TSPP_{st}}$	5.10
		$(3/2-\varepsilon, 2-\varepsilon)$	$(3/2-\varepsilon)$	$\mathrm{TSPP_{s}}$	5.12
2-$\mathrm{TSPP_{s}}$	$(3/2+\varepsilon, 2+\varepsilon)$	$(3/2-\varepsilon, \alpha)$	$(3/2-\varepsilon)$	$\mathrm{TSPP_{s}}$	5.13
		$(3/2+\varepsilon, 2-\varepsilon)$	$(3/2+\varepsilon)$	$\mathrm{TSPP_{st}}$	5.16.1
		$(3/2-\varepsilon, 2-\varepsilon)$	$(3/2-\varepsilon)$	TSP	5.16.2
		$(3/2-\varepsilon, 2-\varepsilon)$	$(3/2-\varepsilon)$	$\mathrm{TSPP_{st}}$	5.16.2
2-$\mathrm{TSPP_{st}}$	$(2+\varepsilon, 2+\varepsilon)$	$(5/3-\varepsilon, \alpha)$	$(5/3-\varepsilon)$	$\mathrm{TSPP_{st}}$	5.17

Table 2: Arguments that indicate the difficulty of improving the obtained randomized approximations where $\varepsilon > 0$ and $\alpha > 1$. The table shows that if the deterministic (resp., randomized) approximation ratio of a two-criteria problem is improved, then this also improves the deterministic (resp., randomized) approximation ratio of a well-studied optimization problem. For instance, if 2-$\mathrm{TSPP_s}$ is randomized $(3/2-\varepsilon, 2-\varepsilon)$-approximable, then TSP is randomized $(3/2-\varepsilon)$-approximable.

the deterministic and randomized approximation algorithms for 2-TSP, while section 4 provides these algorithms for the traveling salesman path problems 2-TSPP, 2-$\mathrm{TSPP_s}$, and 2-$\mathrm{TSPP_{st}}$. In section 5 we argue that it is difficult to obtain certain improvements regarding the approximation ratios of our algorithms. Finally, in section 6 we summarize the open questions of this paper.

2 Preliminaries

2.1 Multi-Criteria Optimization

The tuple $\mathcal{P} = (\mathcal{I}, \mathcal{S}, sol, m, goal)$ is a *k-objective optimization problem* with the set of valid instances $\mathcal{I}$, the solution space $\mathcal{S}$, a mapping $sol\colon \mathcal{I} \to 2^{\mathcal{S}}$ such that for every instance $x \in \mathcal{I}$, $sol(x)$ denotes the set of valid solutions for x, a measure function $m\colon \mathcal{I} \times \mathcal{S} \to \mathbb{N}^k$ with components m_i and an objective vector $goal$ with components $goal_i \in \{\min, \max\}$ for $1 \le i \le k$. To simplify our notation, if $x \in \mathcal{I}$ is clear from the context, we will write $m(y)$ instead of $m(x, y)$ for any $y \in sol(x)$, and if $k = 1$, we will identify the vectors with their unique component, hence $m_1 = m$, $goal_1 = goal$. We will only consider single-criterion and two-criteria minimization problems, hence, for the remainder of this paper, let $goal_1 = goal_2 = \min$. For simplicity, we will define the following concepts only for minimization problems. An extension to general optimization problems is straightforward.

Let $k \in \mathbb{N}$ and $\mathcal{P} = (\mathcal{I}, \mathcal{S}, sol, m, goal)$ be a k-objective optimization problem. We consider k-dimensional vectors $u, v \in \mathbb{R}^k$ and say $u \le v$ if $u_i \le v_i$ for all $i \in \{1, 2, \ldots, k\}$. Let arithmetic operations on such vectors be defined componentwise. For any problem instance $x \in \mathcal{I}$ and any two solutions $y_1, y_2 \in sol(x)$, we say that y_1 *dominates* y_2 if $m(y_1) \le m(y_2)$ and $m(y_1) \ne m(y_2)$.

For any $\alpha \in \mathbb{R}^k$ with $\alpha_i \geq 1$ for all i, we say that y_2 *α-approximates* y_1 if $m(y_2) \leq \alpha \cdot m(y_1)$. We call a solution *Pareto-optimal* if it is not dominated by another solution, and call the set of all Pareto-optimal solutions of an instance its *Pareto set* or, equivalently, its *Pareto curve*. Furthermore, we call a set of solutions $S \subseteq sol(x)$ an *α-approximate Pareto set* of x if for every solution $y_1 \in sol(x)$ there is a solution $y_2 \in S$ such that y_2 α-approximates y_1.

Let $\mathcal{A}$ be an algorithm and $\mathcal{F}$ be a family of algorithms such that, for every $\varepsilon \in \mathbb{R}^k$ with $\varepsilon_i > 0$ for all i, $\mathcal{F}$ contains an algorithm $\mathcal{A}_\varepsilon$.

We call $\mathcal{A}$ an *α-approximation algorithm* for $\mathcal{P}$ if for every $x \in \mathcal{I}$, the algorithm $\mathcal{A}(x)$ returns an α-approximate Pareto set of x in time polynomial in $|x|$ (i.e., the length of x). If, however, $\mathcal{A}(x)$ returns an α-approximate Pareto set of x *with probability at least* $1/2$ *over all executions of* $\mathcal{A}(x)$, we call $\mathcal{A}$ a *randomized α-approximation algorithm*. In all cases, we require $\mathcal{A}(x) \subseteq sol(x)$. If $\mathcal{A}$ is an α-approximation algorithm (randomized α-approximation algorithm, resp.) for $\mathcal{P}$, we call α the *approximation ratio of* $\mathcal{A}$ (*randomized approximation ratio of* $\mathcal{A}$, resp.).

We say that $\mathcal{F}$ is a *fully polynomial time approximation scheme (FPTAS) for* $\mathcal{P}$ if, for every $\varepsilon \in \mathbb{R}^k$ with $\varepsilon_i > 0$ for all i, $\mathcal{A}_\varepsilon$ is an $(1+\varepsilon)$-approximation algorithm for $\mathcal{P}$ that runs in time polynomial in $|x| + 1/\varepsilon_1 + \cdots + 1/\varepsilon_k$ for all $x \in \mathcal{I}$. Analogously, if some $\mathcal{A}_\varepsilon$ is randomized, we call $\mathcal{F}$ a *fully polynomial time randomized approximation scheme (FPRAS)*.

If there is some α-approximation algorithm (randomized α-approximation algorithm, resp.) for $\mathcal{P}$, we say that $\mathcal{P}$ is *α-approximable* (*randomized α-approximable*, resp.). If $\mathcal{P}$ is $(\alpha, \ldots, \alpha)$-approximable, we will simply say $\mathcal{P}$ is α-approximable.

2.2 Relevant Problems

Let V be a finite set (of vertices). A function $c\colon V \times V \to \mathbb{N}$ is called *pseudometric* if for all $u, v, w \in V$: $c(u,u) = 0$, $c(u,v) = c(v,u)$ and $c(u,v) \leq c(u,w) + c(w,v)$. c is called *metric* if additionally $c(u,v) > 0$ whenever $u \neq v$. We extend the (pseudo)metric property to functions $c\colon V \times V \to \mathbb{N}^k$ with k components c_i where each component c_i itself is (pseudo)metric. We further extend a pseudometric function c to sets of edges $S \subseteq V \times V$ by $c(S) = \sum_{(u,v)\in S} c(u,v)$. A set of edges $P \subseteq V \times V$ is a *Hamiltonian path* on a set of vertices V if there is a bijective numbering $\pi\colon \{1, 2, \ldots, \#V\} \to V$ such that $P = \{(\pi(1), \pi(2)), (\pi(2), \pi(3)), \ldots, (\pi(\#V-1), \pi(\#V))\}$ (i.e., a "connected" path that visits each vertex exactly once). We say that P starts at $\pi(1)$ and ends at $\pi(\#V)$. A set of edges $P' \subseteq V \times V$ is a *Hamiltonian tour* if $P' = P \cup \{(t,s)\}$ for some Hamiltonian path P starting at s and ending at t. For an undirected graph G and one of its vertices v, let $\deg_G(v)$ denote the degree of v in G, i.e., the number of edges incident to v. Note that this function can be computed in polynomial time.

We now define the main problems of this paper. Note that, since we only consider single-criterion and two-criteria minimization problems, for each of the following problems, we have $goal_1 = goal2 = \min$. Let us first consider the single-criterion minimization problems.

Traveling Salesman (TSP)

Instance: finite set V (cities), pseudometric function $c\colon V \times V \to \mathbb{N}$ (distances)

Solution: Hamiltonian tour $T \subseteq V \times V$

Measure: $c(T)$

Traveling Salesman Path (TSPP)

Instance: finite set V (cities), pseudometric function $c\colon V \times V \to \mathbb{N}$ (distances)

Solution: Hamiltonian path $P \subseteq V \times V$

Measure: $c(P)$

Traveling Salesman Path with Start Vertex ($\mathbf{TSPP_s}$)

Instance: finite set V (cities), pseudometric function $c\colon V \times V \to \mathbb{N}$ (distances), start point $s \in V$

Solution: Hamiltonian path $P \subseteq V \times V$ starting at s

Measure: $c(P)$

Traveling Salesman Path with Start and End Vertex ($\mathbf{TSPP_{st}}$)

Instance: finite set V (cities), pseudometric function $c\colon V \times V \to \mathbb{N}$ (distances), start point $s \in V$, end point $t \in V$

Solution: Hamiltonian path $P \subseteq V \times V$ starting at s and ending at t

Measure: $c(P)$

Minimum Spanning Tree (MST)

Instance: finite set V (vertices), function $c\colon V \times V \to \mathbb{N}$ (edge costs)

Solution: a spanning tree $T \subseteq V \times V$ of the complete graph $(V, V \times V)$

Measure: $c(T)$

Minimum Perfect Matching (MM)

Instance: finite set V (vertices), function $c\colon V \times V \to \mathbb{N}$ (edge costs)

Solution: a perfect matching $M \subseteq V \times V$ of the complete graph $(V, V \times V)$

Measure: $c(M)$

We can now define the following two-criteria minimization problems.

2-TSP, 2-TSPP, 2-$\mathbf{TSPP_s}$, 2-$\mathbf{TSPP_{st}}$, 2-MST, 2-MM

Let $\mathcal{P}$ be any of the single criterion minimization problems as defined above. We define the two-criteria minimization problem 2-$\mathcal{P}$ by considering two-dimensional functions $c\colon V \times V \to \mathbb{N}^2$ with components c_1, c_2 instead of one-dimensional functions. For 2-TSP, 2-TSPP, 2-$\mathrm{TSPP_s}$ and 2-$\mathrm{TSPP_{st}}$ we require c to be pseudometric.

Our algorithms will use the following known approximation schemes.

Theorem 2.1 ([PY00]) *There is an FPTAS for* 2-MST *and an FPRAS for* 2-MM*.*

We will refer to the FPTAS for 2-MST by `2-MST-Approx`, and denote an execution of `2-MST-Approx` on vertex set V with cost function c and approximation factor ε by `2-MST-Approx`(V, c, ε). Analogously, let $\texttt{2-MM-Approx}_{\texttt{R}}$ refer to the FPRAS for 2-MM, and let $\texttt{2-MM-Approx}_{\texttt{R}}(V, c, \varepsilon)$ denote an execution of $\texttt{2-MM-Approx}_{\texttt{R}}$ on vertex set V, cost function c and approximation factor ε. We will repeatedly call $\texttt{2-MM-Approx}_{\texttt{R}}$ in our randomized approximation algorithms. In each algorithm, we assume that $\texttt{2-MM-Approx}_{\texttt{R}}$ is amplified in a way such that the probability that *all* calls succeed is at least $1/2$.

3 Approximation of 2-TSP

The best known approximation result for 2-TSP is a randomized $(2+\varepsilon)$-approximation which was given by Manthey and Ram [MR09]. In this section we present an algorithm that improves this result in two ways: First, our algorithm is deterministic and not randomized. Second, the approximation ratio is slightly improved from $2+\varepsilon$ to 2.

The second part of this section contains a randomized algorithm that yields both a $(3/2+\varepsilon, 2)$-approximation and a $(3/2, 2+\varepsilon)$-approximation for 2-TSP. So the first component exactly meets the approximation ratio of Christofides' approximation, which is still the best known approximation for TSP.

3.1 Deterministic Approximation for 2-TSP

The deterministic 2-approximation for 2-TSP that is given below uses ideas from Christofides' approximation for TSP. However, we do not compute the perfect matching for the odd degree vertices by the known randomized approximation algorithm for two-criteria minimum matching [PY00]. Instead, we show that a suitable matching can be extracted *deterministically* and *error-free* from an approximate two-criteria minimum spanning tree. More precisely, the algorithm transforms a spanning tree into a matching of at most the same costs. In this way we avoid the randomized approximation algorithm for MM and hence avoid the associated randomness and the ε-error. Of course, the matching we obtain is by far not optimal (indeed computing Pareto-minimal matchings is NP-hard [PY00]). Nevertheless, this matching suffices to improve the approximation, since at present, the bottleneck of approximations for 2-TSP is not the method of *finding* a good matching, but is the argument that a good matching *exists*.

Let V be a finite set (of cities), $c\colon V \times V \mapsto \mathbb{N}^k$ a pseudometric (distance) function, $T \subseteq V \times V$ be a spanning tree of the complete graph $(V, V \times V)$, and $U \subseteq V$ be a set of vertices of even cardinality. By $\mathrm{p}_T(u,v) \subseteq T$ we denote the unique path from node u to node v in T. Note that this path can be computed in time polynomial in the size of T.

To extract a matching with costs less or equal to $c(T)$, we take an arbitrary perfect matching M on U and consider any distinct pair of matching elements $(u,u'),(v,v') \in M$. Assume that $\mathrm{p}_T(u,u')$ and $\mathrm{p}_T(v,v')$ intersect on at least one edge. We can easily remove this intersection by rearranging u, v, u', v' in M (cf. Figure 2). We repeat this process until there are no more intersections in M. It follows from the triangle inequality that $c(M) \leq c(T)$.

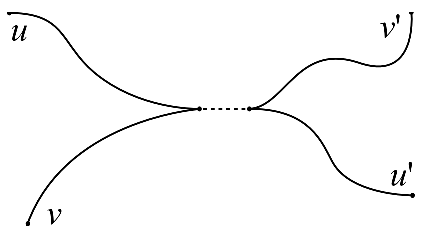

Figure 2: The paths $\mathrm{p}_T(u,u')$ and $\mathrm{p}_T(v,v')$ intersect on the dashed edges. Matching u with v and u' with v' will remove the intersecting edges and thereby improve the costs of M.

We now give a formal definition of the matching algorithm sketched above. In order to simplify the proofs, we use an iterative algorithm. We remark that there exists a recursive algorithm that has the same properties and that additionally runs in linear time.

Algorithm: match(U, T)

Input : A tree T and a subset U of its vertices of even cardinality

Output: A perfect matching M on U such that $c(M) \leq c(T)$ for any pseudometric function c

```
match(U, T)
begin
  find arbitrary perfect matching M ⊆ U × U;
  while there are distinct (u, u'), (v, v') ∈ M with p_T(u, u') ∩ p_T(v, v') ≠ ∅ do
    M := M \ {(u, u'), (v, v')};
    if p_T(u, v) ∩ p_T(u', v') = ∅ then
      M := M ∪ {(u, v), (u', v')}
    else
      M := M ∪ {(u, v'), (u', v)}
    end
  end
  return M
end
```

Lemma 3.1 *Let V be a finite set (of vertices), $k \geq 1$, $c\colon V \times V \mapsto \mathbb{N}^k$ be a pseudometric (distance) function, and $T \subseteq E$ be a spanning tree of the complete graph $G = (V, E)$ on V. Then, for any $U \subseteq V$ of even cardinality,* match(U, T) *will find a perfect matching M on U such that $c(M) \leq c(T)$ in polynomial time.*

Proof Let m denote the number of edges of T, and $S(M, T) = \sum_{(u,v) \in M} \#\mathrm{p}_T(u, v)$ be the sum of the number of edges of all paths used in T for some perfect matching M. Clearly, $S(M, T) \leq m^2/2$, since there are at most m edges per path and at most $m/2$ distinct matching pairs. In every iteration, we switch two matching pairs, which will reduce $S(M, T)$ by at least two. Hence, the algorithm will terminate after at most $m^2/4$ iterations. Since all operations of the algorithm (comparison of two unique paths in a tree and set operations) are polynomially fast, we obtain a polynomial-time algorithm.

After termination of the algorithm, for any two distinct $(u, v), (u', v') \in M$, we have distinct paths in the tree, i.e., $\mathrm{p}_T(u, v) \cap \mathrm{p}_T(u', v') = \emptyset$. By the triangle inequality we can now estimate the overall costs of M by

$$\begin{aligned} c(M) &= \sum_{(u,v) \in M} c(u, v) \\ &\leq \sum_{(u,v) \in M} c(\mathrm{p}_T(u, v)) \\ &\leq c(T). \end{aligned}$$

□

We proceed with the following deterministic algorithm that on input of a finite set V and a pseudometric function $c\colon V \times V \to \mathbb{N}^2$ computes a $(2, 2)$-approximation for 2-TSP. Please recall the definition of the algorithm 2-MST-Approx (text after Theorem 2.1).

Algorithm: `2-TSP-ApproxDet`(V, c)

Input : A finite set V (cities) and a pseudometric (distance) function $c\colon V \times V \mapsto \mathbb{N}^2$
Output: A set $S \subseteq \{T \subseteq V \times V \mid T \text{ is Hamiltonian tour of } V\}$

1 `2-TSP-ApproxDet(`V`, `c`)`
2 `begin`
3 $\varepsilon := \frac{1}{2\#V}$`;`
4 $S := \emptyset$`;`
5 $P :=$ `2-MST-Approx(`V`, `c`, `ε`);`
6 `foreach` $(T_1, T_2) \in P \times P$ `do`
7 $U := \{v \in V \mid \deg_{T_1}(v)$ `is odd`$\}$`;`
8 $M :=$ `match(`U`, `T_2`);`
9 $T_{\texttt{euler}} :=$ `Eulerian tour of` V `using the edges from` T_1 `and` M`;`
10 $T_{\texttt{approx}} :=$ `Hamiltonian tour computed from` $T_{\texttt{euler}}$ `by skipping previously visited vertices;`
11 $S := S \cup \{T_{\texttt{approx}}\}$
12 `end`
13 `return` S
14 `end`

Theorem 3.2 2-TSP *is* $(2,2)$*-approximable.*

Proof Let V be a finite set, $c\colon V \times V \rightarrow \mathbb{N}^2$ be a pseudometric (distance) function and $R \subseteq V \times V$ be an arbitrary Pareto-minimal Hamiltonian tour of V with respect to c. We show that `2-TSP-ApproxDet`(V, c) contains a Hamiltonian tour T_{approx} such that $c(T_{\text{approx}}) \leq 2c(R)$.

As in the algorithm `2-TSP-ApproxDet`, let $m = \#V$ and $\varepsilon = \frac{1}{2m}$. For any $i \in \{1, 2\}$ let r_i be the edge of R with maximum costs in the i-th criterion. It holds that $c_i(r_i) \geq \frac{1}{m} c_i(R)$, and, by removing r_i from R, we obtain spanning trees T_i' with the properties

$$\begin{aligned} c(T_1') &\leq ((1 - \frac{1}{m})c_1(R), c_2(R)) \text{ and} \\ c(T_2') &\leq (c_1(R), (1 - \frac{1}{m})c_2(R)). \end{aligned}$$

The FPTAS for the minimum spanning tree, `2-MST-Approx`(V, c, ε), provides an ε-approximation of every (minimum) spanning tree of G. So T_1' and T_2' are approximated by say T_1 and T_2 such that

$$\begin{aligned} c(T_1) &\leq (1 + \frac{1}{2m})c(T_1') \text{ and} \\ c(T_2) &\leq (1 + \frac{1}{2m})c(T_2'). \end{aligned}$$

The number of vertices of odd degree in an undirected graph is even, so $c(M) \leq c(T_2)$ by Lemma 3.1. Constructing the Eulerian tour in line 9 is possible since all vertices in the (multi-) graph with edges from T_1 and M have even degree. The Eulerian tour consists of exactly the edges in T_1 and in M (in a specific order) and skipping vertices in line 10 cannot increase the costs by the triangle equality. All in all we obtain

$$\begin{array}{rcl} c(T_{\text{approx}}) & \leq & c(T_{\text{euler}}) \\ & = & c(T_1) + c(M) \\ & \leq & c(T_1) + c(T_2) \\ & \leq & (1 + \dfrac{1}{2m})(c(T_1') + c(T_2')) \\ & \leq & (1 + \dfrac{1}{2m})((1 - \dfrac{1}{m}) + 1)c(R) \\ & = & (2 - \dfrac{1}{2m^2})c(R) \\ & < & 2c(R). \end{array}$$

Hence $c(T_{\text{approx}}) \leq 2c(R)$.

It remains to show that `2-TSP-ApproxDet` runs in polynomial time. The runtime of the FPTAS `2-MST-Approx` is polynomially bounded in $m + \frac{1}{\varepsilon} = 3m$. Thus, the cardinality of P itself is bounded by a polynomial in m, say p. For each of the p^2 combinations of spanning trees, the steps 7–11 can be done in polynomial time (cf. Lemma 3.1). Hence `2-TSP-ApproxDet` is a polynomial-time algorithm. □

3.2 Randomized Approximation for 2-TSP

The randomized algorithm that is given below provides both a $(3/2 + \varepsilon, 2)$-approximation and a $(3/2, 2+\varepsilon)$-approximation for 2-TSP. This algorithm is essentially the randomized approximation for 2-TSP that was given by Manthey and Ram [MR09]: First, one computes approximations of the Pareto-minimal spanning trees, considers the vertices that have odd degree in a single tree, computes approximations of the Pareto-minimal perfect matchings of these vertices, and finally pairwise combines all trees with all suitable matchings which results in several Eulerian tours. By a precise analysis of this algorithm, we obtain approximation ratios that are better than the ones stated in [MR09].

The algorithm below calls the algorithms `2-MST-Approx` and $\texttt{2-MM-Approx}_{\texttt{R}}$, which were defined after Theorem 2.1. The use of the FPTAS for the two-criteria minimum spanning tree problem is essential, as it allows us to reduce the error far enough such that it is dominated by the costs of a single edge in an optimal Hamiltonian tour. This makes it possible to remove an ε-error in one of the two criteria.

Algorithm: 2-TSP-ApproxRand$_\varepsilon$(V, c)

Input : A finite set V (cities) and a pseudometric (distance) function $c\colon V \times V \mapsto \mathbb{N}^2$
Output: A set $S \subseteq \{T \subseteq V \times V \mid T$ is Hamiltonian tour of $V\}$

1 2-TSP-ApproxRand$_\varepsilon$(V, c)
2 begin
3 $\quad m := \#V$; $\varepsilon_1 := \frac{\varepsilon}{m^2}$; $\varepsilon_2 := \frac{\varepsilon}{2m}$;
4 $\quad S := \emptyset$;
5 $\quad P :=$ 2-MST-Approx(V, c, ε_1);
6 $\quad$ foreach $T \in P$ do
7 $\quad\quad U := \{v \in V \mid \deg_T(v)$ is odd$\}$;
8 $\quad\quad A :=$ 2-MM-Approx$_\mathtt{R}$((U, U^2), c, ε_2);
9 $\quad\quad$ foreach $M \in A$ do
10 $\quad\quad\quad T_{\texttt{euler}} :=$ Eulerian tour of V using the edges from T and M;
11 $\quad\quad\quad T_{\texttt{approx}} :=$ Hamiltonian tour computed from $T_{\texttt{euler}}$ by skipping previously visited vertices;
12 $\quad\quad\quad S := S \cup \{T_{\texttt{approx}}\}$
13 $\quad\quad$ end
14 $\quad$ end
15 $\quad$ return S
16 end

Lemma 3.3 *For every $\varepsilon > 0$, the algorithm 2-TSP-ApproxRand$_\varepsilon$ runs in polynomial time.*

Proof Since 2-MST-Approx is an FPTAS, its running time is polynomial in $n + \frac{1}{\varepsilon_1} = n + \frac{m^2}{\varepsilon}$ where n is the size of the input (V, c). So we can obtain P in polynomial time and P contains only polynomially many elements. This means that the first loop is iterated polynomially often. 2-MM-Approx$_\mathtt{R}$ runs in polynomial time in the length of $((U, U^2), c)$ and thus also in n. This in turn means that A contains only polynomially many matchings so the second loop is iterated only polynomially often. The operations in lines 10 and 11 can obviously be carried out in polynomial time and thus the whole algorithm runs in polynomial time. $\square$

Theorem 3.4 2-TSP *is randomized* $(3/2 + \varepsilon, 2)$*-approximable for every* $\varepsilon > 0$.

Proof We show that 2-TSP-ApproxRand$_\varepsilon$ computes a $(3/2 + \varepsilon, 2)$-approximation for 2-TSP. By Lemma 3.3 the algorithm runs in polynomial time so it remains to show that the approximation is correct. Let $\varepsilon > 0$ and w.l.o.g. let $\varepsilon \leq 1$ (otherwise, just call the algorithm with $\varepsilon = 1$). Furthermore, assume that there are at least two cities in the input. We show that the algorithm 2-TSP-ApproxRand$_\varepsilon$ computes a $(3/2 + \varepsilon, 2)$-approximation for every pseudometric 2-TSP instance (V, c) with probability at least $1/2$.

Let V (a finite set) and $c\colon V \to \mathbb{N}^2$ (a pseudometric distance function) be the inputs for the algorithm 2-TSP-ApproxRand$_\varepsilon$. Furthermore, let $R = (r_1, \ldots, r_m) \subseteq V \times V$ be a Pareto-optimal Hamiltonian tour in G with respect to c. Choose an $r \in R$ with maximal costs in the second criterion. Hence $c_2(r) \geq \frac{1}{m} c_2(R)$. If we delete r from R, then we obtain a spanning tree T' of G such that

$$c(T') \quad \leq \quad (c_1(R), \frac{m-1}{m} c_2(R)).$$

By $\varepsilon_1 = \frac{\varepsilon}{m^2}$, the algorithm `2-MST-Approx` finds a spanning tree T with costs

$$c(T) \leq (1 + \frac{\varepsilon}{m^2})(c_1(R), \frac{m-1}{m} c_2(R)).$$

Let $U \subseteq V$ be the vertices of odd degree in T (U has even cardinality). From R we can easily find two distinct perfect matchings M_1 and M_2 on U such that $c(M_1) + c(M_2) \leq c(R)$: We simply use every other edge from the sets of edges obtained by connecting every vertex $u \in U$ to the next (with respect to the order of R) vertex in U. Hence, there is some perfect matching M' on U such that $c(M') \leq (\frac{1}{2} c_1(R), c_2(R))$. By $\varepsilon_2 = \frac{\varepsilon}{2m}$, there must be some approximate minimum matching M in A (with probability at least 1/2) such that

$$\begin{aligned} c(M) &\leq (1 + \frac{\varepsilon}{2m}) c(M') \\ &\leq (1 + \frac{\varepsilon}{2m})(\frac{1}{2} c_1(R), c_2(R)). \end{aligned}$$

Similarly to the argumentation in the proof of Theorem 3.2, we obtain the following (note that $m \geq 2$ and $\varepsilon \leq 1$):

$$\begin{aligned} c_1(T_{\text{approx}}) &\leq c_1(T) + c_1(M) \\ &\leq (1 + \frac{\varepsilon}{m^2}) c_1(R) + (1 + \frac{\varepsilon}{2m}) \frac{1}{2} c_1(R) \\ &= (\frac{3}{2} + \varepsilon(\frac{1}{m^2} + \frac{1}{4m})) c_1(R) \\ &\leq (\frac{3}{2} + \varepsilon) c_1(R) \end{aligned}$$

and

$$\begin{aligned} c_2(T_{\text{approx}}) &\leq c_2(T) + c_2(M) \\ &\leq ((1 + \frac{\varepsilon}{m^2}) \frac{m-1}{m} + (1 + \frac{\varepsilon}{2m})) c_2(R) \\ &\leq ((1 + \frac{1}{m^2}) \frac{m-1}{m} + 1 + \frac{1}{2m}) c_2(R) \\ &= (\frac{m-1}{m} + \frac{m-1}{m^3} + 1 + \frac{1}{2m}) c_2(R) \\ &= (2 - \frac{1}{m} + \frac{m-1}{m^3} + \frac{1}{2m}) c_2(R) \\ &= (2 + \frac{-m^2 + m - 1 + \frac{1}{2} m^2}{m^3}) c_2(R) \\ &= (2 + \frac{-\frac{1}{2} m^2 + m - 1}{m^3}) c_2(R) \\ &\leq (2 + \frac{-\frac{1}{2} m^2 + \frac{1}{2} m^2 - 1}{m^3}) c_2(R) \\ &= (2 - \frac{1}{m^3}) c_2(R) \\ &\leq 2 c_2(R) \end{aligned}$$

□

A similar estimation shows that 2-TSP-ApproxRand$_\varepsilon$ guarantees the approximation ratio $(3/2, 2+\varepsilon)$ as well, i.e., the ε is in the second instead of the first component.

Theorem 3.5 2-TSP *is randomized* $(3/2, 2+\varepsilon)$*-approximable for every* $\varepsilon > 0$.

Proof We show that 2-TSP-ApproxRand$_\varepsilon$ computes a $(3/2, 2+\varepsilon)$-approximation for 2-TSP. By Lemma 3.3 the algorithm runs in polynomial time so it remains to show that the approximation is correct. Let $\varepsilon > 0$ and w.l.o.g. let $\varepsilon \le 1$ (otherwise, just call the algorithm with $\varepsilon = 1$). Furthermore, assume that there are at least two cities in the input. We show that the algorithm 2-TSP-ApproxRand$_\varepsilon$ computes a $(3/2, 2+\varepsilon)$-approximation for every pseudometric 2-TSP instance (V, c) with probability at least $1/2$.

Let V (a finite set) and $c\colon V \to \mathbb{N}^2$ (a pseudometric distance function) be the inputs for the algorithm 2-TSP-ApproxRand$_\varepsilon$. Furthermore, let $R = (r_1, \ldots, r_m) \subseteq V \times V$ be a Pareto-optimal Hamiltonian tour in G with respect to c.

Choose an $r \in R$ with maximal costs in the first criterion. Hence $c_1(r) \ge \frac{1}{m} c_1(R)$. If we delete r from R we obtain a spanning tree T' of G such that

$$c(T') \quad \le \quad (\frac{m-1}{m} c_1(R), c_2(R)).$$

By $\varepsilon_1 = \frac{\varepsilon}{m^2}$, the algorithm 2-MST-Approx finds a spanning tree T with costs

$$c(T) \quad \le \quad (1 + \frac{\varepsilon}{m^2})(\frac{m-1}{m} c_1(R), c_2(R)).$$

Let $U \subseteq V$ be the vertices of odd degree in T (U has even cardinality). From R we can easily find two distinct perfect matchings M_1 and M_2 on U such that $c(M_1) + c(M_2) \le c(R)$, we simply use every other edge from the sets of edges obtained by connecting every vertex $u \in U$ to the next (with respect to the order of R) vertex in U. Hence, there is some perfect matching M' on U such that $c(M') \le (\frac{1}{2} c_1(R), c_2(R))$. By $\varepsilon_2 = \frac{\varepsilon}{2m}$, there must be some approximate minimum matching M in A (with probability at least $\frac{1}{2}$) such that

$$\begin{aligned} c(M) \quad &\le \quad (1 + \frac{\varepsilon}{2m}) c(M') \\ &\le \quad (1 + \frac{\varepsilon}{2m})(\frac{1}{2} c_1(R), c_2(R)). \end{aligned}$$

Similarly to the argumentation in the proof of Theorem 3.2, we obtain the following (note that $m \ge 2$ and $\varepsilon \le 1$):

$$
\begin{aligned}
c_1(T_{\text{approx}}) &\leq c_1(T) + c_1(M) \\
&\leq (1 + \frac{\varepsilon}{m^2})\frac{m-1}{m}c_1(R) + (1 + \frac{\varepsilon}{2m})\frac{1}{2}c_1(R) \\
&\leq (1 + \frac{1}{m^2})\frac{m-1}{m}c_1(R) + (1 + \frac{1}{2m})\frac{1}{2}c_1(R) \\
&= (\frac{m-1}{m} + \frac{m-1}{m^3} + \frac{1}{2} + \frac{1}{4m})c_1(R) \\
&= (\frac{3}{2} - \frac{1}{m} + \frac{m-1}{m^3} + \frac{1}{4m})c_1(R) \\
&= (\frac{3}{2} + \frac{-m^2 + m - 1 + \frac{1}{4}m^2}{m^3})c_1(R) \\
&= (\frac{3}{2} + \frac{-1 + m - \frac{3}{4}m^2}{m^3})c_1(R) \\
&\leq (\frac{3}{2} - \frac{1}{m^3})c_1(R) \\
&\leq \frac{3}{2}c_1(R)
\end{aligned}
$$

and

$$
\begin{aligned}
c_2(T_{\text{approx}}) &\leq c_2(T) + c_2(M) \\
&\leq (1 + \frac{\varepsilon}{m^2})c_2(R) + (1 + \frac{\varepsilon}{2m})c_2(R) \\
&\leq (1 + \frac{\varepsilon}{4})c_2(R) + (1 + \frac{\varepsilon}{4})c_2(R) \\
&\leq (2 + \varepsilon)c_2(R).
\end{aligned}
$$

□

4 Approximation of Traveling Salesman Path Problems

This section provides the following approximation algorithms:

- randomized $(3/2 + \varepsilon, 5/3 + \varepsilon)$-approximation for 2-TSPP
- randomized $(3/2 + \varepsilon, 2 + \varepsilon)$-approximation for 2-$\text{TSPP}_\text{s}$
- $(2 + \varepsilon, 2 + \varepsilon)$-approximation for 2-$\text{TSPP}_\text{st}$

The deterministic approximation for 2-TSPP_st is easily obtained by a tree-doubling of the approximated Pareto-minimal spanning trees. The constructions of the randomized approximation algorithms are similar to each other, but more complicated. Each of them relies on an argument that assures the existence of a matching with sufficiently low costs. These matchings are constructed in separate lemmas using combinatorial arguments. With these lemmas at hand we can follow a strategy similar to Christofides' approximation for TSP: We compute approximations of the Pareto-minimal spanning trees and, for every single tree, we consider the vertices that have odd degree, compute approximations of the Pareto-minimal matchings of these vertices and finally pairwise combine these matchings with their corresponding tree.

Theorem 4.1 2-TSPP *is randomized* $(3/2+\varepsilon, 5/3+\varepsilon)$*-approximable for every* $\varepsilon > 0$.

For the proof of this theorem we need the following argument which assures the existence of a matching with sufficiently low costs.

Lemma 4.2 *Let* V *be a finite set of vertices,* $c\colon V \times V \to \mathbb{N}^2$ *a pseudometric distance function, and let* $U \subseteq V$ *be a nonempty set of even cardinality. Then, for every Hamiltonian path* P *on* V *there exists a matching* m *on* U *that leaves exactly two vertices of* N *unmatched and that has costs* $c(m) \leq (\frac{1}{2}c_1(P), \frac{2}{3}c_2(P))$.

For the moment we postpone the proof of this lemma and start to show the approximation result on 2-TSPP.

Proof of Theorem 4.1 Let $\varepsilon > 0$. The approximation is achieved by the following algorithm which works on input of a finite set V of vertices and a pseudometric distance function $c\colon V \times V \to \mathbb{N}^2$. Please recall the definition of the algorithms `2-MST-Approx` and $\texttt{2-MM-Approx}_{\texttt{R}}$ (text after Theorem 2.1).

Algorithm: $\texttt{2-TSPP-Approx}_\varepsilon(V,\ c)$

Input : A finite set V (cities) and a pseudometric (distance) function $c\colon V \times V \to \mathbb{N}^2$
Output: A set $S \subseteq \{P \subseteq V \times V \mid P \text{ is Hamiltonian path of } V\}$

1 $\texttt{2-TSPP-Approx}_\varepsilon(V,\ c)$
2 **begin**
3 $\quad S := \emptyset$;
4 $\quad P :=$ `2-MST-Approx`$(V,\ c,\ \frac{\varepsilon}{2})$;
5 $\quad$ **foreach** $T \in P$ **do**
6 $\qquad U := \{v \in V \mid \deg_T(v) \text{ is odd}\}$;
7 $\qquad$ **foreach** $s, t \in U$ **with** $s \neq t$ **do**
8 $\qquad\quad A := \texttt{2-MM-Approx}_{\texttt{R}}(N \setminus \{s,t\},\ c,\ \frac{\varepsilon}{2})$;
9 $\qquad\quad$ **foreach** $M \in A$ **do**
10 $\qquad\qquad P_{\texttt{euler}}$:= Eulerian path from s to t using the edges from M and T;
11 $\qquad\qquad P_{\texttt{approx}}$:= Hamiltonian tour computed from $P_{\texttt{euler}}$ by skipping previously visited vertices;
12 $\qquad\qquad S := S \cup \{P_{\texttt{approx}}\}$
13 $\qquad\quad$ **end**
14 $\qquad$ **end**
15 $\quad$ **end**
16 $\quad$ **return** S
17 **end**

Observe that the set U in line 6 is nonempty and has an even number of elements. Also, note that in line 10, the Eulerian path exists, since after combining M and T, the vertices s and t have odd degree, while all remaining vertices have even, nonzero degree. Since line 8 uses an FPRAS [PY00], our algorithm is randomized. Observe that each line of the algorithm is computable in polynomial time. Therefore, $\texttt{2-TSPP-Approx}_\varepsilon$ is a randomized polynomial-time algorithm.

It remains to argue that `2-TSPP-Approx`$_\varepsilon$ computes a $(3/2+\varepsilon, 5/3+\varepsilon)$-approximate Pareto set. For this, let $P^* \subseteq V \times V$ denote an arbitrary Pareto-minimal Hamiltonian path. We show that `2-TSPP-Approx`$_\varepsilon$ outputs at least one Hamiltonian path P_{approx} such that

$$c_1(P_{\text{approx}}) \leq (\frac{3}{2}+\varepsilon)c_1(P^*) \quad \text{and} \quad c_2(P_{\text{approx}}) \leq (\frac{5}{3}+\varepsilon)c_2(P^*).$$

Fix a spanning tree T_{approx} with costs $c(T_{\text{approx}}) \leq (1+\frac{\varepsilon}{2})c(P^*)$ from the $(1+\frac{\varepsilon}{2})$-approximate Pareto set P computed in line 4. P contains such a tree, because P^* is a spanning tree on V, and for every spanning tree, the algorithm finds an approximation within ratio $(1+\frac{\varepsilon}{2})$. From now on we consider the iteration of the loop in line 5 that uses the tree T_{approx}.

By Lemma 4.2, there exists a matching M on U that leaves exactly two vertices $s, t \in N$ unmatched and that has costs $c(M) \leq (\frac{1}{2}c_1(P^*), \frac{2}{3}c_2(P^*))$. Therefore, in line 8, the $(1+\frac{\varepsilon}{2})$-approximate Pareto set A contains a matching M_{approx} that leaves some s and t unmatched such that

$$c(M_{\text{approx}}) \;\leq\; (1+\frac{\varepsilon}{2})\cdot\Big(\frac{1}{2}c_1(P^*), \frac{2}{3}c_2(P^*)\Big) \;\leq\; \bigg(\Big(\frac{1}{2}+\frac{\varepsilon}{2}\Big)c_1(P^*), \Big(\frac{2}{3}+\frac{\varepsilon}{2}\Big)c_2(P^*)\bigg).$$

We combine T_{approx} and M_{approx} to obtain an Eulerian tour P_{euler} from s to t with costs

$$\begin{aligned} c(P_{\text{euler}}) &= c(T_{\text{approx}}) + c(M_{\text{approx}}) \\ &\leq (1+\frac{\varepsilon}{2})c(P^*) + \bigg(\Big(\frac{1}{2}+\frac{\varepsilon}{2}\Big)c_1(P^*), \Big(\frac{2}{3}+\frac{\varepsilon}{2}\Big)c_2(P^*)\bigg) \\ &\leq \bigg(\Big(\frac{3}{2}+\varepsilon\Big)c_1(P^*), \Big(\frac{5}{3}+\varepsilon\Big)c_2(P^*)\bigg). \end{aligned}$$

Observe that taking shortcuts in P_{euler} to obtain a Hamiltonian path P_{approx} does not increase costs. Hence P^* is $(3/2+\varepsilon, 5/3+\varepsilon)$-approximated by P_{approx}. □

Proof of Lemma 4.2 We show the lemma by contradiction. So assume that Lemma 4.2 does not hold. Hence for every matching m on the vertices of U that leaves exactly two vertices unmatched it holds that

$$c_1(m) > \tfrac{1}{2}c_1(P) \quad \text{or} \quad c_2(m) > \tfrac{2}{3}c_2(P). \tag{1}$$

This also holds for *perfect* matchings on U, since otherwise, by removing an arbitrary edge, we obtain a matching that leaves two vertices unmatched and that contradicts (1).

Let $S = (v_1, \ldots, v_{2l})$ denote the vertices of U in the order in which they are visited by P. Let P' be the path that visits exactly the vertices of U (taking shortcuts compared with P). Since c is pseudometric, we have $c(P') \leq c(P)$. For the remaining part of this proof we use two distinct matchings that partition the edges of P' into *odd* and *even edges*:

$$\begin{aligned} m_{\text{odd}} &= \{(v_1,v_2),(v_3,v_4),\ldots,(v_{2l-1},v_{2l})\} \\ m_{\text{even}} &= \{(v_2,v_3),(v_4,v_5),\ldots,(v_{2l-2},v_{2l-1})\} \end{aligned}$$

Case 1: $c_2(m_{\text{odd}}) \leq \frac{2}{3}c_2(P)$ and $c_2(m_{\text{even}}) \leq \frac{2}{3}c_2(P)$.

From $c_1(m_{\text{even}}) + c_1(m_{\text{odd}}) = c_1(P') \leq c_1(P)$ it follows that $c_1(m_{\text{odd}}) \leq \frac{1}{2}c_1(P)$ or $c_1(m_{\text{even}}) \leq \frac{1}{2}c_1(P)$. So at least one of the matchings m_{odd} and m_{even} contradicts (1).

Case 2: $c_2(m_{\text{odd}}) > \frac{2}{3}c_2(P)$ or $c_2(m_{\text{even}}) > \frac{2}{3}c_2(P)$.

We assume $c_2(m_{\text{odd}}) > \frac{2}{3}c_2(P)$; the case $c_2(m_{\text{even}}) > \frac{2}{3}c_2(P)$ is treated analogously. Since $c_2(m_{\text{even}}) + c_2(m_{\text{odd}}) = c_2(P') \leq c_2(P)$, we have $c_2(m_{\text{even}}) \leq c_2(P) - c_2(m_{\text{odd}}) < \frac{1}{3}c_2(P)$ and thus

$$c_2(m_{\text{even}}) < \tfrac{1}{3}c_2(P). \tag{2}$$

For every odd $1 \leq k < 2l$, the part of P' that lies left (resp., right) of the edge (v_k, v_{k+1}) is denoted by l_k (resp., r_k), i.e.,

$$l_k = \{(v_i, v_{i+1}) \mid 1 \leq i < k\} \qquad \text{and} \qquad r_k = \{(v_i, v_{i+1}) \mid k+1 \leq i < 2l\}.$$

Consider the largest odd k such that $c_2(l_k \cap m_{\text{odd}}) \leq \frac{1}{2}c_2(m_{\text{odd}})$. From $(v_k, v_{k+1}) \in m_{\text{odd}}$ and $(v_k, v_{k+1}) \notin l_k \cup r_k$ it follows that $c_2(r_k \cap m_{\text{odd}}) \leq \frac{1}{2}c_2(m_{\text{odd}})$.

We now show that either the matching $m_1 = (l_k \cap m_{\text{odd}}) \cup (r_k \cap m_{\text{even}})$ or the matching $m_2 = (l_k \cap m_{\text{even}}) \cup (r_k \cap m_{\text{odd}})$ is a matching that contradicts (1). Observe that both m_1 and m_2 leave exactly two vertices of U unmatched (namely $\{v_k, v_{2l}\}$ or $\{v_1, v_{k+1}\}$, see Figure 3).

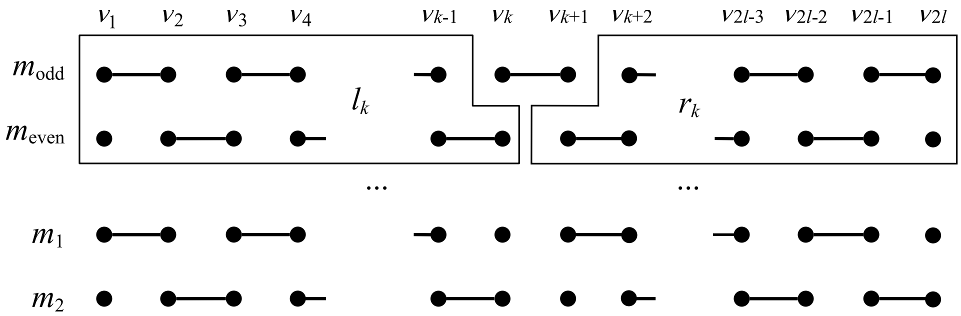

Figure 3: We obtain m_1 by merging the left part of m_{odd} with the right part of m_{even}, leaving v_k and v_{2l} unmatched. Analogously, we obtain m_2 by merging the left part of m_{even} with the right part of m_{odd}, leaving v_1 and v_{k+1} unmatched.

Let us estimate the costs in the second component of m_1 and m_2:

$$\begin{aligned} c_2(m_1) &= c_2(l_k \cap m_{\text{odd}}) + c_2(r_k \cap m_{\text{even}}) \\ &\leq \tfrac{1}{2}c_2(m_{\text{odd}}) + c_2(m_{\text{even}}) \end{aligned} \qquad \begin{aligned} c_2(m_2) &= c_2(l_k \cap m_{\text{even}}) + c_2(r_k \cap m_{\text{odd}}) \\ &\leq c_2(m_{\text{even}}) + \tfrac{1}{2}c_2(m_{\text{odd}}) \end{aligned}$$

By (2) we know that $c_2(m_{\text{even}}) < \frac{1}{3}c_2(P)$ and thus we obtain

$$\begin{aligned} c_2(m_1), c_2(m_2) &\leq \tfrac{1}{2}c_2(m_{\text{odd}}) + c_2(m_{\text{even}}) \\ &= \tfrac{1}{2}(c_2(P') - c_2(m_{\text{even}})) + c_2(m_{\text{even}}) \\ &= \tfrac{1}{2}(c_2(P') + c_2(m_{\text{even}})) \\ &< \tfrac{1}{2}(c_2(P) + \tfrac{1}{3}c_2(P)) \\ &= \tfrac{2}{3}c_2(P). \end{aligned}$$

Note that m_1 and m_2 are disjoint. Therefore, $c_1(m_1) + c_1(m_2) \leq c_1(P)$ and hence

$$c_1(m_1) \leq \tfrac{1}{2}c_1(P) \quad \text{or} \quad c_1(m_2) \leq \tfrac{1}{2}c_1(P).$$

So m_1 or m_2 contradicts (1). This finished the proof of Lemma 4.2. $\square$

Theorem 4.3 2-TSPP$_\mathrm{s}$ *is randomized* $(3/2 + \varepsilon, 2 + \varepsilon)$*-approximable for every* $\varepsilon > 0$.

The following lemma again guarantees the existence of a matching with low costs. Note that, since we only leave one vertex unmatched, we have less choices than we had in Lemma 4.2, and therefore can only guarantee a higher bound.

Lemma 4.4 *Let* V *be a finite set of vertices,* $c\colon V \times V \to \mathbb{N}^2$ *a pseudometric distance function, and let* $U \subseteq V$ *be a set of odd cardinality. Then, for every Hamiltonian path* P *on* V *there exists a matching* m *on* U *that leaves exactly one vertex of* U *unmatched and that has costs* $c(m) \leq (\frac{1}{2}c_1(P), c_2(P))$.

Proof of Lemma 4.4 Let $S = (v_1, \ldots, v_{2l+1})$ denote the vertices of U in the order in which they are visited by P. Let P' be the path that visits exactly the vertices of S (taking shortcuts compared with P). Since c_1 and c_2 are pseudometrics, we have $c(P') \leq c(P)$. Define the following distinct matchings that partition the edges of P'.

$$\begin{aligned} m_{\mathrm{odd}} &= \{(v_1, v_2), (v_3, v_4), \ldots, (v_{2l-1}, v_{2l})\} \\ m_{\mathrm{even}} &= \{(v_2, v_3), (v_4, v_5), \ldots, (v_{2l}, v_{2l+1})\} \end{aligned}$$

Note that $c_2(m_{\mathrm{odd}}) \leq c_2(P') \leq c_2(P)$ and $c_2(m_{\mathrm{even}}) \leq c_2(P') \leq c_2(P)$. From $c_1(m_{\mathrm{even}}) + c_1(m_{\mathrm{odd}}) = c_1(P') \leq c_1(P)$ it follows that $c_1(m_{\mathrm{odd}}) \leq \frac{1}{2}c_1(P)$ or $c_1(m_{\mathrm{even}}) \leq \frac{1}{2}c_1(P)$. So at least one of the matchings m_{odd} and m_{even} has costs $\leq (\frac{1}{2}c_1(P), c_2(P))$. $\square$

With Lemma 4.4 at hand we now proceed to show the approximation result on 2-TSPP$_\mathrm{s}$.

Proof of Theorem 4.3 The proof is similar to the proof of Theorem 4.1. Therefore, the proof below concentrates on the details that are different.

Let $\varepsilon > 0$. The approximation is achieved by the following algorithm which works on input of a finite set V of vertices, a pseudometric distance function $c\colon V \times V \to \mathbb{N}^2$ and a starting vertex $s \in V$.

Algorithm: 2-TSPP$_s$-Approx$_\varepsilon$(V, c, s)

Input : A finite set V (cities), a pseudometric (distance) function $c\colon V \times V \to \mathbb{N}^2$ and some $s \in V$ (starting city)

Output: A set $S \subseteq \{P \subseteq V \times V \mid P \text{ is Hamiltonian path of } V \text{ starting at } s\}$

1 2-TSPP$_s$-Approx$_\varepsilon$(V, c, s)
2 begin
3   $S := \emptyset$;
4   $P :=$ 2-MST-Approx(V, c, $\frac{\varepsilon}{2}$);
5   foreach $T \in P$ do
6     $U := \{v \in V \mid \deg_T(v)$ is odd if and only if $v \neq s\}$;
7   foreach $t \in U$ do
8       $A :=$ 2-MM-Approx$_\mathtt{R}$($U \setminus \{t\}$, c, $\frac{\varepsilon}{2}$);
9   foreach $M \in A$ do
10         $P_{\mathtt{euler}}$:= Eulerian path from s to t using the edges from M and T;
11         $P_{\mathtt{approx}} :=$ Hamiltonian tour computed from $P_{\mathtt{euler}}$ by skipping previously visited vertices;
12         $S := S \cup \{P_{\mathtt{approx}}\}$
13   end
14   end
15   end
16   return S
17 end

Observe that the set U in line 6 has an odd number of elements. Also, note that in line 10, the Eulerian path exists, since after combining M and T, the vertices s, t have odd degree, while all remaining vertices have even degree. We obtain that 2-TSPP$_s$-Approx$_\varepsilon$ is a randomized polynomial-time algorithm.

It remains to argue that 2-TSPP$_s$-Approx$_\varepsilon$ computes a $(3/2 + \varepsilon, 2 + \varepsilon)$-approximate Pareto set. For this, let $P^* \subseteq V \times V$ denote a Pareto-minimal Hamiltonian path that starts in s. We show that 2-TSPP$_s$-Approx$_\varepsilon$ contains at least one Hamiltonian path P_{approx} that starts in s such that

$$c_1(P_{\text{approx}}) \leq (\frac{3}{2} + \varepsilon)c_1(P^*) \quad \text{and} \quad c_2(P_{\text{approx}}) \leq (2 + \varepsilon)c_2(P^*).$$

Fix a spanning tree T_{approx} with costs $c(T_{\text{approx}}) \leq (1 + \frac{\varepsilon}{2})c(P^*)$ from the $(1 + \frac{\varepsilon}{2})$-approximate Pareto set P computed in line 4. From now on we consider the iteration of the loop in line 5 that uses the tree T_{approx}.

By Lemma 4.4, there exists a matching M on U that leaves exactly one vertex $t \in U$ unmatched and that has costs $c(M) \leq (\frac{1}{2}c_1(P^*), c_2(P^*))$. Therefore, in line 8 the approximate Pareto set A contains a matching M_{approx} that leaves some t unmatched such that

$$c(M_{\text{approx}}) \;\leq\; (1 + \frac{\varepsilon}{2}) \cdot \Big(\frac{1}{2}c_1(P^*), c_2(P^*)\Big) \;\leq\; \Bigg(\Big(\frac{1}{2} + \frac{\varepsilon}{2}\Big)c_1(P^*), \Big(1 + \frac{\varepsilon}{2}\Big)c_2(P^*)\Bigg).$$

We combine T_{approx} and M_{approx} to obtain an Eulerian tour P_{euler} from s to t with costs

$$\begin{aligned}
c(P_{\text{euler}}) &= c(T_{\text{approx}}) + c(M_{\text{approx}}) \\
&\leq (1+\frac{\varepsilon}{2})c(P^*) + \left(\left(\frac{1}{2}+\frac{\varepsilon}{2}\right)c_1(P^*), \left(1+\frac{\varepsilon}{2}\right)c_2(P^*)\right) \\
&\leq \left(\left(\frac{3}{2}+\varepsilon\right)c_1(P^*), \left(2+\varepsilon\right)c_2(P^*)\right).
\end{aligned}$$

Hence P^* is $(3/2+\varepsilon, 2+\varepsilon)$-approximated by P_{approx}. □

Theorem 4.5 2-TSPP$_{\text{st}}$ *is* $(2+\varepsilon, 2+\varepsilon)$*-approximable for every* $\varepsilon > 0$*.*

Proof We argue that the tree doubling method will deterministically find a $(2+\varepsilon, 2+\varepsilon)$-approximate Pareto set for 2-TSPP$_{\text{st}}$.

For a finite set V of vertices, $s, t \in V$ with $s \neq t$ and $\varepsilon > 0$, let $A = \texttt{2-MST-Approx}(V, c, \frac{\varepsilon}{2})$. For each tree $T \in A$ we do the following: We double each edge in T and then delete the unique path from s to t once. Clearly, we obtain a connected multigraph whose vertices have even degree except for s and t. Therefore we can easily find a Hamiltonian path $P \subseteq V \times V$ from s to t, having costs $c(P) \leq 2c(T)$.

Fix any arbitrary Pareto-minimal Hamiltonian path $P^* \subseteq V \times V$ from s to t. Since P^* is a spanning tree, there is a spanning tree $T \in A$ such that $c(T) \leq (1+\frac{\varepsilon}{2})c(P^*)$. By the tree doubling method we get a Hamiltonian path $P \subseteq V \times V$ from s to t with $c(P) \leq 2c(T) \leq (2+\varepsilon)c(P^*)$. □

Corollary 4.6 2-TSPP *and* 2-TSPP$_{\text{s}}$ *are* $(2+\varepsilon, 2+\varepsilon)$*-approximable for every* $\varepsilon > 0$*.*

5 Lower Bound Arguments

In this section we provide several arguments that indicate the hardness of improving the two-criteria approximation algorithms given in sections 3 and 4. In summary, our randomized approximations for 2-TSP, 2-TSPP, and 2-TSPP$_{\text{s}}$ cannot be improved, unless at the same time one improves the best known approximations for TSP, TSPP$_{\text{s}}$, and TSPP$_{\text{st}}$. Considerable improvements of the latter, well-studied approximations seem very difficult to obtain, not least because they are unbeaten for decades. So we can conclude that the improvements of our algorithms are difficult as well. Table 2 summarizes the lower bound arguments obtained in this section.

5.1 Lower Bound Arguments for 2-TSP

Below we construct an approximation preserving reduction from TSPP$_{\text{st}}$ to 2-TSP. This gives evidence for the difficulty of improving the randomized approximations for 2-TSP that are given in Theorems 3.4 and 3.5.

If for any of these approximations one could improve the second component, then this would result in a *considerable* improvement of the currently best known approximation algorithm for $\mathrm{TSPP_{st}}$. More precisely, the $5/3$-approximation that is known by Hoogeveen [Hoo91] could then be replaced by a randomized $(3/2+\varepsilon)$-approximation. Such an improvement, if possible at all, seems hard to obtain. Moreover, if for any of the approximations one could improve the first component by an ε, then this would improve Christofides' approximation for TSP [Chr76].

So the situation is as follows: We know that 2-TSP is randomized $(3/2, 2+\varepsilon)$-approximable and $(3/2+\varepsilon, 2)$-approximable. It is difficult to improve these approximations with respect to any component. Moreover, by Corollary 5.7 below, it is difficult to obtain a $(5/3-\varepsilon, 2-\varepsilon)$-approximation. However, we have no evidence in favor of or against an (α,β)-approximation where $5/3 \leq \alpha, \beta < 2$. The question for such an algorithm remains open.

The first lower bound argument is the easy observation that each approximation algorithm for 2-TSP can be used as an approximation algorithm for TSP.

Proposition 5.1 *Let $\alpha > 1$ and $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* 2-TSP *is $(3/2-\varepsilon, \alpha)$-approximable, then* TSP *is $(3/2-\varepsilon)$-approximable.*

Theorem 5.2 *Let $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* 2-TSP *is $(\alpha, 2-\varepsilon)$-approximable, then* $\mathrm{TSPP_{st}}$ *is α-approximable.*

Proof Let $\mathcal{A}$ be an algorithm that on input of a finite set V and a pseudometric (distance) function $c\colon V\times V \to \mathbb{N}^2$ returns an $(\alpha, 2-\varepsilon)$-approximation for 2-TSP for some $\alpha \geq 1$ and some $\varepsilon > 0$. Let (V, c', s, t) be an arbitrary $\mathrm{TSPP_{st}}$-instance where $V = \{s, t, v_1, \ldots, v_k\}$. We will construct an instance I of 2-TSP for $\mathcal{A}$ that depends on some natural number $r > 1/\varepsilon$ (cf. Figure 4). We start by creating a copy $V' = \{s', t', v_1', \ldots, v_k'\}$ of V and denote by $v' \in V'$ the copy of $v \in V$. Furthermore, we create "bridges" from s to s' and from t to t' using $r-1$ additional vertices each, which will be called $B^s = \{s = b_0^s, b_1^s, \ldots, b_{r-1}^s, s' = b_r^s\}$ and $B^t = \{t = b_0^t, b_1^t, \ldots, b_{r-1}^t, t = b_r^t\}$. So the vertices of our 2-TSP instance are $V \cup V' \cup B^s \cup B^t$. The pseudometric distance function will be defined as follows. First, we define it directly for some of the edges:

- for $e \in (V\times V) \cup (V' \times V')$, we set $c(e) = (c'(e), 0)$
- for $e = (b_i^s, b_{i+1}^s)$ or $e = (b_i^t, b_{i+1}^t)$, we set $c(e) = (0, 1/r)$

For all other vertex pairs and for each component, we indirectly define the distance as the length of the shortest path between these vertices using only edges from the above two categories.

In order to show that the functions c_1 and c_2 are pseudometric, we have to show that the directly defined distance between any two vertices is not longer than any path between them that uses edges with directly defined distances. For c_2, this is obviously the case.

We now argue for c_1. Let $u, v \in V$ and consider a path between u and v. If the path does not use the bridges and V', then it cannot be shorter than $c'(u,v) = c_1(u,v)$, since c' is pseudometric on V. So let us assume that the path uses the bridges and V'; w.l.o.g. the s-bridge is used first. So the length of the path is at least

$$\begin{aligned} c_1(u,s) + c_1(s,s') + c_1(s',t') + c_1(t',t) + c_1(t,v) &= c'(u,s) + 0 + c'(s,t) + 0 + c'(t,v) \\ &\geq c'(u,v) = c_1(u,v). \end{aligned}$$

The case where $u, v \in V'$ is of course symmetric and this property obviously holds for bridge edges, since they have distance 0. Hence c_1 is pseudometric.

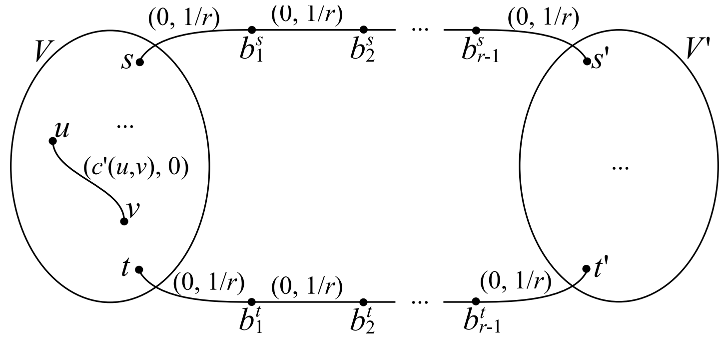

Figure 4: Creating an instance of 2-TSP from an instance (V, c', s, t) of TSPP_{st}. We first make a copy V' of V and, for each $u, v \in V$, we set $c(u, v) = (c'(u, v), 0)$ and $c(u', v') = (c'(u, v), 0)$. We further connect s with s' and t with t' by $r-1$ bridge vertices b_i^s, b_i^t for $1 \le i \le r-1$, and distribute the distance of $c(s, s') = c(t, t') = (0, r)$ equally among the bridge edges.

Let P be the c'-shortest Hamiltonian path between s and t in V and P' its (reversed) copy in V'. $P \cup \{(t, b_1^t), \ldots, (b_{r-1}^t, t')\} \cup P' \cup \{(s', b_{r-1}^s), \ldots, (b_1^s, s)\}$ is obviously a Hamiltonian cycle in the new graph with costs $(2c'(P), 2)$. Since it is a valid solution, $\mathcal{A}$ must return an $(\alpha, 2-\varepsilon)$-approximation of it. So $\mathcal{A}$ must return a solution S such that $c_2(S) \le 4 - 2\varepsilon$. We will now show that from S we can extract a Hamiltonian s-t-Path (in V) with length of at most $\alpha \cdot c'(P)$.

Let $E_{B^t} := \{(b_{i-1}^t, b_i^t) \mid 1 \le i \le r\} \cup \{(b_i^t, b_{i-1}^t) \mid 1 \le i \le r\}$ be the "simple" edges of the t-bridge and E_{B^s} be the analogously defined "simple" edges of the s-bridge. We can modify S such that edges crossing the set boundaries of V, V', B^t and B^s are replaced by a detour via the corresponding "portal" s, t, s', or t', possibly using a bridge. In other words, we only allow edges from the set $(V \times V) \cup (V' \times V') \cup E_{B^s} \cup E_{B^t}$. This modification does not raise any costs, as the costs for edges crossing these boundaries are in fact defined by taking detours via the portals. Hence, from now on we may assume that S satisfies the condition $S \subseteq (V \times V) \cup (V' \times V') \cup E_{B^s} \cup E_{B^t}$.

We will now argue that S uses each bridge exactly once. We denote by $u(x, y)$ the number of times the (undirected) edge (x, y) is used in S and by $d(v)$ the degree of a vertex v in S considered as a multi-graph. Furthermore, $d(V) = u(s, b_1^s) + u(t, b_1^t)$ and $d(V') = u(b_{r-1}^s, s') + u(b_{r-1}^t, t')$ are the "degrees" of the subgraphs V and V'.

Claim 5.3 *The degrees $d(v)$ for every vertex v and $d(V)$ and $d(V')$ are all even.*

Proof This holds because S is a Hamiltonian circuit. □

Claim 5.4 *The parity of $u(e)$ is the same for all edges $e \in E_{B^s} \cup E_{B^t}$.*

Proof We first show that for $x \in \{s, t\}$ the parity of $u(e)$ is the same for all edges $e \in E_{B^x}$.

Assume that the parity is not the same for all edges on one bridge. Then there is a vertex b_i^x with adjacent edges e_1 and e_2 such that $u(e_1)$ is odd and $u(e_2)$ is even. In this case, $d(b_i^x)$ must be odd which contradicts Claim 5.3.

Assume now that the parity is the same for each edge of the same bridge but different on the two bridges. Then $d(V)$ must be odd, which contradicts Claim 5.3. $\square$

Claim 5.5 *There can be at most one edge $e \in E_{B^s} \cup E_{B^t}$ such that $u(e) = 0$.*

Proof If there were two such edges, S would consist of (at least) two unconnected cycles. $\square$

Claim 5.6 *All bridge edges $e \in E_{B^s} \cup E_{B^t}$ have odd usage count $u(e)$ and for each bridge there exists at least one edge with usage count 1.*

Proof Assume that all bridge edges $e \in E_{B^s} \cup E_{B^t}$ have even usage count $u(e)$. This means that $u(e) \geq 2$ for all edges with at most one exception (Claim 5.5) and thus

$$c_2(S) = \sum_{e \in E_{B^s} \cup E_{B^t}} \frac{1}{r} u(e) \geq (2r-1) \cdot \frac{1}{r} \cdot 2 + 0 = 4 - 2\frac{1}{r} > 4 - 2\varepsilon \qquad (\text{since } r > \frac{1}{\varepsilon})$$

which contradicts the approximation ratio of $\mathcal{A}$.

Let us now assume that all bridge edges $e \in E_{B^s} \cup E_{B^t}$ have odd usage count $u(e)$ and for at most one bridge there exist edges with usage count 1. So the edges of one bridge have usage count of at least 1 and the edges of the other bridge have usage count of at least 3. Similarly to the case above, we obtain $c_2(S) \geq r \cdot \frac{1}{r} \cdot 1 + r \cdot \frac{1}{r} \cdot 3 = 4$ which contradicts the approximation ratio of $\mathcal{A}$. $\square$

We will now modify the solution S such that the usage count is 1 for all bridge edges. Let e, e' be two neighboring bridge edges such that $u(e) = 1$ and $u(e') \geq 3$. Since e is oriented in S, the oriented cycle must "turn back" on e', so we can remove these two turn-back-edges. Repeat this until all bridge edges have usage count 1.

So we may assume S to be of the form such that every bridge edge is used exactly once. This means that S starts at s, visits every vertex in V, goes to t, uses the bridge to t', visits every vertex in V', goes to s' and uses the bridge back to s. So $S \cap V \times V$ is a Hamiltonian path from s to t and another one can be obtained from $S \cap V' \times V'$. We can thus extract a Hamiltonian path with length at most $\frac{1}{2} c_1(S) \leq \frac{1}{2} \cdot 2 \cdot c'(P) \cdot \alpha = c'(P) \cdot \alpha$. This is an α-approximation for the $\mathrm{TSPP_{st}}$-instance (V, c', s, t). $\square$

Corollary 5.7 *Let $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* 2-TSP *is $(5/3 - \varepsilon, 2 - \varepsilon)$-approximable, then* $\mathrm{TSPP_{st}}$ *is $(5/3 - \varepsilon)$-approximable.*

5.2 Lower Bound Arguments for 2-TSPP

This section provides two approximation preserving reductions, one from $\mathrm{TSPP_{st}}$ to 2-TSPP, and another one from $\mathrm{TSPP_{s}}$ to 2-TSPP. Both reductions give evidence that the randomized approximation for 2-TSPP that is given in Theorem 4.1 is difficult to improve.

More precisely, an ε-improvement in the first component (i.e., $(3/2+\varepsilon, 5/3+\varepsilon) \to (3/2-\varepsilon, 5/3+\varepsilon)$) would result in a randomized approximation that improves the $3/2$-approximations for TSPP and $\mathrm{TSPP_{s}}$ which are known by Hoogeveen [Hoo91]. An ε-improvement in *both* components (i.e., $(3/2+\varepsilon, 5/3+\varepsilon) \to (3/2-\varepsilon, 5/3-\varepsilon)$) would result in a randomized approximation that improves the $5/3$-approximation for $\mathrm{TSPP_{st}}$ [Hoo91]. Both results give evidence for the difficulty of improving Theorem 4.1.

Again we start with the observation that an approximation algorithm for a two-criteria problem also approximates the underlying single-criterion problem.

Proposition 5.8 *Let $\alpha > 1$ and $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* 2-TSPP *is $(3/2 - \varepsilon, \alpha)$-approximable, then* TSPP *is $(3/2 - \varepsilon)$-approximable.*

Theorem 5.9 *Let $\alpha > 1$ and $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* 2-TSPP *is $(\alpha, 3/2 - \varepsilon)$-approximable, then* $\mathrm{TSPP_{st}}$ *is α-approximable.*

Proof We proceed analogously to the proof of Theorem 5.2 and reduce $\mathrm{TSPP_{st}}$ to 2-TSPP. Let therefore $\mathcal{A}$ be an algorithm that on input of a finite set V and a pseudometric (distance) function $c\colon V \times V \to \mathbb{N}^2$ returns an $(\alpha, 3/2 - \varepsilon)$-approximation for 2-TSPP for some $\alpha > 1$ and some $\varepsilon > 0$, and let furthermore $I = (V, c', s, t)$ be a $\mathrm{TSPP_{st}}$-instance where $V = \{s, t, v_1, \ldots, v_k\}$.

For each $e \in V \times V$, let $c_1(e) = c'(e)$. We define $c_2\colon V \times V \mapsto \mathbb{N}$ as follows:

$$\begin{aligned} c_2(u,v) &= 0 && \text{for } u,v \in V \setminus \{s,t\} \\ c_2(s,u) &= 1 && \text{for } u \in V \setminus \{s,t\} \\ c_2(t,u) &= 1 && \text{for } u \in V \setminus \{s,t\} \\ c_2(s,t) &= 2 \end{aligned}$$

Both c_1 and c_2 are pseudometric functions on V. Let $c = (c_1, c_2)$ and define a 2-TSPP instance as $I' = (V, c)$.

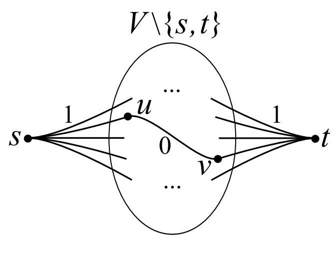

Figure 5: Structure of distance function c_2. For any $u \in V \setminus \{s,t\}$, we have $c_2(s,u) = c_2(t,u) = 1$. Inside $V \setminus \{s,t\}$, we have zero c_2 distances, hence $c_2(u,v) = 0$ for all $u, v \in V \setminus \{s,t\}$.

Figure 5 shows the structure of the distance function c_2. Obviously, all Hamiltonian paths $y \subseteq V \times V$ between s and t have length $c_2(y) = 2$, whereas all other Hamiltonian paths $y' \subseteq V \times V$ must have length $c_2(y') \geq 3$.

Let y^* be an optimal Hamiltonian path between s and t with respect to $c' = c_1$. Since y^* is a Hamiltonian path between s and t we have $c_2(y^*) = 2$. The approximate Pareto set provided by $\mathcal{A}(I')$ contains an approximate solution y' of the Hamiltonian path y^* such that $c_2(y') \leq (\frac{3}{2} - \varepsilon)c_2(y^*)$ and $c_1(y') \leq \alpha c_1(y^*)$. Hence $3 > c_2(y') = 2$ and therefore, y' is in fact a Hamiltonian path between s and t. This means that y' is an α-approximation of the optimal Hamiltonian path y^* between s and t with respect to $c' = c_1$. □

Corollary 5.10 *Let $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* 2-TSPP *is* $(3/2 - \varepsilon, 5/3 - \varepsilon)$*-approximable, then* TSPP_{st} *is* $(5/3 - \varepsilon)$*-approximable.*

Theorem 5.11 *Let $\alpha > 1$ and $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* 2-TSPP *is* $(\alpha, 2 - \varepsilon)$*-approximable, then* TSPP_{s} *is α-approximable.*

Proof As in the other cases, we reduce TSPP_{s} to 2-TSPP. Let again $\mathcal{A}$ be an algorithm that on input of a finite set V and a pseudometric (distance) function $c \colon V \times V \to \mathbb{N}^2$ returns an $(\alpha, 2 - \varepsilon)$-approximation for 2-TSPP for some $\alpha > 1$ and some $\varepsilon > 0$, and let $x = (V, c', s)$ be the TSPP_{s}-instance, where $V = \{s, v_1, \ldots, v_k\}$.

For each $e \in V \times V$, let $c_1(e) = c'(e)$. We define $c_2 \colon V \times V \mapsto \mathbb{N}$ as follows:

$$\begin{aligned} c_2(u, v) &= 0 && \text{for } u, v \in V \setminus \{s\} \\ c_2(s, u) &= 1 && \text{for } u \in V \setminus \{s\} \end{aligned}$$

Again, c_1 and c_2 are pseudometric functions on $V \times V$. We let $c = (c_1, c_2)$ and define a 2-TSPP instance as $x' = (V, c)$.

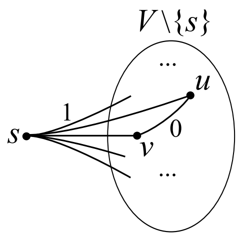

Figure 6: Structure of distance function c_2 of x'. For all $u \in V \setminus \{s\}$, we set $c_2(s, u) = 1$. All c_2 distances inside $V \setminus \{s\}$ are zero: $c_2(u, v) = 0$ for all $u, v \in V \setminus \{s\}$.

Figure 6 shows the structure of the distance function c_2. This time, all Hamiltonian paths $y \subseteq V \times V$ with endpoint s have length $c_2(y) = 1$, whereas all other Hamiltonian paths $y' \subseteq V \times V$ must have length of $c_2(y') = 2$.

Let y^* be an optimal Hamiltonian path of x. Since y^* has endpoint s, we have $c_2(y^*) = 1$. Then, the approximate Pareto set provided by $\mathcal{A}(x')$ contains an approximate solution y' of y^* such that $c_2(y') \leq (2 - \varepsilon)c_2(y^*)$ and $c_1(y') \leq \alpha c_1(y^*)$. Hence $2 > c_2(y') = 1$ and therefore, y' is a Hamiltonian path with endpoint s. This means that y' is an α-approximation of the optimal Hamiltonian path of x with endpoint s. □

Corollary 5.12 *Let $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* 2-TSPP *is* $(3/2 - \varepsilon, 2 - \varepsilon)$*-approximable, then* TSPP_{s} *is* $(3/2 - \varepsilon)$*-approximable.*

5.3 Lower Bound Arguments for 2-TSPP$_\mathrm{s}$

Below we construct an approximation preserving reduction from TSPP$_\mathrm{st}$ to 2-TSPP$_\mathrm{s}$, and a similar reduction from TSP to 2-TSPP$_\mathrm{s}$. This gives evidence for the difficulty of improving the randomized approximability of 2-TSPP$_\mathrm{s}$ that is given in Theorem 4.3.

More precisely, an ε-improvement in the first component (i.e., $(3/2+\varepsilon, 2+\varepsilon) \to (3/2-\varepsilon, 2+\varepsilon)$) gives a randomized approximation that improves the $3/2$-approximation for TSPP$_\mathrm{s}$ which is known by Hoogeveen [Hoo91]. An ε-improvement in the second component (i.e., $(3/2+\varepsilon, 2+\varepsilon) \to (3/2+\varepsilon, 2-\varepsilon)$) gives a randomized approximation that *considerably* improves the $5/3$-approximation for TSPP$_\mathrm{st}$ [Hoo91]. An ε-improvement in *both* components (i.e., $(3/2+\varepsilon, 2+\varepsilon) \to (3/2-\varepsilon, 2-\varepsilon)$) additionally yields a randomized approximation that even improves the $\frac{3}{2}$-approximation for TSP which is known by Christofides [Chr76]. These results give evidence for the difficulty of improving Theorem 4.3.

Again we start with the observation that an approximation algorithm for a two-criteria problem also approximates the underlying single-criterion problem.

Proposition 5.13 *Let $\alpha > 1$ and $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* 2-TSPP$_\mathrm{s}$ *is $(3/2-\varepsilon, \alpha)$-approximable, then* TSPP$_\mathrm{s}$ *is $(3/2-\varepsilon)$-approximable.*

Theorem 5.14 *Let $\alpha > 1$ and $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* 2-TSPP$_\mathrm{s}$ *is $(\alpha, 2-\varepsilon)$-approximable, then* TSPP$_\mathrm{st}$ *is α-approximable.*

Proof We show both assertions at the same time by reducing TSPP$_\mathrm{st}$ to 2-TSPP$_\mathrm{s}$. Let $\mathcal{A}$ be a (randomized) algorithm that $(\alpha, 2-\varepsilon)$-approximates 2-TSPP$_\mathrm{s}$ for some $\alpha > 1$ and some $\varepsilon > 0$. Let (V, c, s, t) be an arbitrary TSPP$_\mathrm{st}$-instance where $V = \{s, t, v_1, \ldots, v_k\}$. We construct a 2-TSPP$_\mathrm{s}$ instance I for $\mathcal{A}$ by adding a second distance function that places t "far away" from all other vertices and thus enforces that the path computed by $\mathcal{A}$ ends in t (cf. Figure 7). More precisely, $I = (V, c', s)$ where $c' = (c, c_2)$ and $c_2(x, y) = 0$ for all $x, y \neq t$ and $c_2(x, t) = 1$ for all $x \in \{s, v_1, \ldots, v_k\}$.

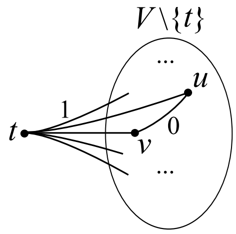

Figure 7: Structure of distance function c_2 of I. For all $u \in V \setminus \{t\}$, we set $c_2(t, u) = 1$. All c_2 distances inside $V \setminus \{t\}$ are zero: $c_2(u, v) = 0$ for all $u, v \in V \setminus \{t\}$.

We argue that $\mathcal{A}(I)$ computes an α-approximation for the TSPP$_\mathrm{st}$-instance (V, c, s, t). Let $P \subseteq V \times V$ be an optimal s-t-path with respect to c. Then P is also a valid solution of the TSPP$_\mathrm{s}$-instance I. This means that $\mathcal{A}(I)$ must return an $(\alpha, 2-\varepsilon)$-approximation A of P. Since $c_2(P) = 1$, we must have $c_2(A) \leq (2-\varepsilon)c_2(P) = 2-\varepsilon < 2$ and thus only one edge incident to t can be used by A, because all edges to t have length 1. So t must be the end point of A and s must be the starting point (or vice-versa) and we have $c(A) = c_1(A) \leq \alpha c(P)$. This means that A is an α-approximation of P. $\square$

Proposition 5.15 *Let $\alpha > 1$. The following holds for deterministic/randomized approximations: If* $\mathrm{TSPP_{st}}$ *is α-approximable, then* TSP *is α-approximable.*

Proof Assume that $\mathrm{TSPP_{st}}$ is (randomized) α-approximable. The following (randomized) algorithm α-approximates TSP.

Let $I = (V, c)$ be a given TSP-instance where $V = \{v_1, \ldots, v_m\}$. For all $t \in V \backslash \{v_1\}$, approximate an optimal Hamiltonian path between v_1 and t, and add the edge (t, v_1) to this tour. Finally, under all tours obtained in this way, choose the shortest one.

Observe that a suitable choice of t (e.g., t = successor of v_1 in an optimal Hamiltonian tour) yields an α-approximation of an optimal tour. □

Corollary 5.16 *Let $\varepsilon > 0$. The following holds for deterministic/randomized approximations:*

1. *If* $2\text{-}\mathrm{TSPP_s}$ *is $(3/2 + \varepsilon, 2 - \varepsilon)$-approximable, then* $\mathrm{TSPP_{st}}$ *is $(3/2 + \varepsilon)$-approximable.*
2. *If* $2\text{-}\mathrm{TSPP_s}$ *is $(3/2 - \varepsilon, 2 - \varepsilon)$-approximable, then* $\mathrm{TSPP_{st}}$ *and* TSP *are $(3/2 - \varepsilon)$-approximable.*

Proof Follows from Theorem 5.14 and Proposition 5.15. □

5.4 Lower Bound Arguments for 2-$\mathrm{TSPP_{st}}$

Regarding lower bounds for $2\text{-}\mathrm{TSPP_{st}}$ we only have the weak argument that an approximation algorithm for the two-criteria problem also approximates the underlying single-criterion problem.

Proposition 5.17 *Let $\alpha > 1$ and $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* $2\text{-}\mathrm{TSPP_{st}}$ *is $(5/3 - \varepsilon, \alpha)$-approximable, then* $\mathrm{TSPP_{st}}$ *is $(5/3 - \varepsilon)$-approximable.*

6 Open Questions

The results in the previous sections raise the following questions:

1. By Theorem 3.4 and 3.5, 2-TSP is randomized $(3/2, 2 + \varepsilon)$-approximable and $(3/2 + \varepsilon, 2)$-approximable. By Proposition 5.1 and Corollary 5.7, it is difficult to improve these approximations with respect to any component. It is even difficult to obtain a $(5/3 - \varepsilon, 2 - \varepsilon)$-approximation. However, so far there is no evidence in favor of or against an (α, β)-approximation where $5/3 \leq \alpha, \beta < 2$. Can one find such an approximation for 2-TSP? Or can one find evidence for the difficulty of such an improvement?

2. By Theorem 4.1, 2-TSPP is randomized $(3/2 + \varepsilon, 5/3 + \varepsilon)$-approximable, and by Corollary 5.12, it is difficult to obtain an ε-improvement in the first component. However, up to now we have no evidence for the difficulty of improving the second component. So from this point of view, there is no argument against a randomized $(3/2 + \varepsilon, \alpha)$-approximation for 2-TSPP where $3/2 \leq \alpha \leq 5/3$. Can one find such an approximation for 2-TSPP? Or can one find evidence for the difficulty of such an improvement?

   Similarly, by Theorem 4.5, 2-TSPP$_{\mathrm{st}}$ is $(2+\varepsilon, 2+\varepsilon)$-approximable, and by Proposition 5.17, it is difficult to obtain a randomized $(5/3 - \varepsilon, 2 + \varepsilon)$-approximation. Can one find a randomized $(\alpha, 2 + \varepsilon)$-approximation for 2-TSPP where $5/3 \leq \alpha \leq 2$? Or can one find evidence for the difficulty of such an improvement?

3. In section 5, we gave the following reductions that were used to translate approximations from one to another optimization problem: TSPP$_{\mathrm{st}}$ $\leq$ 2-TSP, TSPP$_{\mathrm{st}}$ $\leq$ 2-TSPP, TSPP$_{\mathrm{s}}$ $\leq$ 2-TSPP, TSPP$_{\mathrm{st}}$ $\leq$ 2-TSPP$_{\mathrm{s}}$, and TSP $\leq$ TSPP$_{\mathrm{st}}$. Can one find nontrivial reductions between the single-criterion problems TSP, TSPP, TSPP$_{\mathrm{s}}$, and TSPP$_{\mathrm{st}}$? For instance, does the existence of a $(5/3 - \varepsilon)$-approximation for TSPP$_{\mathrm{st}}$ imply a $(3/2 - \varepsilon)$-approximation for TSP? Conversely, does the existence of a $(3/2 - \varepsilon)$-approximation for TSP imply a $(5/3 - \varepsilon)$-approximation for TSPP$_{\mathrm{st}}$? Such translations of the approximability between the single-criterion problems would give a better understanding of the difficulty of these problems.

Acknowledgements. The authors thank Heinz Schmitz for valuable discussions on multi-criteria optimization and in particular on multi-criteria traveling salesman problems.

References

[ABG04] E. Angel, E. Bampis, and L. Gourvès. Approximating the Pareto curve with local search for the bicriteria TSP(1, 2) problem. *Theoretical Computer Science*, 310(1-3):135–146, 2004.

[ABGM05] E. Angel, E. Bampis, L. Gourvès, and J. Monnot. (Non-)approximability for the multi-criteria TSP(1,2). In *Fundamentals of Computation Theory*, volume 3623 of *Lecture Notes in Computer Science*, pages 329–340. Springer Berlin / Heidelberg, 2005.

[Aro98] S. Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782, 1998.

[Chr76] N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA, 1976.

[EG00] M. Ehrgott and X. Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spectrum*, 22(4):425–460, 2000.

[Ehr00] M. Ehrgott. Approximation algorithms for combinatorial multicriteria optimization problems. *International Transactions in Operational Research*, 7:5–31, 2000.

[Ehr05] M. Ehrgott. *Multicriteria Optimization.* Springer Verlag, 2005.

[GW86] A. Gupta and A. Warburton. Approximation methods for multiple criteria traveling salesman problems, towards interactive and intelligent decision support systems. In *Proceedings of 7th International Conference on Multiple Criteria Decision Making*, pages 211–217. Springer, 1986.

[Hoo91] J. A. Hoogeveen. Analysis of Christofides' heuristic: Some paths are more difficult than cycles. *Operations Research Letters*, 10:291–295, 1991.

[JP85] D. S. Johnson and C. H. Papadimitriou. Performance guarantees for heuristics. In E. L. Lawler, Jan Karel Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors, *The Traveling Salesman Problem. A Guided Tour of Combinatorial Optimization*, chapter 5, pages 145–180. Wiley, 1985.

[Kar72] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[Man09] B. Manthey. On approximating multi-criteria tsp. In *Proceedings of 26th Annual Symposium on Theoretical Aspects of Computer Science*, volume 09001 of *Dagstuhl Seminar Proceedings*, pages 637–648. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2009.

[MR09] B. Manthey and L. S. Ram. Approximation algorithms for multi-criteria traveling salesman problems. *Algorithmica*, 53(1):69–88, 2009.

[PV06] C. Papadimitriou and S. Vempala. On the approximability of the traveling salesman problem. *Combinatorica*, 26(1):101–120, 2006.

[PY93] C. Papadimitriou and M. Yannakakis. The traveling salesman problem with distances one and two. *Mathematics of Operations Research*, 18(1):1–11, 1993.

[PY00] C. H. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *FOCS '00: Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 86–95, Washington, DC, USA, 2000. IEEE Computer Society.

[Voi31] *Der Handlungsreisende, wie er sein soll, und was er zu thun hat, um Aufträge zu erhalten und eines glüklichen Erfolgs in seinen Geschäften gewiß zu sein. Von einem alten Commis Voyageur.* Bernhard Friedrich Voigt, Ilmenau, 1831.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de