Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 76 (2009)

Improved and Generalized Approximations for Two-Objective Traveling Salesman

Christian Glaßer Christian Reitwießner Maximilian Witek

Julius-Maximilians-Universität Würzburg, Germany
{glasser,reitwiessner,witek}@informatik.uni-wuerzburg.de

4th March 2010

Abstract

We propose a generalized definition for the multi-objective traveling salesman problem which uses multigraphs and which allows multiple visits of cities. The definition has two benefits: it captures typical real-world scenarios and it contains the conventional definition (componentwise metric cost function) as a special case.

We provide approximation algorithms for this general version of the two-objective traveling salesman problem (2-TSP). At the same time, with these algorithms we improve the best known approximations for the conventional case. For 2-TSP we obtain a deterministic 2-approximation, a randomized $(3/2+\varepsilon, 2)$-approximation, and a randomized $(3/2, 2+\varepsilon)$-approximation. Moreover, we construct similar algorithms for two-objective traveling salesman path problems.

Further we present 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). In this way, we can narrow down the approximation ratios for 2-TSP that could be within reach, i. e., that will not immediately improve well-studied approximations. 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]. In the 1950s and 1960s the traveling salesman problem became increasingly popular in mathematics and computer science.

In the original formulation, the salesman is not allowed to visit a city more than once. There are two arguments against this restriction: First, it does not make sense for the substantial majority of real-world traveling salesman problems, including all geometric versions [JP85]. Second, it considerably degrades the approximability of the problem. Therefore, with a minimum loss of generality, one often studies the traveling salesman problem where *multiple visits of cities are allowed.* This is

ISSN 1433-8092

equivalent to the *metric traveling salesman problem* (TSP), where one assumes a metric distance function. A special case of TSP is the *Euclidean* variant, where each city is located at some point in the plane, and the distance function is defined as the Euclidean distance of two cities.

In 1972, a breakthrough was achieved by Karp [Kar72] who proved the NP-hardness of TSP. This shows that the search for a polynomial-time algorithm for TSP is an extremely challenging endeavor and raises the question for good approximation algorithms. For a long time, the best known approximation for the metric and the Euclidean variant was the simple tree-doubling method. In 1976, Christofides [Chr76] improved this 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 metric traveling salesman problem.

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, TSPP_s, and 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 TSPP_s, and a $5/3$-approximation for TSPP_st.

Two-Objective TSP: In this paper we study the traveling salesman problem in the presence of *two* cost functions. For instance, we could be interested in tours that minimize both the transportation costs and the transportation time. Since these objectives are conflicting, we cannot hope for a single optimal solution, but there will be trade-offs. The *Pareto set* captures the notion of optimality in this setting. It consists of all solutions that are optimal in the sense that there is no solution that is strictly better. So for a given situation (i. e., cities and connections with transportation costs and transportation time), the Pareto set shows all optimal decisions (i. e., all optimal tours).

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

The multi-objective traveling salesman problem was first studied by Gupta and Warburton [GW86]. Angel, Bampis, and Gourvès [ABG04] give a $3/2$-approximation for the two-objective variant of TSP(1,2). Furthermore, Angel et al. [ABGM05] investigate the non-approximability of this problem. Ehrgott [Ehr00] studies the multi-objective traveling salesman problem and uses the l_1-norm (i.e., the sum of the components) to aggregate the cost functions into one. The approximation ratio obtained by this approach is incomparable to the two-component approximation ratios that we consider in our paper. Manthey and Ram [MR09] give a $(2+\varepsilon)$-approximation algorithm for multi-objective TSP with componentwise metric cost functions.

Our Contribution: *1. Generalized Realistic Models*
We propose a generalized definiton for the multi-objective traveling salesman problem that is based on multigraphs and that allows multiple visits of cities. It captures typical real-world scenarios and it contains as a special case the conventional definition, the componentwise metric multi-objective TSP [MR09, Ehr00]. The latter, however, is not generally applicable, since in many realistic situations the cost function is not componentwise metric.

2. Improved Approximation Algorithms
Even though we generalize the definition of the multi-objective TSP, we can provide more accurate approximations. For 2-TSP we obtain a deterministic 2-approximation, a randomized $(3/2+\varepsilon, 2)$-approximation, and a randomized $(3/2, 2+\varepsilon)$-approximation, where we build on the following known approximation schemes: An FPTAS for multi-objective minimum spanning tree, an FPTAS for multi-objective shortest path, and an FPRAS for multi-objective minimum matching. All three approximation schemes are known by Papadimitriou and Yannakakis [PY00], while an FPTAS for the two-objective shortest path is already known by Hansen [Han79]. In order to apply these algorithms, we have to extend them to multigraphs. So as a byproduct we provide approximation schemes for the multigraph variants of the mentioned problems.

The deterministic 2-approximation for 2-TSP is inspired by Christofides' approximation for TSP, but contains three new aspects. First, we have to work on multigraphs. Second, we cannot assume metric instances and therefore, we have to switch from matchings to *path matchings*. Third, since we aim at a deterministic algorithm, we cannot use the randomized approximation scheme for multi-objective minimum matching (or its extension to multi-objective minimum path matching). Instead, we deterministically extract a suitable two-objective path matching from an approximate two-objective minimum spanning tree. We show how to charge the error introduced by the FPTAS for two-objective minimum spanning tree against an error-reduction that is obtained by combining suitable pairs of spanning trees and path matchings. This results in the approximation ratio of 2.

If we allow randomness, then we can use an FPRAS for multi-objective minimum path matching. This computes the path matchings more accurately and results in the improved approximation ratios $(3/2+\varepsilon, 2)$ and $(3/2, 2+\varepsilon)$. Note that $3/2$ exactly meets the ratio of Christofides' approximation.

Manthey [Man09] notes that most approximation algorithms for multi-objective TSP use randomness for computing approximate Pareto sets of cycle covers (resp., matchings), and he raises the question of whether there are improved and derandomized algorithms for multi-objective TSP:

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

Our results give a positive answer to this question. With the deterministic 2-approximation for 2-TSP we slightly improve the deterministic $(2+\varepsilon)$-approximation for the componentwise metric 2-TSP [MR09]. Furthermore, the version of our algorithm for simple graphs 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 12 and the preceding remark).

Besides the approximations for 2-TSP, we further construct approximation algorithms for the two-objective traveling salesman path problems 2-TSPP, 2-$\mathrm{TSPP}_{\mathrm{s}}$, and 2-$\mathrm{TSPP}_{\mathrm{st}}$. Table 1 summarizes the obtained approximation ratios.

3. Lower Bound Arguments
We present arguments that indicate the hardness of improving our approximation algorithms. For this we demonstrate approximation preserving reductions that allow us to translate well-studied problems like TSP or $\mathrm{TSPP}_{\mathrm{st}}$ to the two-objective optimization problems 2-TSP, 2-TSPP, and 2-$\mathrm{TSPP}_{\mathrm{s}}$. From this we obtain that certain improvements of the approximation algorithms for 2-TSP, 2-TSPP, and 2-$\mathrm{TSPP}_{\mathrm{s}}$ force us to improve the best known approximation algorithms for TSP, $\mathrm{TSPP}_{\mathrm{s}}$, and $\mathrm{TSPP}_{\mathrm{st}}$ [Chr76, Hoo91]. Improvements of the latter approximations seem very difficult to obtain and hence improving our algorithms is difficult as well. Table 2 summarizes these arguments.

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.

Organization of the Paper: The preliminaries in section 2 give some basics on the concept of multi-objective optimization and define the problems studied here. Section 3 provides approximation schemes for the multigraph variants of multi-objective minimum spanning tree, multi-objective shortest path, and multi-objective minimum matching. Section 4 contains the deterministic and randomized approximation algorithms for 2-TSP, while section 5 provides algorithms for the traveling salesman path problems 2-TSPP, 2-$\mathrm{TSPP}_{\mathrm{s}}$, and 2-$\mathrm{TSPP}_{\mathrm{st}}$. In section 6 we argue that it is difficult to improve the approximation ratios of our algorithms. Finally, in section 7 we summarize the open questions of this paper.

2 Preliminaries

2.1 Multi-Objective Optimization

Consider some minimization problem with a k-dimensional cost function c with components c_i that maps solutions to $\mathbb{N}^k$. For solutions y, y' of some problem instance x we say y $(\alpha_1, \ldots, \alpha_k)$-*approximates* y' if $c_i(y) \leq \alpha_i \cdot c_i(y')$ for all i. We call a set of solutions $(\alpha_1, \ldots, \alpha_k)$-*approximate Pareto set* if every solution y' is $(\alpha_1, \ldots, \alpha_k)$-approximated by some y contained in the set. We say that some algorithm is an α-*approximation algorithm* if it returns an α-approximate Pareto set of x in polynomial time for all input instances x, and call it *randomized* if it does so with probability at least $1/2$ over all of its executions (however, in all cases we require its output to be a set of valid solutions of the input instance).

An algorithm is an FPTAS (fully polynomial-time approximation scheme) for a given optimization problem, if on input x and $\varepsilon > 0$ it computes a $(1+\varepsilon)$-approximate Pareto set of x in time polynomial in $|x| + 1/\varepsilon$. If the algorithm is randomized, then it is called FPRAS (fully polynomial-time randomized approximation scheme).

Problem	Det. Approx.	Randomized Approx.	Ref.
2-TSP	$(2,2)$	$(3/2+\varepsilon,2)$, $(3/2,2+\varepsilon)$	4.2, 4.4
2-TSPP	$(2+\varepsilon,2+\varepsilon)$	$(3/2+\varepsilon,5/3+\varepsilon)$	5.5, 5.2
2-TSPP$_\mathrm{s}$	$(2+\varepsilon,2+\varepsilon)$	$(3/2+\varepsilon,2+\varepsilon)$	5.5, 5.3
2-TSPP$_\mathrm{st}$	$(2+\varepsilon,2+\varepsilon)$	$(2+\varepsilon,2+\varepsilon)$	5.4

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

Problem	Approximation 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	6.1
	$(3/2+\varepsilon,2)$	$(5/3-\varepsilon,2-\varepsilon)$	$(5/3-\varepsilon)$	TSPP$_\mathrm{st}$	6.7
2-TSPP	$(3/2+\varepsilon,5/3+\varepsilon)$	$(3/2-\varepsilon,\alpha)$	$(3/2-\varepsilon)$	TSPP	6.8
		$(3/2-\varepsilon,2-\varepsilon)$	$(3/2-\varepsilon)$	TSPP$_\mathrm{s}$	6.12
		$(3/2-\varepsilon,5/3-\varepsilon)$	$(5/3-\varepsilon)$	TSPP$_\mathrm{st}$	6.10
2-TSPP$_\mathrm{s}$	$(3/2+\varepsilon,2+\varepsilon)$	$(3/2-\varepsilon,\alpha)$	$(3/2-\varepsilon)$	TSPP$_\mathrm{s}$	6.13
		$(3/2+\varepsilon,2-\varepsilon)$	$(3/2+\varepsilon)$	TSPP$_\mathrm{st}$	6.16.1
		$(3/2-\varepsilon,2-\varepsilon)$	$(3/2-\varepsilon)$	TSP	6.16.2
2-TSPP$_\mathrm{st}$	$(2+\varepsilon,2+\varepsilon)$	$(5/3-\varepsilon,\alpha)$	$(5/3-\varepsilon)$	TSPP$_\mathrm{st}$	6.17

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-objective problem is improved, then this also improves the deterministic (resp., randomized) approximation ratio of a well-studied optimization problem. Each argument holds for *all* considered versions of the multi-objective problems: the general version, the metric version, and the componentwise metric version. For instance, if componentwise metric 2-TSPP$_\mathrm{s}$ is randomized $(3/2-\varepsilon,2-\varepsilon)$-approximable, then TSP is randomized $(3/2-\varepsilon)$-approximable.

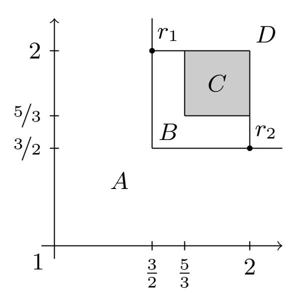

Figure 1: Approximation ratios for 2-TSP. An approximation ratio inside A would improve Christofides' approximation and a ratio inside B would improve Hoogeveen's approximation. We further prove approximation ratios r_1 and r_2, hence area D is of no further interest. However, evidence against approximation algorithms within C is not known.

2.2 Relevant Problems

A *multigraph* is a pair $G = (V, E)$ of finite sets such that $E \subseteq \{(\{u, v\}, i) \mid u, v \in V, i \in \mathbb{N}\}$. V is the set of vertices (or nodes) and E is the set of edges, where the last component of an edge is used to distinguish edges that connect the same pair of vertices. For example, the edges $e_1 = (\{u, v\}, 1)$ and $e_2 = (\{u, v\}, 2)$ are different objects that both connect the vertices u and v. If there are no different edges that connect the same vertices, we call G *simple*. For a given edge $e = (\{u, v\}, i)$, the *connection induced by* e is denoted by $[e] = \{u, v\}$. If e is an edge of a simple graph, we identify e with $[e]$. An $\mathbb{N}^k$*-labeled multigraph* ($\mathbb{N}^k$*-labeled simple graph*, resp.) is a triple $G = (V, E, c)$ where (V, E) is a multigraph (simple graph, resp.) and $c \colon E \to \mathbb{N}^k$. We also extend c in the obvious way to sets and multisets of edges. For any vertex $v \in V$, let $\deg_G(v)$ denote the degree of v in G, i. e., the number of edges incident to v.

Let a *walk (from v_0 to v_m)* be a sequence $v_0, e_1, v_1, \ldots, e_m, v_m$ of vertices and edges where e_i connects v_{i-1} and v_i. A *closed walk* is a walk with $v_0 = v_m$, a *spanning walk* is a walk that covers all vertices of G, and a *path* is a walk without repeated vertices. For any $U \subseteq V$ with $\#U = 2r$ we define a *path matching of U in G* as a set P of r paths in G such that every vertex in U is endpoint of exactly one path in P. For simplicity, we will represent walks as sets of edges.

We now define the main problems of this paper. Recall that we only consider minimization problems, so all costs have to be minimized.

Traveling Salesman (*k*-TSP)
Instance: $\mathbb{N}^k$-labeled multigraph (V, E, c)
Solution: closed spanning walk W
Costs: $c(W)$

We justify the terminology "k-TSP" as follows: In the case of 1-TSP, multiple edges between the same vertices do not make sense as they can be replaced by the edges with minimal weight. Therefore, 1-TSP is equivalent (w.r.t. approximation preserving reductions) to the single-objective TSP where multiple visits of cities are allowed. Similarly, 1-TSP is equivalent to the metric single-objective TSP which is the most commonly studied variant of TSP. So for a single objective, these natural variants of TSP are equivalent.

The situation changes when we consider TSP with multiple objectives. Here there exist at least three natural variants of different strengths. k-TSP is the most general variant, which handles arbitrary multigraphs and which allows multiple visits of cities. **Metric *k*-TSP** is the restriction of k-TSP where we require that if there is a path between two points in the multigraph, then there is also a direct edge that is at least as short in all components (i. e., a direct connection is always a shortest path between two points). Note that in this variant one can easily avoid multiple visits by taking a shortcut if a node was visited before. **Componentwise metric *k*-TSP** is the restriction of *metric* k-TSP where we require a simple graph that is componentwise metric, i. e., the triangle inequality holds with respect to each component. This variant was studied by Manthey and Ram [MR09], and here again it is easy to avoid multiple visits. Against the background of several variants of different strengths we use the notion k-TSP for the most general variant of the problem.

We define the *traveling salesman path problems* similar to k-TSP except that we do not require the walk to be closed (but rather to have fixed endpoints in $\{s, t\} \subseteq V$ if s, t are given). Regarding terminology, the same remarks apply as in the case of k-TSP. In particular, there exist metric and componentwise metric restrictions of these problems.

Traveling Salesman Path ($\boldsymbol{k}$-TSPP)
Instance: $\mathbb{N}^k$-labeled multigraph (V, E, c)
Solution: spanning walk W
Costs: $c(W)$

Traveling Salesman Path with Start Vertex ($\boldsymbol{k}$-TSPP$_\mathbf{s}$)
Instance: $\mathbb{N}^k$-labeled multigraph (V, E, c), start vertex $s \in V$
Solution: spanning walk W starting at s
Costs: $c(W)$

Traveling Salesman Path with Start and End Vertex ($\boldsymbol{k}$-TSPP$_\mathbf{st}$)
Instance: $\mathbb{N}^k$-labeled multigraph (V, E, c), start vertex $s \in V$, end vertex $t \in V$
Solution: spanning walk W starting at s and ending at t
Costs: $c(W)$

Finally, we extend the minimum spanning tree (resp., minimum matching and shortest path) problem to $\mathbb{N}^k$-labeled multigraphs and introduce the problem of finding a minimum cost *path matching* for an input graph and some given subset of vertices with even cardinality.

Minimum Spanning Tree on Multigraphs (multigraph $\boldsymbol{k}$-MST)
Instance: $\mathbb{N}^k$-labeled multigraph (V, E, c)
Solution: spanning tree $T \subseteq E$
Costs: $c(T)$

Minimum Perfect Matching on Multigraphs (multigraph $\boldsymbol{k}$-MM)
Instance: $\mathbb{N}^k$-labeled multigraph (V, E, c)
Solution: perfect matching M of V in (V, E)
Costs: $c(M)$

Minimum Path Matching on Multigraphs (multigraph $\boldsymbol{k}$-MPM)
Instance: $\mathbb{N}^k$-labeled multigraph (V, E, c), set $U \subseteq V$ of even cardinality (vertices to match)
Solution: path matching M of U in (V, E)
Costs: $c(M)$

Shortest Path on Multigraphs (multigraph $\boldsymbol{k}$-SP)
Instance: $\mathbb{N}^k$-labeled multigraph (V, E, c), start and end vertices $s, t \in V$
Solution: path P from s to t in (V, E)
Costs: $c(P)$

3 Matching and Spanning Tree Algorithms on Multigraphs

It is known that the multi-objective variants of minimum spanning tree (**$\boldsymbol{k}$-MST**), shortest path (**$\boldsymbol{k}$-SP**), and minimum matching (**$\boldsymbol{k}$-MM**) are NP-hard [PY82, GJ79, PY00]. We need approximation algorithms for the multigraph variants of these problems. For this we extend known approximation schemes on simple graphs such that they work for multigraphs. In a second step we extend the approximation scheme for multi-objective minimum matching on multigraphs such that it works for multi-objective minimum path matching on multigraphs.

Theorem 3.1 ([PY00]) *For any $k \geq 1$ there is an FPTAS for k-*MST*, an FPRAS for k-*MM*, and an FPTAS for k-*SP*.*

Theorem 3.2 *For any $k \geq 1$ there is an FPTAS for* multigraph k-MST*, an FPRAS for* multigraph k-MM*, and an FPTAS for* multigraph k-SP*.*

Proof

1. multigraph k-MST: We reduce the problem for $\mathbb{N}^k$-labeled multigraphs to the case of $\mathbb{N}^k$-labeled simple graphs, where the existence of an FPTAS is known by Theorem 3.1.

Let $G = (V, E, c)$ be an $\mathbb{N}^k$-labeled multigraph. We transform G to a simple graph G' by splitting each edge into three parts. If an edge $e \in E$ connects the vertices u and v, then we add two new vertices u_e and v_e to the graph and replace e by the three edges $f(e) = \{\{u, u_e\}, \{u_e, v_e\}, \{v_e, v\}\}$. Furthermore, the edge in the middle $\{u_e, v_e\}$ is labeled with $c(e)$, while the remaining two edges are labeled with $(0, \ldots, 0)$.

Formally, $G' = (V \cup U, E', c')$ where $U = \{v_e \mid e \in E, v \in [e]\}$, $E' = \bigcup_{e \in E} f(e)$, and $c' \colon E' \to \mathbb{N}^k$ such that

$$c'(e') = \begin{cases} c(e) & \text{if } [e'] = \{u_e, v_e\} \text{ for some } u, v \in V, e \in E \text{ and} \\ (0, \ldots, 0) & \text{if } [e'] \nsubseteq U. \end{cases}$$

We extend f to subsets $E_1 \subseteq E$:

$$f(E_1) = \bigcup_{e \in E_1} f(e)$$

So f translates subsets $E_1 \subseteq E$ into subsets $E'_1 \subseteq E'$. For the converse translation, let

$$g(E'_1) = \{e \in E \mid \{u_e, v_e\} \in E'_1 \text{ for some } u, v \in V\}$$

for $E'_1 \subseteq E'$. Observe that f and g respect the sum of the labels, i. e.,

$$c(E_1) = c'(f(E_1)) \quad \text{and} \quad c'(E'_1) = c(g(E'_1)). \tag{1}$$

Each path in some $E'_1 \subseteq E'$ that starts and ends in nodes from V induces a path in $g(E'_1)$ with the same start and end nodes. Therefore,

$$E'_1 \text{ is connected and covers } V \quad \Rightarrow \quad g(E'_1) \text{ is connected and covers } V. \tag{2}$$

We describe the FPTAS for multigraph k-MST on input $G = (V, E, c)$ and $\varepsilon > 0$: Transform G into G' as described above and run the FPTAS by [PY00] on input G' and ε. We obtain a $(1+\varepsilon)$-approximation $\mathcal{A}$ of all minimal spanning trees of G'. For each $T' \in \mathcal{A}$, compute $g(T')$, prune this graph as long as it contains any cycles, and output the resulting tree.

The running time of the algorithm is polynomial in $|G|$ and $1/\varepsilon$.

Let T be a minimal spanning tree of $G = (V, E, c)$. We argue that the algorithm above outputs a $(1+\varepsilon)$-approximation of T. Observe that $f(T) \cup \{e' \in E' \mid [e'] \nsubseteq U\}$ is a spanning tree of G' with costs $c'(f(T)) = c(T)$. So $\mathcal{A}$ contains a spanning tree T' of G' such that $c'(T') \leq (1+\varepsilon) \cdot c(T)$. By (2), $g(T')$ is connected and by (1), $c(g(T')) = c'(T') \leq (1+\varepsilon) \cdot c(T)$. Hence the algorithm outputs a spanning tree of G with costs at most $(1+\varepsilon) \cdot c(T)$.

2. multigraph k-MM: We again reduce this problem to the case of minimal perfect matchings in an $\mathbb{N}^k$-labeled simple graph, where the existence of an FPTAS is known by Theorem 3.1. For this, let $G = (V, E, c)$ be an $\mathbb{N}^k$-labeled multigraph. We transform G to the same $\mathbb{N}^k$-labeled simple graph $G' = (V \cup U, E', c'')$ as in the first part, only the cost function c'' is defined differently. So recall U and E' from the first part. The label of the original edge $c(e)$ is put on both outer edges, while the edge in the middle is labeled with zero. More formally:

$$c''(e') = \begin{cases} c(e) & \text{if } [e'] = \{v, v_e\} \text{ for some } v \in V, e \in E \text{ and} \\ (0, \ldots, 0) & \text{if } [e'] \subseteq U \end{cases}$$

We also define a different converse translation h (which is not inverse to f):

$$h(E'_1) = \{e \in E \mid \{u_e, v_e\} \notin E'_1 \text{ for } u, v \in V \text{ with } \{u, v\} = [e]\}$$

for $E'_1 \subseteq E'$.

The FPRAS for multigraph k-MM on input $G = (V, E, c)$ and $\varepsilon > 0$ works as follows: Transform G into G' as described above and run the FPRAS by [PY00] on input G' and ε. We obtain a $(1+\varepsilon)$-approximation $\mathcal{A}$ of all minimal perfect matchings G'. For each $M' \in \mathcal{A}$, output $h(M')$.

The running time of the algorithm is polynomial in $|G|$ and $1/\varepsilon$. It remains to show that the returned edge sets are in fact perfect matchings and that they approximate the minimal perfect matchings of G with probability at least $1/2$.

For the first part, consider some perfect matching $M' \subseteq E'$ of G'. Observe that for any $e \in E$ with $[e] = \{u, v\}$, we have $\{u, u_e\}, \{v_e, v\} \in M' \iff \{u_e, v_e\} \notin M'$. So since any vertex $v \in V$ must be matched exactly once in M', there is exactly one $e \in E$ such that $\{v, v_e\} \in M'$. Since u is uniquely determined by $\{u, v\} = [e]$, we get that there is exactly one $u \in V$ such that $\{u, v\} \in h(M')$ and thus $h(M')$ is a perfect matching of G'.

For the second part, first note that for any perfect matching M' of G' we have

$$c''(M') = 2c(h(M')). \tag{3}$$

Now let M be a perfect matching of G and consider $M' = \{\{v, v_e\} \mid v \in [e], e \in M\} \cup \{\{u_e, v_e\} \mid \{u, v\} = [e], e \in E \setminus M\}$. Obviously, M' is a perfect matching of G' and $h(M') = M$. So we have $c''(M') = 2c(M)$, and the output of the FPRAS must contain some perfect matching $\tilde{M}'$ of G' such that $c''(\tilde{M}') \leq (1+\varepsilon)c''(M') = 2(1+\varepsilon)c(M)$ with probability at least $1/2$. From $\tilde{M}'$, we obtain a perfect matching $\tilde{M} = h(\tilde{M}')$ of G such that $c(\tilde{M}) = \frac{1}{2}c''(\tilde{M}') \leq (1+\varepsilon)c(M)$ and thus the assertion is proved.

3. k-SP: For multigraph shortest path, we use exactly the same construction as in part 1. So recall the notions from part 1. The algorithm works as follows:

On input of a multigraph (V, E, c), $s, t \in V$ and $\varepsilon > 0$, construct $G' = (V \cup U, E', c')$ as in part 1, run the FPTAS for k-SP on G', s, t and ε, apply g to the paths found by the FPTAS and return the results.

The algorithm obviously runs in polynomial time and solves the problem because of the properties of f, g and c already noted in part 1. □

The FPRAS for multigraph k-MPM that is stated in the following theorem is based on the approximation schemes for multigraph k-SP and multigraph k-MM.

Theorem 3.3 *For $k \geq 1$ there is an FPRAS for* multigraph k-MPM*.*

Proof Let $G = (V, E, c)$ be an $\mathbb{N}^k$-labeled multigraph, $U \subseteq V$ be a set of even cardinality and $\varepsilon > 0$.

We start with the construction of an $\mathbb{N}^k$-labeled multigraph $G' = (U, E', c')$ by approximately computing shortest paths between vertices of U and letting each path be an edge in G' between its endpoints. More formally: For every two-element subset $\{s, t\}$ of U, run the FPTAS for multigraph k-SP (cf. Theorem 3.2) on G, s, t and ε to obtain the approximate Pareto set $\{p_1^{\{s,t\}}, \ldots, p_{m_{\{s,t\}}}^{\{s,t\}}\}$ of shortest paths between s and t in G and let $E' = \{(\{s, t\}, i) \mid s, t \in U, s \neq t \text{ and } 1 \leq i \leq m_{\{s,t\}}\}$ and $c'(\{s, t\}, i) = c(p_i^{\{s,t\}})$ for $(\{s, t\}, i) \in E'$.

Now run the FPRAS for multigraph k-MM (cf. Theorem 3.2) on G' and ε and obtain an approximate Pareto set of matchings $\mathcal{M}$. Finally, for each perfect matching $M \in \mathcal{M}$, return $\{p_i^{\{s,t\}} \mid (\{s, t\}, i) \in M\}$.

The running time of the algorithm is polynomial in $|G|$ and $1/\varepsilon$, and the returned sets are path matchings, since every vertex $s \in U$ is matched by exactly one edge in M and thus by exactly one path. Concerning the approximation ratio, let P be a path matching of U in G. Every path $p \in P$ with endpoints $s, t \in U$ is approximated by the FPTAS for multigraph k-SP, which means that there is some $1 \leq i \leq m_{\{s,t\}}$ such that $c(p_i^{\{s,t\}}) \leq (1+\varepsilon)c(p)$. For $\tilde{P}$ being the set of these (approximately) shortest paths $p_i^{\{s,t\}}$ for all paths $p \in P$, we obtain $c(\tilde{P}) \leq (1 + \varepsilon)c(P)$. Furthermore, $\tilde{P}$ is a path matching and thus corresponds to a perfect matching M of G' with the same costs. This perfect matching is approximated by a perfect matching $\tilde{M}$ using the FPRAS for multigraph k-MM. For the path matching $\tilde{P}'$ finally obtained from $\tilde{M}$, we have the inequality

$$\begin{aligned} c(\tilde{P}') &= c'(\tilde{M}) \\ &\leq (1 + \varepsilon)c'(M) = (1 + \varepsilon)c(\tilde{P}) \\ &\leq (1 + \varepsilon)(1 + \varepsilon)c(P) = (1 + 2\varepsilon + \varepsilon^2)c(P) \\ &\leq (1 + 3\varepsilon)c(P) \end{aligned}$$

which means that the algorithm described above is an FPRAS for k-MPM. □

We denote the multigraph approximation schemes for $k = 2$ by `2-MST-Approx`(V, E, c, ε), `2-MM-Approx`$_{\mathtt{R}}(V, E, c, \varepsilon)$, `2-SP-Approx`$(V, E, c, s, t, \varepsilon)$, and `2-MPM-Approx`$_{\mathtt{R}}(V, E, c, U, \varepsilon)$, where

(V, E, c) is an $\mathbb{N}^k$-labeled multigraph, $s, t \in V$ are start and end vertices, $U \subseteq V$ are the vertices to match (even cardinality), and ε is the approximation factor. We will repeatedly call the approximations $\texttt{2-MM-Approx}_\texttt{R}$ and $\texttt{2-MPM-Approx}_\texttt{R}$ in our randomized algorithms that will follow. In each algorithm, we assume that these approximations are amplified in a way such that the probability that *all* calls succeed is at least $1/2$.

4 Approximation for 2-TSP

The best known approximation for a multi-objective traveling salesman problem is the deterministic $(2+\varepsilon)$-approximation for the componentwise metric 2-TSP that was given by Manthey and Ram [MR09]. In this section we present algorithms that improve this result in two ways:

1. The new algorithms manage more general and more realistic scenarios of the multi-objective traveling salesman problem. More precisely, we do not assume componentwise metric instances, but only allow multiple visits of cities, which is a much weaker assumption.
2. We improve the known approximation ratios. More precisely, for 2-TSP we give a deterministic 2-approximation and randomized approximations with ratios $(3/2+\varepsilon, 2)$ and $(3/2, 2+\varepsilon)$. In particular, the first component exactly meets the approximation ratio of Christofides' algorithm, which is still the best known approximation for the single-objective TSP.

4.1 Deterministic Approximation for 2-TSP

Our deterministic 2-approximation for 2-TSP is inspired by Christofides' approximation for TSP. In contrast to the single-objective problem, we cannot assume that the instances of 2-TSP are metric. Therefore, it does not suffice to compute perfect matchings for the odd degree vertices in the tree, but we have to switch to the notion of path matchings. The second major difference to Christofides' algorithm is the fact that we do not directly compute the two-objective minimal path matchings (e. g., by generalizing the randomized approximation algorithm for two-objective minimum matching [PY00]). Instead, we show that a suitable two-objective path matching can be extracted *deterministically* from an approximate two-objective minimum spanning tree. More precisely, we transform a spanning tree into a path matching of at most the same costs. Thus we avoid the randomness of the approximation algorithm for two-objective minimum path matching (2-MPM). Of course, we obtain a path matching that is by far not optimal. Nevertheless, this path matching suffices to improve the approximation, since at present, the bottleneck of approximations for 2-TSP is not the method of *finding* a good path matching, but is the argument that a good path matching *exists*.

Let V be a set of nodes, $U \subseteq V$ a set of even cardinality, and T a spanning tree on V. 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 path matching of U with costs less than or equal to $c(T)$, we take an arbitrary path matching M on U and consider any distinct paths $p, p' \in M$, where $p = \mathrm{p}_T(u, u')$ and $p' = \mathrm{p}_T(v, v')$ for some $u, u', v, v' \in U$. If p and p' intersect on at least one edge, we can easily remove this intersection by re-pairing p and p' in M (cf. Figure 2). We repeat this process until there are no more intersections in M. It follows that $c(M) \leq c(T)$.

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 : tree T and subset U of its vertices of even cardinality
Output: path matching P on U in T

let $M \subseteq U \times U$ **be some subdivision of** U **into pairs;**
$P := \{\mathrm{p}_T(u_1, u_2) \mid (u_1, u_2) \in M\}$;
while there are distinct but non-disjoint $\mathrm{p}_T(u, u'), \mathrm{p}_T(v, v') \in P$ **do**
  $P := P \setminus \{\mathrm{p}_T(u, u'), \mathrm{p}_T(v, v')\}$;
  if $\mathrm{p}_T(u, v) \cap \mathrm{p}_T(u', v') = \emptyset$ **then**
    $P := P \cup \{\mathrm{p}_T(u, v), \mathrm{p}_T(u', v')\}$
  else
    $P := P \cup \{\mathrm{p}_T(u, v'), \mathrm{p}_T(u', v)\}$ // note that $\mathrm{p}_T(u, v') \cap \mathrm{p}_T(u', v) = \emptyset$
  end
end
return P

Lemma 4.1 *Let $G = (V, E, c)$ be some $\mathbb{N}^k$-labeled multigraph for $k \geq 1$ and let $T \subseteq E$ be a spanning tree of G. Then, for any $U \subseteq V$ of even cardinality,* `match`(U, T) *finds in polynomial time a path matching M of U in T such that $c(M) \leq c(T)$.*

Proof Let m denote the number of edges of T, and $S(P, T) = \sum_{p \in P} \#p$ be the sum of the number of edges of all paths used in T for some path matching P. Note that at any time in the algorithm, the value of the variable P is a path matching on U in T. Clearly, $S(P, T) \leq m^2/2$, since there are at most m edges per path and at most $m/2$ distinct pairs of endpoints of paths. In every iteration, we redirect two paths, which reduces $S(P, T)$ by at least two. Hence, the algorithm terminates 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 time-bounded, we obtain a polynomial-time algorithm.

After the termination of the algorithm, any two distinct $\mathrm{p}_T(u, v), \mathrm{p}_T(u', v') \in P$ are completely disjoint. We can now estimate the overall costs of P by

$$c(P) = \sum_{p \in P} c(p) \leq c(T).$$

□

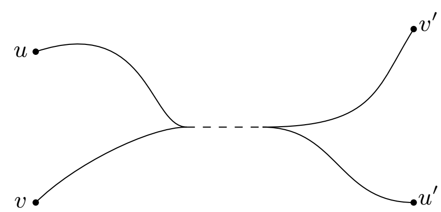

Figure 2: The paths $\mathrm{p}_T(u, u')$ and $\mathrm{p}_T(v, v')$ intersect on the dashed edges. In this setting, pairing u with v and u' with v' will remove the intersecting edges and thereby improve the costs of M.

We proceed with the following deterministic algorithm that on input of an $\mathbb{N}^2$-labeled multigraph (V, E, c) computes a $(2,2)$-approximation for 2-TSP. For this, recall the definition of the algorithm `2-MST-Approx` (in the text after Theorem 3.3).

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

Input : $\mathbb{N}^2$-labeled multigraph (V, E, c)
Output: set of closed spanning walks of (V, E, c)

$\varepsilon := \frac{1}{2\#V}$;
$\mathcal{S} := \emptyset$;
$\mathcal{T} :=$ `2-MST-Approx(`V, E, c, ε`)`;
`foreach` $(T_1, T_2) \in \mathcal{T} \times \mathcal{T}$ `do`
    $U := \{v \in V \mid \deg_{T_1}(v)$ `is odd`$\}$;
    $M :=$ `match(`U, T_2`)`;
    $W_{\text{approx}} :=$ `closed spanning walk of` (V, E) `using the edges of` T_1 `and` M;
    $\mathcal{S} := \mathcal{S} \cup \{W_{\text{approx}}\}$
`end`
`return` $\mathcal{S}$

Theorem 4.2 2-TSP *is* $(2,2)$*-approximable.*

Proof Let V be a finite set, E a finite set of edges for V, $c\colon E \to \mathbb{N}^2$ a function representing the costs and W an arbitrary closed spanning walk of the multigraph (V, E). We show that `2-TSP-ApproxDet(`V, E, c`)` contains a closed spanning walk W_{approx} such that $c(W_{\text{approx}}) \leq 2c(W)$.

As in the algorithm `2-TSP-ApproxDet`, let $m = \#V$ and $\varepsilon = \frac{1}{2m}$. We split W into contiguous subwalks $W_1, \ldots, W_m$ such that every vertex is at one end of at least one of the subwalks. This can be achieved by letting every subwalk start at the first occurrence of some vertex in W. For every $i \in \{1, 2\}$ there is some $1 \leq p_i \leq m$ such that $c_i(W_{p_i}) \geq \frac{1}{m} c_i(W)$. By removing W_{p_i} from W, the multiset of edges E_i thus obtained is connected and covers every vertex of V. Thus (V, E) has spanning trees T_i' with no higher costs than E_i, which means that

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

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

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

Let us now consider the iteration of the loop for exactly this pair of trees and let M and W_{approx} be as in the algorithm. The number of vertices of odd degree in an undirected graph is even, so $c(M) \leq c(T_2)$ by Lemma 4.1. W_{approx} can be easily constructed, since all vertices of odd degree in

T_1 are matched by edges in M and thus every vertex has even degree when the edges of T_1 and M are used. Concerning the costs we obtain

$$\begin{aligned} c\left(W_{\text{approx}}\right) &\leq c\left(T_1\right) + c\left(M\right) \\ &\leq c\left(T_1\right) + c\left(T_2\right) \\ &\leq \left(1 + \frac{1}{2m}\right)\left(c\left(T_1'\right) + c\left(T_2'\right)\right) \\ &\leq \left(1 + \frac{1}{2m}\right)\left(\left(1 - \frac{1}{m}\right) + 1\right) c\left(W\right) \\ &= \left(2 - \frac{1}{2m^2}\right) c\left(W\right) \\ &< 2c\left(W\right). \end{aligned}$$

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 $\mathcal{T}$ itself is bounded by a polynomial in m, say p. For each of the p^2 combinations of spanning trees, the steps 5–8 can be done in polynomial time (cf. Lemma 4.1). Hence `2-TSP-ApproxDet` is a polynomial-time algorithm. □

4.2 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 an enhanced variant of a randomized approximation for the componentwise metric 2-TSP that was studied by Manthey and Ram [MR09]. First, it computes approximations of the Pareto-minimal spanning trees, then considers the vertices that have odd degree in a single tree, computes approximations of the Pareto-minimal path matchings of these vertices, and finally pairwise combines all trees with all suitable matchings which results in a set of closed spanning walks. A precise analysis provides approximation ratios that are better than the ones stated in [MR09], even though the new algorithm manages a more general variant of the problem.

The algorithm below calls the algorithms `2-MST-Approx` and `2-MPM-Approx`$_{\texttt{R}}$, which were defined in Section 3. The use of the FPTAS for the two-objective 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 contiguous subwalk of an optimal walk. This makes it possible to remove an ε-error in one of the two objectives.

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

Input : $\mathbb{N}^2$-labeled multigraph (V, E, c)
Output: set of closed spanning walks of (V, E, c)

1 $m := \#V$; $\varepsilon_1 := \frac{\varepsilon}{m^2}$; $\varepsilon_2 := \frac{\varepsilon}{2m}$;
2 $\mathcal{S} := \emptyset$;
3 $\mathcal{P} :=$ 2-MST-Approx(V, E, c, ε_1);
4 foreach $T \in \mathcal{P}$ do
5     $U := \{v \in V \mid \deg_T(v)$ is odd$\}$;
6     $\mathcal{A} :=$ 2-MPM-Approx$_\mathtt{R}$$(V, E, c, U, \varepsilon_2)$;
7   foreach $M \in \mathcal{A}$ do
8         $W_{\text{approx}} :=$ closed spanning walk using the edges of T and M;
9         $\mathcal{S} := \mathcal{S} \cup \{W_{\text{approx}}\}$
10   end
11 end
12 return $\mathcal{S}$

Lemma 4.3 *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, E, c). So we can obtain $\mathcal{P}$ in polynomial time and $\mathcal{P}$ contains only polynomially many elements. This means that the first loop is iterated polynomially often. 2-MPM-Approx$_\mathtt{R}$ runs in polynomial time in the length of $(V, E, c, U) + \frac{1}{\varepsilon_2}$ and thus also in n. This in turn means that $\mathcal{A}$ contains only polynomially many matchings, so the second loop is iterated only polynomially often. The operation in line 8 can obviously be carried out in polynomial time and thus the whole algorithm runs in polynomial time. $\square$

Theorem 4.4 *For every $\varepsilon > 0$,* 2-TSP *is randomized $(3/2 + \varepsilon, 2)$-approximable and randomized $(3/2, 2 + \varepsilon)$-approximable.*

Proof We show that these approximations are realized by 2-TSP-ApproxRand$_\varepsilon$. By Lemma 4.3, the algorithm runs in polynomial time, so it remains to show the approximation ratio. 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 vertices in the input graph. We show that the algorithm 2-TSP-ApproxRand$_\varepsilon$ computes a $(3/2 + \varepsilon, 2)$-approximation and a $(3/2, 2 + \varepsilon)$-approximation for every 2-TSP instance (V, E, c) with probability at least $1/2$.

Let (V, E, c) be some $\mathbb{N}^2$-labeled multigraph and W be an arbitrary closed walk of (V, E).

We split W into contiguous subwalks $W_1, \ldots, W_m$ such that every vertex is at one end of at least one of the subwalks. This can be achieved by letting every subwalk start at the first occurrence of some vertex in W. Then, there is some $1 \leq r \leq m$ such that $c_2(W_r) \geq \frac{1}{m} c_2(W)$. By removing W_r from W, the multiset of edges E' thus obtained is connected and covers every vertex of V. Thus, (V, E, c) has a spanning tree T' with no higher costs than E', which means that

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

By $\varepsilon_1 = \frac{\varepsilon}{m^2}$, the algorithm `2-MST-Approx` finds a spanning tree T_1 with costs

$$c(T_1) \quad \leq \quad \left(1+\frac{\varepsilon}{m^2}\right)\left(c_1(W), \frac{m-1}{m} c_2(W)\right).$$

By a symmetric argumentation, `2-MST-Approx` also finds a spanning tree T_2 with costs

$$c\left(T_2\right) \quad \leq \quad \left(1+\frac{\varepsilon}{m^2}\right)\left(\frac{m-1}{m} c_1\left(W\right), c_2\left(W\right)\right).$$

Let $U_1 \subseteq V$ be the vertices of odd degree in T_1 and note that U_1 has even cardinality. From W we can easily find two path matchings M_1 and M_2 of U_1 in (V, E, c) such that $c(M_1) + c(M_2) \leq c(W)$: For each $u \in U_1$, fix the first occurrence in W, and cut W at each of those $\#U_1$ positions to obtain $\#U_1$ subwalks $S_1, \ldots, S_{\#U_1}$. For each i, we remove any cycles from S_i and obtain a path P_i with $c(P_i) \leq c(S_i)$. Then, both $M_1 = \{P_i \mid i \text{ even}\}$ and $M_2 = \{P_j \mid j \text{ odd}\}$ are path matchings of U_1 in (V, E, c). Hence, there is some path matching M' of U_1 in (V, E, c) such that $c(M') \leq (\frac{1}{2} c_1(W), c_2(W))$.

By $\varepsilon_2 = \frac{\varepsilon}{2m}$, the algorithm $\texttt{2-MPM-Approx}_{\texttt{R}}(V, E, c, U_1, \varepsilon_2)$ must return some approximate minimum path matching M_1 (with probability at least 1/2) such that

$$\begin{aligned} c\left(M_1\right) \quad &\leq \quad \left(1+\frac{\varepsilon}{2m}\right) c\left(M'\right) \\ &\leq \quad \left(1+\frac{\varepsilon}{2m}\right)\left(\frac{1}{2} c_1\left(W\right), c_2\left(W\right)\right). \end{aligned}$$

Again using symmetric arguments, $\texttt{2-MPM-Approx}_{\texttt{R}}(V, E, c, U_2, \varepsilon_2)$ must return some approximate minimum path matching M_2 (with probability at least 1/2) such that

$$c\left(M_2\right) \quad \leq \quad \left(1+\frac{\varepsilon}{2m}\right)\left(c_1\left(W\right), \frac{1}{2} c_2\left(W\right)\right).$$

By combining T_1 and M_1 we obtain a spanning walk $W_{\text{approx},1}$ such that the following holds (note that $m \geq 2$ and $\varepsilon \leq 1$):

$$\begin{aligned} c_1\left(W_{\text{approx},1}\right) \quad &\leq \quad c_1\left(T_1\right) + c_1\left(M_1\right) \\ &\leq \quad \left(1+\frac{\varepsilon}{m^2}\right) c_1\left(W\right) + \left(1+\frac{\varepsilon}{2m}\right) \frac{1}{2} c_1\left(W\right) \\ &= \quad \left(\frac{3}{2} + \varepsilon\left(\frac{1}{m^2} + \frac{1}{4m}\right)\right) c_1\left(W\right) \\ &\leq \quad \left(\frac{3}{2} + \varepsilon\right) c_1\left(W\right) \end{aligned}$$

and

$$\begin{aligned} c_2\left(W_{\text{approx},1}\right) \quad &\leq \quad c_2\left(T_1\right) + c_2\left(M_1\right) \\ &\leq \quad \left(\left(1+\frac{\varepsilon}{m^2}\right)\frac{m-1}{m} + \left(1+\frac{\varepsilon}{2m}\right)\right) c_2\left(W\right) \end{aligned}$$

$$
\begin{aligned}
&\leq \left(\left(1+\frac{1}{m^2}\right)\frac{m-1}{m}+1+\frac{1}{2m}\right)c_2\left(W\right)\\
&= \left(\frac{m-1}{m}+\frac{m-1}{m^3}+1+\frac{1}{2m}\right)c_2\left(W\right)\\
&= \left(2-\frac{1}{m}+\frac{m-1}{m^3}+\frac{1}{2m}\right)c_2\left(W\right)\\
&= \left(2+\frac{-m^2+m-1+\frac{1}{2}m^2}{m^3}\right)c_2\left(W\right)\\
&= \left(2+\frac{-\frac{1}{2}m^2+m-1}{m^3}\right)c_2\left(W\right)\\
&\leq \left(2+\frac{-\frac{1}{2}m^2+\frac{1}{2}m^2-1}{m^3}\right)c_2\left(W\right)\\
&= \left(2-\frac{1}{m^3}\right)c_2\left(W\right)\\
&\leq 2c_2\left(W\right).
\end{aligned}
$$

This shows the first part of the theorem. For the second part, we combine T_2 and M_2 which results in a spanning walk $W_{\text{approx},2}$ such that:

$$
\begin{aligned}
c_1\left(W_{\text{approx},2}\right) &\leq c_1\left(T_2\right)+c_1\left(M_2\right)\\
&\leq \left(1+\frac{\varepsilon}{m^2}\right)\frac{m-1}{m}c_1\left(W\right)+\left(1+\frac{\varepsilon}{2m}\right)\frac{1}{2}c_1\left(W\right)\\
&\leq \left(1+\frac{1}{m^2}\right)\frac{m-1}{m}c_1\left(W\right)+\left(1+\frac{1}{2m}\right)\frac{1}{2}c_1\left(W\right)\\
&= \left(\frac{m-1}{m}+\frac{m-1}{m^3}+\frac{1}{2}+\frac{1}{4m}\right)c_1\left(W\right)\\
&= \left(\frac{3}{2}-\frac{1}{m}+\frac{m-1}{m^3}+\frac{1}{4m}\right)c_1\left(W\right)\\
&= \left(\frac{3}{2}+\frac{-m^2+m-1+\frac{1}{4}m^2}{m^3}\right)c_1\left(W\right)\\
&= \left(\frac{3}{2}+\frac{-1+m-\frac{3}{4}m^2}{m^3}\right)c_1\left(W\right)\\
&\leq \left(\frac{3}{2}-\frac{1}{m^3}\right)c_1\left(W\right)\\
&\leq \frac{3}{2}c_1\left(W\right)
\end{aligned}
$$

and

$$
c_2\left(W_{\text{approx},2}\right) \leq c_2\left(T_2\right)+c_2\left(M_2\right)
$$

$$
\begin{aligned}
&\le \left(1+\frac{\varepsilon}{m^2}\right) c_2\,(W) + \left(1+\frac{\varepsilon}{2m}\right) c_2\,(W) \\
&\le \left(1+\frac{\varepsilon}{4}\right) c_2\,(W) + \left(1+\frac{\varepsilon}{4}\right) c_2\,(W) \\
&\le (2+\varepsilon)\, c_2\,(W)\,.
\end{aligned}
$$

□

5 Approximation for Traveling Salesman Path Problems

Regarding two-objective traveling salesman path problems we obtain the following results.

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

The deterministic approximation for 2-TSPP$_{\mathrm{st}}$ is easily obtained by a tree-doubling of the approximated Pareto-minimal spanning trees. The constructions of the randomized approximation algorithms are more complicated. Each of them relies on an argument that assures the existence of a path matching with sufficiently low costs, which is constructed in a separate, combinatorial lemma. With this lemma at hand we can follow the standard strategy 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 path matchings of these vertices and finally pairwise combine these matchings with their corresponding tree.

The following lemma assures that path matchings with sufficiently low costs exist.

Lemma 5.1 *Let $G=(V,E,c)$ be some $\mathbb{N}^2$-labeled multigraph, $U \subseteq V$ be a nonempty set and W be some spanning walk of (V,E).*

1. *If $\#U$ is odd, then there exist a vertex $s \in U$ and a path matching M of $U \setminus \{s\}$ in (V,E) such that*
$$c(M) \le \left(\tfrac{1}{2}c_1(W), c_2(W)\right).$$
2. *If $\#U$ is even, then there exist distinct vertices $s,t \in U$ and a path matching M of $U \setminus \{s,t\}$ in (V,E) such that*
$$c(M) \le \left(\tfrac{1}{2}c_1(W), \tfrac{2}{3}c_2(W)\right).$$

Proof The lemma is obvious for $\#U < 3$, so assume $\#U \ge 3$. Let $(v_1,\dots,v_r)$ denote the vertices of U in the order of their first appearance in W. We cut the walk W at each v_i to obtain a set of $r-1$ subwalks $\{W_1, W_2, \dots, W_{r-1}\}$, where W_i connects v_i and v_{i+1}, and extract a set of paths $P = \{P_1, P_2, \dots, P_{r-1}\}$ by removing cycles in each W_i.

Define the following distinct sets that partition P.

$$
\begin{aligned}
M_{\mathrm{odd}} &= \{P_i \in P \mid i \text{ is odd}\} \\
M_{\mathrm{even}} &= \{P_i \in P \mid i \text{ is even}\}
\end{aligned}
$$

If $\#U$ is odd, then M_{odd} is a path matching on $U \setminus \{v_r\}$, and M_{even} is a path matching on $U \setminus \{v_1\}$. In this case, choose the path matching with the lower costs in c_1. From $M_{\text{even}} \cup M_{\text{odd}} = P$ and $c(P) \le c(W)$, the first part of the lemma follows.

So assume $\#U$ is even. In this case M_{odd} is a path matching on U and M_{even} is a path matching on $U \setminus \{v_1, v_r\}$. We show this part by contradiction. Hence assume that the second part of Lemma 5.1 does not hold, which means that for every path matching M of $U \setminus \{s, t\}$ for any two $s, t \in U$ it holds that

$$c_1(M) > \tfrac{1}{2}c_1(W) \quad \text{or} \quad c_2(M) > \tfrac{2}{3}c_2(W). \tag{4}$$

This also holds for path matchings of U, since otherwise, by removing an arbitrary path, we obtain a path matching that leaves two vertices unmatched and that contradicts (4).

Consider the following two cases:

Case 1: $c_2(M_{\text{odd}}) \le \frac{2}{3}c_2(W)$ and $c_2(M_{\text{even}}) \le \frac{2}{3}c_2(W)$.

From $c_1(M_{\text{even}}) + c_1(M_{\text{odd}}) = c_1(P) \le c_1(W)$ it follows that $c_1(M_{\text{odd}}) \le \frac{1}{2}c_1(W)$ or $c_1(M_{\text{even}}) \le \frac{1}{2}c_1(W)$. So at least one of the path matchings M_{odd} (of U) and M_{even} (of $U \setminus \{v_1, v_r\}$) contradicts (4).

Case 2: $c_2(M_{\text{odd}}) > \frac{2}{3}c_2(W)$.

Since $c_2(M_{\text{even}}) + c_2(M_{\text{odd}}) = c_2(P) \le c_2(W)$, we have

$$c_2(M_{\text{even}}) \le c_2(W) - c_2(M_{\text{odd}}) < \tfrac{1}{3}c_2(W). \tag{5}$$

For every $1 \le k < r$, the set of paths P_i that lie left (resp., right) of the first visit of v_k is denoted by L_k (resp., R_k), i. e.,

$$L_k = \{P_i \mid 1 \le i < k\} \qquad \text{and} \qquad R_k = \{P_i \mid k+1 \le i < r\}.$$

Consider the largest odd k such that $c_2(L_k \cap M_{\text{odd}}) \le \frac{1}{2}c_2(M_{\text{odd}})$. From $P_k \in M_{\text{odd}}$, $P_k \notin L_k \cup R_k$, and the maximality of k, it follows that $c_2(R_k \cap M_{\text{odd}}) \le \frac{1}{2}c_2(M_{\text{odd}})$.

Referring to Figure 3, we now show that either the path matching $M_1 = (L_k \cap M_{\text{odd}}) \cup (R_k \cap M_{\text{even}})$ on $U \setminus \{v_k, v_r\}$ or the path matching $M_2 = (L_k \cap M_{\text{even}}) \cup (R_k \cap M_{\text{odd}})$ on $U \setminus \{v_1, v_{k+1}\}$ is a matching that contradicts (4).

Let us estimate the costs in the second component of M_1 and M_2:

$$\begin{aligned} c_2(M_1) &= c_2(L_k \cap M_{\text{odd}}) + c_2(R_k \cap M_{\text{even}}) \\ &\le \tfrac{1}{2}c_2(M_{\text{odd}}) + c_2(M_{\text{even}}) \end{aligned} \qquad \begin{aligned} c_2(M_2) &= c_2(L_k \cap M_{\text{even}}) + c_2(R_k \cap M_{\text{odd}}) \\ &\le c_2(M_{\text{even}}) + \tfrac{1}{2}c_2(M_{\text{odd}}) \end{aligned}$$

By (5) we know that $c_2(M_{\text{even}}) < \frac{1}{3}c_2(W)$ and thus we obtain

$$\begin{aligned} c_2(M_1), c_2(M_2) &\le \tfrac{1}{2}c_2(M_{\text{odd}}) + c_2(M_{\text{even}}) \\ &= \tfrac{1}{2}(c_2(P) - c_2(M_{\text{even}})) + c_2(M_{\text{even}}) \\ &= \tfrac{1}{2}(c_2(P) + c_2(M_{\text{even}})) \\ &< \tfrac{1}{2}(c_2(W) + \tfrac{1}{3}c_2(W)) \\ &= \tfrac{2}{3}c_2(W). \end{aligned}$$

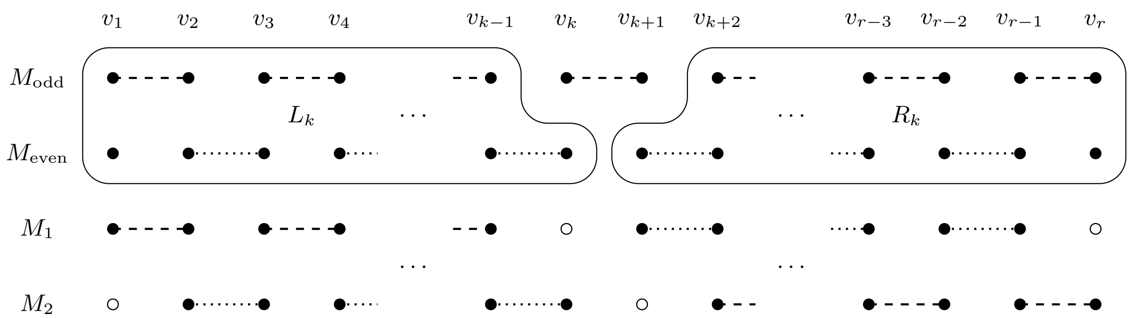

Figure 3: We obtain M_1 by merging the left part of M_{odd} with the right part of M_{even}, leaving v_k and v_r unmatched. Analogously, we obtain M_2 by merging the left part of M_{even} with the right part of M_{odd}, leaving v_1 and v_{k+1} unmatched.

Note that M_1 and M_2 are disjoint. Therefore, $c_1(M_1) + c_1(M_2) \leq c_1(P)$ and hence

$$c_1(M_1) \leq \tfrac{1}{2}c_1(W) \quad \text{or} \quad c_1(M_2) \leq \tfrac{1}{2}c_1(W).$$

So M_1 or M_2 contradicts (4).

Case 3: $c_2(M_{\text{even}}) > \frac{2}{3}c_2(W)$.

This case is very similar to Case 2, so we concentrate on the differences. We get $c_2(M_{\text{odd}}) < \frac{1}{3}c_2(W)$ and define L_k and R_k in the same way as in Case 2. Consider now the largest *even* k such that $c_2(L_k \cap M_{\text{even}}) \leq \frac{1}{2}c_2(M_{\text{even}})$. From $P_k \in M_{\text{even}}$, $P_k \notin L_k \cup R_k$, and the maximality of k, it follows that $c_2(R_k \cap M_{\text{even}}) \leq \frac{1}{2}c_2(M_{\text{even}})$.

We now show that either the path matching $M_1 = (L_k \cap M_{\text{odd}}) \cup (R_k \cap M_{\text{even}})$ on $U \setminus \{v_{k+1}, v_r\}$ or the path matching $M_2 = (L_k \cap M_{\text{even}}) \cup (R_k \cap M_{\text{odd}})$ on $U \setminus \{v_1, v_k\}$ is a matching that contradicts (4).

We similarly get $c_2(M_1), c_2(M_2) \leq c_2(M_{\text{odd}}) + \frac{1}{2}c_2(M_{\text{even}})$ and using $c_2(M_{\text{odd}}) < \frac{1}{3}c_2(W)$ we obtain $c_2(M_1), c_2(M_2) < \frac{2}{3}c_2(W)$ and also $c_1(M_1) \leq \frac{1}{2}c_1(W)$ or $c_1(M_2) \leq \frac{1}{2}c_1(W)$. So M_1 or M_2 contradicts (4).

This finishes the proof of Lemma 5.1. □

Theorem 5.2 *For every* $\varepsilon > 0$, 2-TSPP *is randomized* $({}^3\!/_2 + \varepsilon, {}^5\!/_3 + \varepsilon)$*-approximable.*

Proof Let $\varepsilon > 0$. The approximation is achieved by the following algorithm which works on input of an $\mathbb{N}^2$-labeled multigraph (V, E, c). Please recall the definitions of the algorithms `2-MST-Approx` and `2-MPM-Approx`$_{\texttt{R}}$ from section 3.

Algorithm: `2-TSPP-Approx`$_\varepsilon$ (V, E, c)

Input : $\mathbb{N}^2$-labeled multigraph (V, E, c)
Output: set of spanning walks of (V, E, c)

1 $\mathcal{S} := \emptyset$;
2 $\mathcal{P} :=$ `2-MST-Approx`$(V, E, c, \frac{\varepsilon}{2})$;
3 **foreach** $T \in \mathcal{P}$ **do**
4     $U := \{v \in V \mid \deg_T(v)$ is odd$\}$;
5   **foreach** $s, t \in U$ with $s \neq t$ **do**
6         $\mathcal{A} :=$ `2-MPM-Approx`$_\mathrm{R}(V, E, c, U \setminus \{s, t\}, \frac{\varepsilon}{2})$;
7   **foreach** $M \in \mathcal{A}$ **do**
8             $W_{\text{approx}} :=$ spanning walk of (V, E) using the edges of T and M;
9             $\mathcal{S} := \mathcal{S} \cup \{W_{\text{approx}}\}$
10   **end**
11   **end**
12 **end**
13 **return** $\mathcal{S}$

Observe that the set U in line 4 has an even number of elements. Also, note that in line 8, the spanning walk can be constructed, since after combining M and T, the vertices s and t have odd degree, while all remaining vertices have even, nonzero degree. Since line 6 uses an FPRAS, our algorithm is randomized. Observe that each line of the algorithm is computable in polynomial time and each of the sets $\mathcal{P}$, U and $\mathcal{A}$ has only a polynomial number of elements. Therefore, `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 W denote an arbitrary spanning walk of (V, E, c). We show that `2-TSPP-Approx`$_\varepsilon$ outputs at least one spanning walk W_{approx} such that

$$c_1(W_{\text{approx}}) \leq \left(\tfrac{3}{2} + \varepsilon\right) c_1(W) \quad \text{and} \quad c_2(W_{\text{approx}}) \leq \left(\tfrac{5}{3} + \varepsilon\right) c_2(W).$$

Fix a spanning tree T_{approx} with costs $c(T_{\text{approx}}) \leq (1+\frac{\varepsilon}{2})c(W)$ from the $(1+\frac{\varepsilon}{2})$-approximate Pareto set $\mathcal{P}$ computed in line 2. $\mathcal{P}$ contains such a tree, because W is a spanning walk on (V, E, c) and thus contains a spanning tree whose costs are not greater than those of W, 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 beginning in line 3 that uses the tree T_{approx}.

By Lemma 5.1, there exists a path matching M on U that leaves exactly two vertices $s, t \in U$ unmatched and that has costs $c(M) \leq (\frac{1}{2}c_1(W), \frac{2}{3}c_2(W))$. Therefore, with probability at least $1/2$, the $(1 + \frac{\varepsilon}{2})$-approximate Pareto set $\mathcal{A}$ in line 6 contains a path matching M_{approx} that leaves some s and t unmatched such that

$$c(M_{\text{approx}}) \;\leq\; \left(1 + \tfrac{\varepsilon}{2}\right) \cdot \left(\tfrac{1}{2}c_1(W), \tfrac{2}{3}c_2(W)\right) \;\leq\; \left(\left(\tfrac{1}{2} + \tfrac{\varepsilon}{2}\right) c_1(W), \left(\tfrac{2}{3} + \tfrac{\varepsilon}{2}\right) c_2(W)\right).$$

We combine T_{approx} and M_{approx} to obtain a spanning walk from s to t with costs

$$\begin{aligned} c(W_{\text{approx}}) &= c(T_{\text{approx}}) + c(M_{\text{approx}}) \\ &\leq \left(1 + \tfrac{\varepsilon}{2}\right) c(W) + \left(\left(\tfrac{1}{2} + \tfrac{\varepsilon}{2}\right) c_1(W), \left(\tfrac{2}{3} + \tfrac{\varepsilon}{2}\right) c_2(W)\right) \\ &\leq \left(\left(\tfrac{3}{2} + \varepsilon\right) c_1(W), \left(\tfrac{5}{3} + \varepsilon\right) c_2(W)\right) \end{aligned}$$

and hence W is $(3/2 + \varepsilon, 5/3 + \varepsilon)$-approximated by W_{approx}. □

Theorem 5.3 *For every $\varepsilon > 0$, 2-TSPP$_{\mathrm{s}}$ is randomized $(3/2 + \varepsilon, 2 + \varepsilon)$-approximable.*

Proof The proof is similar to the proof of Theorem 5.2. 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 an $\mathbb{N}^2$-labeled multigraph (V, E, c) and some starting vertex $s \in V$.

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

Input : $\mathbb{N}^2$-labeled multigraph (V, E, c) and some $s \in V$
Output: set of spanning walks of (V, E, c), each starting at s

1 $\mathcal{S} := \emptyset$;
2 $\mathcal{P} :=$ `2-MST-Approx`$(V, E, c, \frac{\varepsilon}{2})$;
3 `foreach` $T \in \mathcal{P}$ `do`
4     $U := \{v \in V \mid \deg_T(v)$ `is odd if and only if` $v \neq s\}$;
5   `foreach` $t \in U$ `do`
6         $\mathcal{A} :=$ `2-MPM-Approx`$_{\mathtt{R}}$$(V, E, c, U \setminus \{t\}, \frac{\varepsilon}{2})$;
7   `foreach` $M \in \mathcal{A}$ `do`
8             $W_{\mathrm{approx}} :=$ `spanning walk of` (V, E) `using the edges of` T `and` M;
9             $\mathcal{S} := \mathcal{S} \cup \{W_{\mathrm{approx}}\}$
10   `end`
11   `end`
12 `end`
13 `return` $\mathcal{S}$

Observe that the set U in line 4 has an odd number of elements. Also, note that in line 8, the spanning walk exists and can be easily constructed, since after combining M and T, the vertices s and t have odd degree, while all remaining vertices have even degree greater than zero. 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 W denote an arbitrary spanning walk that starts at s. We show that `2-TSPP`$_s$`-Approx`$_\varepsilon$ contains at least one spanning walk W_{approx} that starts in s such that

$$c_1(W_{\mathrm{approx}}) \leq \left(\tfrac{3}{2} + \varepsilon\right) c_1(W) \quad \text{and} \quad c_2(W_{\mathrm{approx}}) \leq (2 + \varepsilon)\, c_2(W).$$

Fix a spanning tree T_{approx} with costs $c(T_{\mathrm{approx}}) \leq (1 + \frac{\varepsilon}{2}) c(W)$ from the $(1 + \frac{\varepsilon}{2})$-approximate Pareto set $\mathcal{P}$ computed in line 2. From now on we consider the iteration of the loop starting in line 3 that uses the tree T_{approx}.

By Lemma 5.1, there exists a path matching M on U that leaves exactly one vertex $t \in U$ unmatched and that has costs $c(M) \leq (\frac{1}{2} c_1(W), c_2(W))$. Therefore, with probability at least $1/2$, the approximate Pareto set $\mathcal{A}$ in line 6 contains a path matching M_{approx} that leaves some t unmatched such that

$$c(M_{\mathrm{approx}}) \;\leq\; \left(1 + \tfrac{\varepsilon}{2}\right) \cdot \left(\tfrac{1}{2} c_1(W), c_2(W)\right) \;\leq\; \left(\left(\tfrac{1}{2} + \tfrac{\varepsilon}{2}\right) c_1(W), \left(1 + \tfrac{\varepsilon}{2}\right) c_2(W)\right).$$

We combine T_{approx} and M_{approx} to obtain a spanning walk W_{approx} from s to t with costs

$$\begin{aligned} c(W_{\mathrm{approx}}) &= c(T_{\mathrm{approx}}) + c(M_{\mathrm{approx}}) \\ &\leq \left(1 + \tfrac{\varepsilon}{2}\right) c(W) + \left(\left(\tfrac{1}{2} + \tfrac{\varepsilon}{2}\right) c_1(W), \left(1 + \tfrac{\varepsilon}{2}\right) c_2(W)\right) \\ &\leq \left(\left(\tfrac{3}{2} + \varepsilon\right) c_1(W), (2 + \varepsilon)\, c_2(W)\right). \end{aligned}$$

Hence W is $(3/2 + \varepsilon, 2 + \varepsilon)$-approximated by W_{approx}. $\square$

Theorem 5.4 *For every $\varepsilon > 0$, 2-TSPP$_{\text{st}}$ is $(2 + \varepsilon, 2 + \varepsilon)$-approximable.*

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

Let (V, E, c) be an arbitrary $\mathbb{N}^2$-labeled multigraph, $s, t \in V$ with $s \neq t$ and $\varepsilon > 0$. Furthermore, let $\mathcal{A} = \texttt{2-MST-Approx}(V, E, c, \frac{\varepsilon}{2})$. For each tree $T \in \mathcal{A}$ 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 greater than zero except for s and t. Therefore we can find a spanning walk W_{approx} from s to t, having costs $c(W_{\text{approx}}) \leq 2c(T)$.

Fix any arbitrary spanning walk W from s to t. Since W contains a spanning tree, there is a spanning tree $T \in \mathcal{A}$ such that $c(T) \leq (1 + \frac{\varepsilon}{2})c(W)$. By the tree doubling method we get a spanning walk W_{approx} from s to t with $c(W_{\text{approx}}) \leq 2c(T) \leq (2 + \varepsilon)c(W)$. $\square$

Corollary 5.5 *For every $\varepsilon > 0$, 2-TSPP and 2-TSPP$_{\text{s}}$ are $(2 + \varepsilon, 2 + \varepsilon)$-approximable.*

6 Lower Bound Arguments

This section provides arguments that indicate the hardness of improving the two-objective approximation algorithms that were given in sections 4 and 5. In summary, if one can improve our randomized approximations for 2-TSP, 2-TSPP, 2-TSPP$_{\text{s}}$, or their componentwise metric restrictions, then this improves the best known approximations for TSP, TSPP$_{\text{s}}$, and TSPP$_{\text{st}}$, i. e., the approximations by Christofides [Chr76] and Hoogeveen [Hoo91]. Improvements of these well-studied approximations seem very difficult to obtain. Table 2 summarizes the results of this section.

6.1 Lower Bound Arguments for 2-TSP

Below we construct an approximation preserving reduction from TSPP$_{\text{st}}$ to componentwise metric 2-TSP. This gives evidence for the difficulty of improving the randomized approximations for 2-TSP that are given in Theorem 4.4.

An improvement of the first component (i. e., a $(3/2 - \varepsilon, 2 + \varepsilon)$-approximation) would improve Christofides' $3/2$-approximation for TSP [Chr76]. An improvement of the second component (i. e., a $(5/3 - \varepsilon, 2 - \varepsilon)$-approximation) would improve Hoogeveen's $5/3$-approximation for TSPP$_{\text{st}}$ [Hoo91]. We have no evidence in favor of or against an (α, β)-approximation where $5/3 \leq \alpha, \beta < 2$.

Open Question: Is 2-TSP (α, β)-approximable where $5/3 \leq \alpha, \beta < 2$?

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

Proposition 6.1 *Let $\alpha > 1$ and $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* componentwise metric 2-TSP *is $(3/2 - \varepsilon, \alpha)$-approximable, then* TSP *is $(3/2 - \varepsilon)$-approximable.*

Theorem 6.2 *Let $\alpha > 1$ and $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* componentwise metric 2-TSP *is $(\alpha, 2 - \varepsilon)$-approximable, then* TSPP_{st} *is α-approximable.*

Proof Let $\mathcal{A}$ be an algorithm that on input of a complete $\mathbb{N}^2$-labeled simple graph (V, E, c) with componentwise metric distance function c returns an $(\alpha, 2 - \varepsilon)$-approximation for componentwise metric 2-TSP for some $\alpha \geq 1$ and some $\varepsilon > 0$. Let $((V, E, c'), s, t)$ be an arbitrary TSPP_{st}-instance where $V = \{s, t, v_1, \ldots, v_k\}$. Since TSPP_{st} is a single-objective problem, we can assume that the graph is simple and complete and c' is metric. We will construct an instance I of componentwise metric 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 $B^s = \{s = b^s_0, b^s_1, \ldots, b^s_{r-1}, s' = b^s_r\}$ and $B^t = \{t = b^t_0, b^t_1, \ldots, b^t_{r-1}, t = b^t_r\}$. So the vertices of our componentwise metric 2-TSP instance are $V \cup V' \cup B^s \cup B^t$ and we have all possible edges since the graph must be complete. The labeling function will be defined as follows. First, we define it directly for some of the edges:

- for $e \subseteq V$ or $e \subseteq V'$, we set $c(e) = (c'(e), 0)$
- for $e = \{b^s_i, b^s_{i+1}\}$ or $e = \{b^t_i, b^t_{i+1}\}$, we set $c(e) = (0, 1)$

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 metric, 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 metric 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 metric.

Let P be the c'-shortest Hamiltonian path between s and t in V and P' its (reversed) copy in V'. $P \cup \{\{t, b^t_1\}, \ldots, \{b^t_{r-1}, t'\}\} \cup P' \cup \{\{s', b^s_{r-1}\}, \ldots, \{b^s_1, s\}\}$ is obviously a Hamiltonian cycle in the new graph with costs $(2c'(P), 2r)$. 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 $c_2(S) \leq 4r - 2\varepsilon r$. 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 $E_{B^t} := \{\{b^t_{i-1}, b^t_i\} \mid 1 \leq i \leq r\} \cup \{\{b^t_i, b^t_{i-1}\} \mid 1 \leq i \leq r\}$ 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

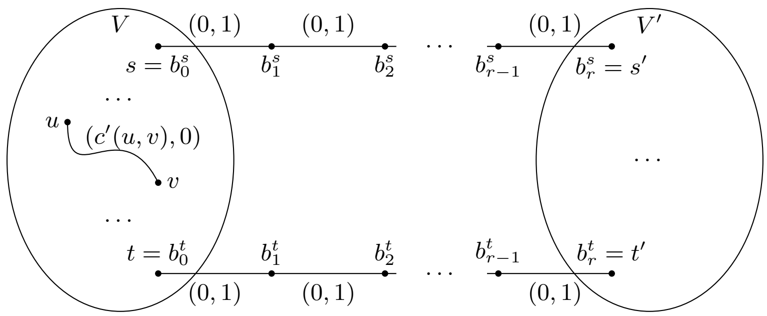

Figure 4: Creating an instance of componentwise metric 2-TSP from an instance $((V,E,c'),s,t)$ of TSPP_{st}. We first make a copy V' of V and, for each $u,v \in V$, we set $c(u,v) = (c'(u,v),0)$ and $c(u',v') = (c'(u,v),0)$. We further connect s with s' and t with t' by $r-1$ bridge vertices b_i^s, b_i^t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

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 $\{\{u,v\} \mid \text{either } u,v \in V \text{ or } u,v \in V'\} \cup E_{B^s} \cup E_{B^t}$. 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 only uses edges from $\{\{u,v\} \mid \text{either } u,v \in V \text{ or } u,v \in V'\} \cup E_{B^s} \cup E_{B^t}$.

We will now argue that S uses each bridge exactly once. We denote by $u(x,y)$ the number of times the 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 6.3 *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 6.4 *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 6.3.

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 6.3. □

Claim 6.5 *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 not be connected. □

Claim 6.6 *All bridge edges $e \in E_{B^s} \cup E_{B^t}$ have odd usage count $u(e)$.*

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 6.5) and thus

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

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

If $u(e) > 1$ for some bridge edge e, we can always remove two uses of this edge from S without destroying the property of S being a Hamiltonian circuit. 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 restricted to V is a Hamiltonian path from s to t and another one can be obtained by restricting S to 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 $((V, E, c'), s, t)$. □

Corollary 6.7 *Let $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* componentwise metric 2-TSP *is $(5/3 - \varepsilon, 2 - \varepsilon)$-approximable, then* $\mathrm{TSPP_{st}}$ *is $(5/3 - \varepsilon)$-approximable.*

6.2 Lower Bound Arguments for 2-TSPP

This section provides two approximation preserving reductions, one from $\mathrm{TSPP_{st}}$ to componentwise metric 2-TSPP, and another one from $\mathrm{TSPP_s}$ to componentwise metric 2-TSPP. Both reductions give evidence that the randomized approximation for 2-TSPP that is given in Theorem 5.2 is difficult to improve.

An improvement of the first component (i. e., a $(3/2 - \varepsilon, 5/3 + \varepsilon)$-approximation) would improve Hoogeveen's $3/2$-approximations for TSPP and $\mathrm{TSPP_s}$. An improvement of both components (i. e., a $(3/2 - \varepsilon, 5/3 - \varepsilon)$-approximation) would improve Hoogeveen's $5/3$-approximation for $\mathrm{TSPP_{st}}$ [Hoo91].

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

Proposition 6.8 *Let $\alpha > 1$ and $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* componentwise metric 2-TSPP *is $(3/2 - \varepsilon, \alpha)$-approximable, then* TSPP *is $(3/2 - \varepsilon)$-approximable.*

Theorem 6.9 *Let $\alpha > 1$ and $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* componentwise metric 2-TSPP *is $(\alpha, 3/2 - \varepsilon)$-approximable, then* $\mathrm{TSPP_{st}}$ *is α-approximable.*

Proof We proceed analogously to the proof of Theorem 6.2 and reduce $\mathrm{TSPP}_{\mathrm{st}}$ to componentwise metric 2-TSPP. Let therefore $\mathcal{A}$ be an algorithm that on input of a complete $\mathbb{N}^2$-labeled simple graph (V, E, c) with componentwise metric distance function c returns an $(\alpha, 3/2 - \varepsilon)$-approximation for componentwise metric 2-TSPP for some $\alpha > 1$ and some $\varepsilon > 0$, and let furthermore $I = ((V, E, c'), s, t)$ be a $\mathrm{TSPP}_{\mathrm{st}}$-instance where $V = \{s, t, v_1, \dots, v_k\}$ and (V, E, c) is a complete simple graph with metric c.

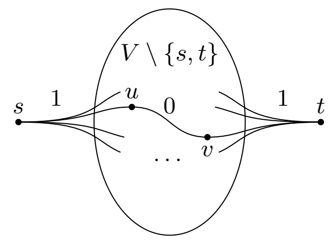

Figure 5: Structure of distance function c_2. For any $u \in V \setminus \{s,t\}$, we have $c_2(s,u) = c_2(t,u) = 1$. Inside $V \setminus \{s,t\}$, we have zero c_2 distances, hence $c_2(u,v) = 0$ for all $u, v \in V \setminus \{s,t\}$.

For each edge $e \in E$, let $c_1(e) = c'(e)$. We define $c_2 \colon E \to \mathbb{N}$ 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 metric functions on V. Let $c = (c_1, c_2)$ and define a componentwise metric 2-TSPP instance as $I' = (V, E, c)$.

Figure 5 shows the structure of the distance function c_2. Obviously, all Hamiltonian paths y between s and t have length $c_2(y) = 2$, whereas all other Hamiltonian paths y' 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 $c' = c_1$. $\square$

Corollary 6.10 *Let $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* componentwise metric 2-TSPP *is $(3/2 - \varepsilon, 5/3 - \varepsilon)$-approximable, then* $\mathrm{TSPP}_{\mathrm{st}}$ *is $(5/3 - \varepsilon)$-approximable.*

Theorem 6.11 *Let $\alpha > 1$ and $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* componentwise metric 2-TSPP *is $(\alpha, 2 - \varepsilon)$-approximable, then* $\mathrm{TSPP}_{\mathrm{s}}$ *is α-approximable.*

Proof We reduce $\mathrm{TSPP_s}$ to componentwise metric 2-TSPP. Let again $\mathcal{A}$ be an algorithm that on input of a complete $\mathbb{N}^2$-labeled simple graph (V, E, c) with componentwise metric c returns an $(\alpha, 2-\varepsilon)$-approximation for componentwise metric 2-TSPP for some $\alpha > 1$ and some $\varepsilon > 0$, and let $x = ((V, E, c'), s)$ be the $\mathrm{TSPP_s}$-instance, where $V = \{s, v_1, \ldots, v_k\}$ and (V, E, c') is a complete simple graph with metric c'.

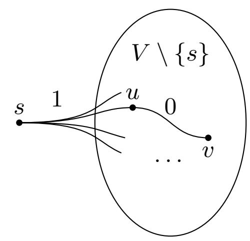

Figure 6: Structure of distance function c_2 of x'. For all $u \in V \setminus \{s\}$, we set $c_2(s,u) = 1$. All c_2 distances inside $V \setminus \{s\}$ are zero: $c_2(u, v) = 0$ for all $u, v \in V \setminus \{s\}$.

For each edge $e \in E$, let $c_1(e) = c'(e)$. We define $c_2 \colon E \to \mathbb{N}$ as follows (cf. Figure 6):

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

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

This time, all Hamiltonian paths y with endpoint s have length $c_2(y) = 1$, whereas all other Hamiltonian paths y' must have a 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. $\square$

Corollary 6.12 *Let $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* componentwise metric 2-TSPP *is $(3/2-\varepsilon, 2-\varepsilon)$-approximable, then* $\mathrm{TSPP_s}$ *is $(3/2-\varepsilon)$-approximable.*

6.3 Lower Bound Arguments for 2-$\mathrm{TSPP_s}$

Below we construct an approximation preserving reduction from $\mathrm{TSPP_{st}}$ to componentwise metric 2-$\mathrm{TSPP_s}$, and a similar reduction from TSP to componentwise metric 2-$\mathrm{TSPP_s}$. This gives evidence for the difficulty of improving the randomized approximability of 2-$\mathrm{TSPP_s}$ that is given in Theorem 5.3.

An improvement of the first component (i. e., a $(3/2 - \varepsilon, 2 + \varepsilon)$-approximation) would improve Hoogeveen's $3/2$-approximation for $\mathrm{TSPP_s}$. An improvement of the second component (i. e., a $(3/2 + \varepsilon, 2 - \varepsilon)$-approximation) would considerably improve Hoogeveen's $5/3$-approximation for

TSPP_{st}[Hoo91]. An improvement of both component (i. e., a $(3/2-\varepsilon, 2-\varepsilon)$-approximation) would improve Christofides' $3/2$-approximation for TSP [Chr76].

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

Proposition 6.13 *Let $\alpha > 1$ and $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* componentwise metric 2-TSPP_{s} *is $(3/2-\varepsilon, \alpha)$-approximable, then* TSPP_{s} *is $(3/2-\varepsilon)$-approximable.*

Theorem 6.14 *Let $\alpha > 1$ and $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* componentwise metric 2-TSPP_{s} *is $(\alpha, 2-\varepsilon)$-approximable, then* TSPP_{st} *is α-approximable.*

Proof We reduce TSPP_{st} to componentwise metric 2-TSPP_{s}. Let $\mathcal{A}$ be a deterministic/randomized algorithm that $(\alpha, 2-\varepsilon)$-approximates componentwise metric 2-TSPP_{s} for some $\alpha > 1$ and some $\varepsilon > 0$. Let $((V, E, c), s, t)$ be an arbitrary TSPP_{st}-instance where (V, E, c) is a complete simple graph with metric c and $V = \{s, t, v_1, \ldots, v_k\}$. We construct a componentwise metric 2-TSPP_{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, $I = ((V, E, c'), s)$ 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 $x \in \{s, v_1, \ldots, v_k\}$.

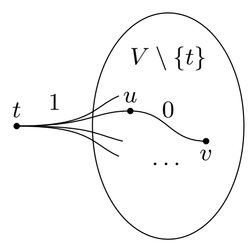

Figure 7: Structure of distance function c_2 of I. For all $u \in V \setminus \{t\}$, we set $c_2(t, u) = 1$. All c_2 distances inside $V \setminus \{t\}$ are zero: $c_2(u, v) = 0$ for all $u, v \in V \setminus \{t\}$.

We argue that $\mathcal{A}(I)$ computes an α-approximation for the TSPP_{st}-instance $((V, E, c), s, t)$. Let $P \subseteq E$ be an optimal s-t-path with respect to c. Then P is also a valid solution of the TSPP_{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 6.15 *Let $\alpha > 1$. The following holds for deterministic/randomized approximations: If* TSPP_{st} *is α-approximable, then* TSP *is α-approximable.*

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

Let $I = (V, E, c)$ be a given TSP-instance where $V = \{v_1, \ldots, v_m\}$. For all $t \in V \setminus \{v_1\}$, 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 6.16 *Let $\varepsilon > 0$. The following holds for deterministic/randomized approximations:*

1. *If* componentwise metric 2-TSPP$_\mathrm{s}$ *is $(3/2 + \varepsilon, 2 - \varepsilon)$-approximable, then* TSPP$_\mathrm{st}$ *is $(3/2 + \varepsilon)$-approximable.*
2. *If* componentwise metric 2-TSPP$_\mathrm{s}$ *is $(3/2 - \varepsilon, 2 - \varepsilon)$-approximable, then* TSPP$_\mathrm{st}$ *and* TSP *are $(3/2 - \varepsilon)$-approximable.*

Proof Follows from Theorem 6.14 and Proposition 6.15. □

6.4 Lower Bound Arguments for 2-TSPP$_\mathrm{st}$

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

Proposition 6.17 *Let $\alpha > 1$ and $\varepsilon > 0$. The following holds for deterministic/randomized approximations: If* componentwise metric 2-TSPP$_\mathrm{st}$ *is $(5/3 - \varepsilon, \alpha)$-approximable, then* TSPP$_\mathrm{st}$ *is $(5/3 - \varepsilon)$-approximable.*

7 Open Questions

The results in the previous sections raise the following questions, which apply to all considered versions of the multi-objective problems: the general version, the metric version, and the componentwise metric version.

1. By Theorem 4.4, 2-TSP is randomized $(3/2, 2 + \varepsilon)$-approximable and randomized $(3/2 + \varepsilon, 2)$-approximable. By Proposition 6.1 and Corollary 6.7, 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 5.2, 2-TSPP is randomized $(3/2 + \varepsilon, 5/3 + \varepsilon)$-approximable, and by Corollary 6.12, 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 5.4, 2-$\mathrm{TSPP}_{\mathrm{st}}$ is $(2+\varepsilon, 2+\varepsilon)$-approximable, and by Proposition 6.17, 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 6, we gave the following reductions that were used to translate approximations from one to another optimization problem: $\mathrm{TSPP}_{\mathrm{st}} \leq$ 2-TSP, $\mathrm{TSPP}_{\mathrm{st}} \leq$ 2-TSPP, $\mathrm{TSPP}_{\mathrm{s}} \leq$ 2-TSPP, $\mathrm{TSPP}_{\mathrm{st}} \leq$ 2-$\mathrm{TSPP}_{\mathrm{s}}$, and TSP $\leq \mathrm{TSPP}_{\mathrm{st}}$. Can one find nontrivial reductions between the single-objective problems TSP, TSPP, $\mathrm{TSPP}_{\mathrm{s}}$, and $\mathrm{TSPP}_{\mathrm{st}}$? For instance, does the existence of a $(5/3-\varepsilon)$-approximation for $\mathrm{TSPP}_{\mathrm{st}}$ 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 $\mathrm{TSPP}_{\mathrm{st}}$? Such translations of the approximability between the single-objective problems would give a better understanding of the difficulty of these problems.

Acknowledgements. The authors thank Volker Diekert, Bodo Manthey, and Heinz Schmitz for valuable discussions on multi-objective 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 multicriteria 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.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

[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.

[Han79] P. Hansen. Bicriterion path problems. In *Proceedings of the 3rd Conference on Multiple Criteria Decision Making Theory and Application*, volume 177 of *Lecture Notes in Economics and Mathematical Systems*, pages 109–127. Springer Verlag, 1979.

[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.

[PY82] C. H. Papadimitriou and M. Yannakakis. The complexity of restricted spanning tree problems. *J. ACM*, 29(2):285–309, 1982.

[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