On the Recoverable Traveling Salesman Problem

Marc Goerigk\textsuperscript{1}[0000–0002–2516–0129], Stefan Lendl\textsuperscript{2}[0000–0002–5660–5397], and
Lasse Wulf\textsuperscript{3}[0000–0001–7139–4092]

\textsuperscript{1} Network and Data Science Management, University of Siegen, Unteres Schloß 3,
57072 Siegen, Germany marc.goerigk@uni-siegen.de
\textsuperscript{2} Institute of Operations and Information Systems, University of Graz,
Universitätsstraße 15, 8010 Graz, Austria stefan.lendl@uni-graz.at
\textsuperscript{3} Institute of Discrete Mathematics, Graz University of Technology,
Steyrergasse 30/II, 8010 Graz, Austria wulf@math.tugraz.at

Abstract. In this paper we consider the \textsc{Recoverable Traveling Salesman Problem} (TSP). Here the task is to find two tours simultaneously, such that the intersection between the tours is at least a given minimum size, while the sum of travel distances with respect to two different distance metrics is minimized. Building upon the classic double-tree method, we derive a 4-approximation algorithm for the \textsc{Recoverable TSP}. We also show that if the required size of the intersection between the tours is constant, a 2-approximation guarantee can be achieved, even if more than two tours need to be constructed. We discuss consequences for approximability results in the more general area of recoverable robust optimization.

Keywords: recoverable robustness; intersection constraints; traveling salesman problem; approximation

1 Introduction

Uncertainty and incomplete problem knowledge can have significant impact on the quality of decision we make. Several paradigms have been developed to include data uncertainty in the decision making process, including recoverable robustness. In this setting, we construct a first solution while data uncertainty is still present, and can later adjust this solution in a second stage, when full problem knowledge is available.

In principle, this approach can be applied to any (combinatorial) optimization problem, including the classic \textsc{Traveling Salesman Problem} (TSP). The TSP is a well-studied fundamental problem in combinatorial optimization, computer science and operations research. We denote by \((V, d)\) a TSP instance, if \(V\) is a set of vertices (cities) and \(d: \binom{V}{2} \rightarrow \mathbb{R}_{\geq 0}\) a distance function. We call \(C = (v_0, v_1, \ldots, v_n)\) a tour on \(V\), if \(|V| = n\), \(v_0 = v_n\) and \(V = \{v_0, \ldots, v_{n-1}\}\). We denote by \(E(C) := \{\{v_i, v_{i+1}\} : i = 0, \ldots, n-1\}\) the edge set of \(C\). The TSP asks for a tour \(C\) of the vertices minimizing \(d(C) = \sum_{e \in E(C)} d(e)\). The TSP is known to be NP-hard and is one of the most-studied problems with respect to
approximation algorithms. In its general form it is inapproximable, but if the distance function \(d\) is a metric on \(V\), then different constant factor approximation algorithms are known. A 2-approximation can be achieved by the classic double-tree shortcutting algorithm [16] and a \(3/2\)-approximation was achieved in the seminal work [5] introducing Christofides’ algorithm. Up until 2020 this was the best-known approximation guarantee for the general metric TSP, when a \(3/2 - \varepsilon\)-approximation was achieved [10].

In this paper, we study a variant of TSP denoted as \textsc{Recoverable Traveling Salesman Problem} (\textsc{RecovTSP}), which was introduced in the area of recoverable robust optimization [4]. We denote by \((V, d_1, d_2, q)\) a \textsc{RecovTSP} instance, if \(V\) is a set of vertices (cities), \(d_1, d_2: \binom{V}{2} \rightarrow \mathbb{R}_{\geq 0}\) are distance functions and \(q \in \mathbb{N}\) is the intersection size parameter. The \textsc{RecovTSP} asks for two tours \(C_1, C_2\) on \(V\) that minimize the objective \(d_1(C_1) + d_2(C_2)\), subject to the constraint that \(C_1\) and \(C_2\) have at least \(q\) edges in common, i.e. \(|E(C_1) \cap E(C_2)| \geq q\). If both \(d_1\) and \(d_2\) are metric distance functions the given instance is an instance of \textsc{Metric RecovTSP}.

We also study a multi-stage generalization of \textsc{RecovTSP} which we call \textsc{k-stage Recoverable Traveling Salesman Problem} (\textsc{k-St-RecovTSP}). Here, \(k \in \mathbb{N}\) is a given number of stages and an instance \((V, d_1, \ldots, d_k, q)\) of \textsc{k-St-RecovTSP} asks for \(k\) tours \(C_1, \ldots, C_k\) that minimize the objective \(\sum_{j=1}^{k} d_j(C_j)\), subject to the constraint \(|\bigcap_{1 \leq j \leq k} E(C_j)| \geq q\).

\textit{Related Results.} The study of discrete optimization problems with intersection constraints, known as recoverable optimization problems, was initiated through their application in recoverable robust optimization [13]. To define a recoverable robust problem, it is necessary to specify the set of scenarios that we wish to consider and protect against. The complexity of the resulting robust problem then depends on the choice of this uncertainty set. We refer to the survey [11] for an overview on existing complexity results, where the recoverable robust problem is denoted as robust optimization with incremental recourse. The \textsc{RecovTSP} we consider is equivalent to a recoverable robust problem with a single scenario or with interval uncertainty, an observation that will be explained more formally in Section 4.

In [12], the \textsc{Recoverable Robust Selection Problem} was considered under discrete and interval uncertainty. In the Selection Problem, the set of feasible solutions consists of all choices of exactly \(p\) out of \(n\) items. They show that in the case of discrete uncertainty, the problem becomes strongly NP-hard, while it can be solved in \(O(qn^2)\) for interval uncertainty, where \(q\) is the size of the intersection between the two solutions. Recently, [13] further improved this solution time to \(O(n)\). The setting of recoverable robustness with interval uncertainty has also been considered in the context of the spanning tree problem. In the \textsc{Recoverable Spanning Tree Problem} (\textsc{RecovST}), one is given a vertex set \(V\), two distance functions \(d_1, d_2\) on \(V\) and an intersection size parameter \(q \in \mathbb{N}\). The goal is to find two spanning trees \(T_1, T_2\) on \(V\) such that \(|T_1 \cap T_2| \geq q\) and \(d_1(T_1) + d_2(T_2)\) is minimized. In [7], it was proven that...
RECOVST can be solved optimally in polynomial time $O(qm^2n)$, where $m$ is the number of edges.

The recoverable robust setting was also considered for the Assignment Problem in [6], where the authors show W[1]-hardness and present special cases that can be solved in polynomial time. In a related single-machine Scheduling Problem setting, [1] derive a 2-approximation algorithm. The Recoverable Robust Shortest Path Problem was studied in [3], where it was shown that the problem becomes NP-hard and not approximable in most settings. The recoverable setting can also be considered for matroid bases [2]. It has been generalized to other intersection constraints in the context of matroid bases [14], where a strongly polynomial algorithm is presented for the case of a lower bound on the intersection. This setting was further generalized in [8], where also non-linear convex cost functions were considered.

So far, only little attention has been given to the RecovTSP – possibly, as the underlying problem is already hard for the non-robust setting. In the short paper [4], different solution methods were proposed for a recoverable robust setting with so-called budgeted uncertainty sets, which generalize interval uncertainty. To the best of our knowledge, no previous work has derived approximation results for this setting.

Our Contributions. This paper is the first to provide complexity results for the RecovTSP. If the required size $q$ of the intersection between the two solutions is part of the input, we show that there exists a polynomial time 4-approximation algorithm. We provide an example that shows that our analysis of the algorithm is tight. If $q$ is a constant number, an improved 2-approximation algorithm can be achieved, based on enumerating all possible intersection sets. This algorithm also extends to a more general setting, where an arbitrary number of tours need to be found, instead of only two. Finally, we discuss consequences of our results in the area of robust optimization.

2 A 4-Approximation Algorithm for the RecovTSP

2.1 Main Result and Proof Idea

In this section we prove the following result.

**Theorem 1.** There is a 4-approximation algorithm for Metric RecovTSP.

We first explain the idea of the algorithm before we describe it formally. The starting point of our algorithm is the RECOVST. Given an instance $(V, d_1, d_2, q)$ of the Recoverable TSP, we begin by obtaining an optimal solution $(T_1, T_2)$ to the RECOVST with the same parameters $(V, d_1, d_2, q)$. The trees $T_1$ and $T_2$ already have a sufficiently large intersection $|T_1 \cap T_2| \geq q$, so we would like to transform them into tours $C_1, C_2$ with the same intersection. However, there is a problem: The vertices in $T_1 \cap T_2$ could have degree greater than 2 in $T_1 \cap T_2$. For example, every component of $T_1 \cap T_2$ could be a star. But clearly in the final tours
C_1 and C_2 every vertex must have degree 2 (here it is important to note that in the \textsc{Recoverable TSP}, we do not allow a TSP tour to travel along the same edge multiple times). So T_1 \cap T_2 = E(C_1) \cap E(C_2) is not possible in general. Usually, in the TSP literature, a tree is transformed into a tour by making it Eulerian (either considering the double-tree or inserting matching edges in Christofides’ algorithm) and shortcutting an Eulerian tour of the resulting graph. However, observe that the shortcutting procedure will result in different outcomes for the trees T_1 and T_2, because they are different outside of their common intersection T_1 \cap T_2. Therefore, it is easy to see that the procedure of shortcutting does not preserve the property that T_1 and T_2 have a sufficiently large intersection. Let K_1, \ldots, K_r be the connected components of T_1 \cap T_2. As described above, the problem is that K_j is not necessarily a path for j = 1, \ldots, r. We can solve this problem by actually replacing K_j with some Hamilton path P_j that traverses all vertices of K_j. In order to find P_j, we can approximately solve the TSP problem on the sub-instance on vertex set K_j and metric d_1 + d_2. Replacing each component K_j by P_j, we obtain some new graph from T_j, which we call T_j'. We will show that if we double all the edges of T_j', we can obtain an Eulerian circuit W_j'' which contains each of the paths P_1, \ldots, P_r as simple subpaths. We finally show that it is possible to shortcut W_j'' into a tour C_i in such a way that the subpaths P_1, \ldots, P_r are preserved. Then we have that P_1, \ldots, P_r \subseteq C_i for i = 1, 2, and therefore |E(C_1) \cap E(C_2)| \geq q. We will finally show that during the whole procedure we lose at most a constant factor compared to the optimal solution of \textsc{Recoverable TSP}. Algorithm \textsc{RecovTSP} provides a description in pseudo-code and \textit{Fig. 1} depicts an example run of the algorithm.

2.2 Recoverable Spanning Trees and TSP

We are now ready to prove the correctness and the approximation guarantee of Algorithm \textsc{RecovTSP}. The key observation is that the optimal objective value of \textsc{Recoverable TSP} can be lower bounded by the optimum objective value of \textsc{Recoverable Spanning Trees}. In the following lemma we assume q \leq n; note that if q = n, then the \textsc{Recoverable TSP} is exactly the classical TSP and therefore we do not need to consider this case.

\textbf{Lemma 1.} Let (V, d_1, d_2, q) be an instance of the Metric \textsc{Recoverable TSP} and let OPT be its optimal objective value. Let T_1, T_2 be an optimal solution to the corresponding Metric \textsc{Recoverable Spanning Trees} instance (V, d_1, d_2, q). If q \leq n, it holds that d_1(T_1) + d_2(T_2) \leq OPT.

\textit{Proof.} Let C_1, C_2 be any feasible solution to the \textsc{Recoverable TSP}, i.e. they are tours on V such that |E(C_1) \cap E(C_2)| \geq q. Since q \leq n we can select e_1 \in E(C_1) and e_2 \in E(C_2) such that for T_1' := E(C_1) \setminus \{e_1\} and T_2' := E(C_2) \setminus \{e_2\} it still holds that |T_1' \cap T_2'| \geq q. Note that both T_1' and T_2' are edge sets of Hamiltionian paths on V and hence feasible solutions for the \textsc{Recoverable Spanning Trees} instance (V, d_1, d_2, q), implying d_1(T_1) + d_2(T_2) \leq d_1(T_1') + d_2(T_2') \leq d_1(C_1) + d_2(C_2) \leq OPT. \hfill \square
On the Recoverable Traveling Salesman Problem

(a) The tree $T_1$ (blue dashed edges and red edges), the tree $T_2$ (green dotted edges and red edges) and their intersection $T_1 \cap T_2$ (red edges).

(b) The tree $T'_1$ is created from $T_1$ by substituting the connected components of $T_1 \cap T_2$ with simple paths $P_1, \ldots, P_r$.

(c) The graph $T''_1$ is created from $T'_1$ by doubling all its edges.

(d) The graph $\tilde{T}_1$ is created from $T''_1$ by substituting each path $P_j$ by a special edge $\tilde{e}_j$. The arrows indicate an Eulerian circuit in $\tilde{T}_1$.

(e) We traverse $P_j$ instead of traversing $\tilde{e}_j$. This way we obtain an Eulerian circuit $W''_1$ of $T''_1$ which contains $P_1, \ldots, P_r$ as subpaths (thick edges).

(f) The final tour $C_1$ is obtained by shortcutting $W''_1$ such that the subpaths $P_1, \ldots, P_r$ are preserved.

Fig. 1: Schematic sketch of the 4-approximation for RECOV TSP.
Algorithm 1: Approximation algorithm for metric recoverable TSP.

Input: An instance \((V, d_1, d_2, q)\) of the metric recoverable TSP, where
- \(V\) is a set of vertices,
- \(d_1, d_2\) are two metric distance functions on \(V\),
- \(q \in \mathbb{N}\) is the intersection size parameter.

Output: Two tours \(C_1, C_2\) on \(V\) such that \(|C_1 \cap C_2| \geq q\).

1. \((T_1, T_2) \leftarrow\) Optimal solution of the \(\text{RecovST}\) instance \((V, d_1, d_2, q)\).
2. Set \(T'_1 \leftarrow T_1; T'_2 \leftarrow T_2; \mathcal{P} \leftarrow \emptyset\).
3. Let \(K_1, \ldots, K_r\) be the connected components of \(T_1 \cap T_2\).
4. \(\textbf{foreach} \ j \in \{1, \ldots, r\} \ \textbf{do}
5. \quad P_j \leftarrow\) Hamilton path in \(K_j\) obtained by approximating the TSP problem
   on \(K_j\) with respect to the distance function \(d_1 + d_2\) by using the
double-tree heuristic.
6. \quad Replace \(K_j\) by \(P_j\) in \(T'_1\) and \(T'_2\).
7. \quad \mathcal{P} \leftarrow \mathcal{P} \cup \{P_j\}\)
8. \(T''_i \leftarrow T'_i + T'_i\) for \(i = 1, 2\).
9. \(W''_i \leftarrow\) Eulerian circuit of the graph \((V, T''_i)\) such that all paths in \(\mathcal{P}\) are
   simple subpaths of \(W''_i\) (see Lemma 3).
10. Execute the shortcutting explained in [Lemma 4] on \(W''_i\) to obtain \(C_i\), for \(i = 1, 2\).
11. \textbf{return} \((C_1, C_2)\)

2.3 Shortcutting Common Subtrees into Paths

We aim to obtain spanning trees that allow for a shortcutting without decreasing
the size of the intersection. As already mentioned, we do this by substituting each
of the components \(K_j\) by a Hamilton path \(P_j\) in \(K_j\). Doing this substitution for
every \(j = 1, \ldots, r\) transforms \(T_i\) into \(T'_i\). The question is which paths to select.
The following lemma shows that if the double-tree heuristic is used, we obtain
an approximation guarantee of a factor 2.

**Lemma 2.** If for each \(j = 1, \ldots, r\) the path \(P_j\) is computed using the double tree
heuristic on \(K_j\) with respect to \(d_1 + d_2\), then \(d_1(T'_1) + d_2(T'_2) \leq 2(d_1(T_1) + d_2(T_2))\).

**Proof.** First, observe that the tree \(T_1 \cap T_2 \cap K_j\) is actually already a minimum
spanning tree of \(K_j\), with respect to the metric \(d_1 + d_2\). This fact together with the
choice of \(P_j\) proves that \((d_1 + d_2)(P_j) \leq 2(d_1(T_1 \cap T_2 \cap K_j))\).
By the definition of \(T'_i\) we have
\[
  d_1(T'_1) + d_2(T'_2) = \left(\sum_{j=1}^{r} (d_1 + d_2)(P_j)\right) + d_1(T_1 \setminus T_2) + d_2(T_2 \setminus T_1)
\]
\[
  \leq 2(d_1(T_1) + d_2(T_2)).
\]
Given the trees $T'_1, T'_2$ and the pairwise vertex-distinct paths $\mathcal{P}$ such that \( \bigcup_{P \in \mathcal{P}} E(P) = T'_1 \cap T'_2 \), we let $T''_i := T'_i + T'_i$ be the multiset which contains every edge of the edge set $T'_i$ exactly twice. The following lemma shows how to obtain an Eulerian cycle $W''_i$ in the (multi-)graph $(V, T''_i)$ such that each $P \in \mathcal{P}$ is a subpath of $W''_i$ for $i = 1, 2$.

**Lemma 3.** For $i = 1, 2$, let $T''_i$ be the tree obtained in line 7 of Algorithm 1. There exists an Eulerian tour $W''_i$ in the graph $(V, T''_i)$ such that all paths $P \in \mathcal{P}$ are subpaths of $W''_i$.

**Proof.** For both $i = 1, 2$, we construct a new graph $(V, \tilde{T}_i)$ by first copying the graph $(V, T''_i)$. Then, for each path $P_j \in \mathcal{P}$ such that $P_j = (v_1, \ldots, v_\ell)$ we remove for each $t = 1, \ldots, \ell - 1$ one of the two copies of the edge $\{v_t, v_{t+1}\}$ from $\tilde{T}_i$ and add the special edge $\tilde{e}_j = \{v_\ell, v_1\}$ to $\tilde{T}_i$. It then still holds that $(V, \tilde{T}_i)$ is Eulerian, since each vertex degree stays even and the graph remains connected. Hence, there exists an Eulerian circuit $\tilde{W}_i$ in $(V, \tilde{T}_i)$. This circuit $\tilde{W}_i$ traverses for each $P_j \in \mathcal{P}$ the previously added special edge $\tilde{e}_j$. We construct $W''_i$ by replacing for each $P_j \in \mathcal{P}$ the edge $\tilde{e}_j$ by the path $P_j$, traversed in the corresponding direction. Note that, as claimed, $W''_i$ is an Eulerian tour in $(V, T''_i)$ with each $P \in \mathcal{P}$ as a subpath. \qed

### 2.4 Shortcutting Without Skipping Paths

**Lemma 4.** Let $W''$ be a closed walk on $V$ and $\mathcal{P}$ be a set of pairwise vertex-distinct subpaths of $W''$. Then, for any metric $d$, there exists a tour $C$ on $V$ such that $d(C) \leq d(W'')$ and $C$ contains all paths $P \in \mathcal{P}$ as subpaths.

**Proof.** We iteratively construct the tour $C$, by following the closed walk $W'' = (v_0, \ldots, v_m)$ on $V$. Note that without loss of generality $v_0 = v_m$ is not an inner vertex of any path $P \in \mathcal{P}$. We follow a strategy similar to the classic shortcutting applied in the double tree and Christofides’ approximation algorithms. This means that for all vertices that are not contained in any path in $\mathcal{P}$, we add them to $C$ the first time the walk $W''$ visits them. Otherwise, they are shortcut, i.e. not added to $C$. To ensure that no edges of paths in $\mathcal{P}$ are shortcut, whenever the closed walk $W''$ visits a vertex $v$ of any path $P \in \mathcal{P}$ and $W''$ is currently not in the first full transversal of $P$, we shortcut this detour to $v$. This ensures that at the point when the closed walk $W''$ traverses the path $P$ for the first time, it holds that none of the vertices in $P$ are yet visited in the current subtour $C$, hence the whole path $P$ is traversed by $C$. At the end of the process, we close the constructed path $C$ by adding $v_0$.

Any vertex in $V$ appears in $C$, since we only shortcut a vertex $v$ if it is already previously visited by the current subtour, or if it occurs in the closed walk $W''$ before its later occurrence as part of the first transversal of a path $P \in \mathcal{P}$ with $v \in P$. Also, every vertex $v \in V$ appears in $C$ exactly once, since $v$ also appears in $W''$ and we shortcut every time $v$ is revisited by $W''$. Hence, $C$ is a tour on $V$. Finally, note that any edge $\{v, w\} \in E(C)$ corresponds to an edge-distinct
subwalk \( P_{v,w} \) of \( W'' \), connecting \( v \) to \( w \). Hence, by the triangle inequality we have \( d(C) \leq d(W'') \).

The application of [Lemma 4 in Algorithm 1](#) transforms the closed walks \( W_i'' \) into tours \( C_i \) for \( i = 1, 2 \) such that it holds that \( d_1(C_1) + d_2(C_2) \leq d_1(W_i'') + d_2(W_i'') \) and \( |E(C_1) \cap E(C_2)| \geq |\bigcup_{j=1}^q E(P_j)| \geq q \). Using the results of the preceding sections we are now ready to derive an approximation guarantee for [Algorithm 1](#).

**Proof (Proof of Theorem 1).** By construction, we know that both \( C_1, C_2 \) are tours and \( |E(C_1) \cap E(C_2)| \geq |\bigcup_{j=1}^q E(P_j)| = |T_1 \cap T_2| = |T_1 \cap T_2| \geq q \), hence the tours \( C_1, C_2 \) are a feasible solution to RECOVTSP. It also holds that

\[
d_1(C_1) + d_2(C_2) \leq d_1(W_1'') + d_2(W_2'')
\leq 2(d_1(T_1) + d_2(T_2)) \leq 4(d_1(T_1) + d_2(T_2)) \leq 4\text{OPT}.
\]

Note that all steps of [Algorithm 1](#) can be implemented in polynomial time. \(\square\)

### 2.5 An Example where the 4-Approximation is Tight

We have successfully shown that Algorithm 1 provides a 4-approximation to the RECOVTSP. Inspired by ideas for the tightness of the double-tree algorithm [9], we show:

**Lemma 5.** There exist problem instances such that Algorithm 1 can return a solution which is (asymptotically) 4 times worse than the optimal solution, even if both \( d_1, d_2 \) are 2-dimensional Euclidean metrics.

**Proof.** Let \( k \geq 2 \) be a fixed integer. We describe a problem instance \((V, d_1, d_2, q)\) of RECOVTSP. For each vertex \( v \in V \), we assign a position \( p_v^1 \in \mathbb{R}^2 \) and a possibly different position \( p_v^2 \in \mathbb{R}^2 \) to it. We define \( d_i(x,y) := \|p_i^x - p_i^y\|_2 \) for \( i = 1, 2 \). Clearly \( d_1 \) and \( d_2 \) are Euclidean metrics. Let \( 0 < \varepsilon < 1/k^2 \) be some small quantity. A so-called satellite gadget is depicted in [Fig. 2a](#). It is a gadget around some central vertex \( v \) such that \( v \) is surrounded by eight additional vertices. The position of these eight vertices in relation to the position of \( v \) is exactly like depicted in [Fig. 2a](#). Here, the dashed lines between vertices symbolize their respective horizontal or vertical distance. The vertices \( v_1, \ldots, v_4 \) are called satellites. For every satellite \( w \) we have \( p_w^1 \neq p_w^2 \) and the position \( p_w^2 \) is like specified in [Fig. 2b](#). For every vertex \( w \) which is not a satellite, we have \( p_w^1 = p_w^2 \).

Now the actual problem instance of RECOVTSP is created by considering the \( 2 \times k \) regular unit grid on the grid points \((i,j)_{i=1,2, j=1,\ldots, k}\). [Fig. 2](#) depicts the case where \( k = 3 \). On each grid point, we place a copy of the satellite gadget. Additionally, we introduce \( 2k - 1 \) additional so called helper-vertices at distance \( 3\varepsilon \) from the grid points. The instance is depicted in [Figs. 2c and 2d](#). Helper vertices are marked with a cross. Finally, we let \( q := 12k - 1 \). This completes our description of the RECOVTSP instance \((V, d_1, d_2, q)\). The following observations can now be readily made:
On the Recoverable Traveling Salesman Problem

Fig. 2: An example instance where the 4-approximation is tight.
The unique minimum spanning tree $T_1$ ($T_2$) with respect to $d_1$ ($d_2$) is depicted in [Fig. 2c] [Fig. 2d]. (The role of the helper vertices is to make the minimum spanning tree unique.)

The intersection $T_1 \cap T_2$ is depicted in [Fig. 2e]. We have $|T_1 \cap T_2| = q$ and $(T_1, T_2)$ is the optimal solution to the RECOVST.

The path $P$ depicted in [Fig. 2f] is a possible outcome when running line 5 of Algorithm 1.

[Fig. 2g] depicts the corresponding Eulerian graph $T_1''$.

The shortcutting procedure can run in such a way that for metric $d_1$ the tour $C_1$ depicted in [Fig. 2h] is the output of Algorithm 1. We have $d_1(C_1) = (1 + o(1))(8k - 4)$.

Analogously, the shortcutting procedure can run in such a way that for metric $d_2$ some tour $C_2$ is output such that $d_2(C_2) = (1 + o(1))(8k - 4)$.

On the other hand, consider the tour $C$ depicted in [Fig. 2i]. We might have $d_1(C) \neq d_2(C)$, but still we have $d_1(C) = (1 + o(1))2k$ and $d_2(C) = (1 + o(1))2k$. If we let $C'_1 := C'_2 := C$, then $|E(C'_1) \cap E(C'_2)| = |V| \geq q$. So $(C'_1, C'_2)$ is a solution to the RECOVSTP of value $(1 + o(1))4k$. This is asymptotically a factor 4 better than $d_1(C_1) + d_2(C_2)$. This proves the lemma. \hfill \Box

### 2.6 Pitfalls when Applying Christofides’ Algorithm

We give two short remarks which show that some trivial ideas to modify Algorithm 1 do not work. Hence there are likely new ideas needed to obtain an approximation guarantee better than 4.

**Remark 1.** We remark that the bound provided in [Lemma 2] is tight. We give an example where this is the case: Assume that the intersection $T_1 \cap T_2$ is a star on the vertex set $U$ with $n + 1$ vertices such that some vertex $u_0$ is the center of the star. Assume furthermore that for all vertices $x, y \in U$ we have $d_1(x, y) = d_2(x, y)$ and that the metric used on $U$ is the Paris railway metric: Here we have $d_1(x, y) = 0$ if $x = y$, otherwise $d_1(x, y) = 1$ if $x = u_0$ or $y = u_0$, and otherwise $d_1(x, y) = 2$ for $i = 1, 2$. Furthermore assume that the intersection $T_1 \cap T_2$ makes up almost all of the cost of $T_1$ and $T_2$, that is $d_1(T_1) + d_2(T_2) = d_1(T_1 \cap T_2) + d_2(T_1 \cap T_2) + \varepsilon$ for some small $\varepsilon > 0$. Then $d_1(T_1) + d_2(T_2) = 2n + \varepsilon$.

On the other hand, every Hamilton path in $U$ has cost at least $4n - 4$. Therefore we will have $d_1(T'_1) + d_2(T'_2) \geq 4n - 4$, independent of which path $P_j$ will be picked in order to replace $T_1 \cap T_2$. This example shows that even though there are better approximation algorithms known than the double-tree heuristic, using these algorithms instead of the double-tree heuristic in line 5 of Algorithm 1 does not yield a better approximation guarantee than a factor of 2 for [Lemma 2].

**Remark 2.** Because the graph $(V, T''_i)$ is the double-tree of the graph $(V, T'_i)$, an approximation factor of 2 is introduced. One could also ask whether one can apply Christofides’ algorithm to obtain some Eulerian graph $T'''_i$ from $T'_i$ plus a matching, and therefore only introduce a factor of $3/2$. However, this idea does not work: If one analogously transforms $T'''_i$ into $T_i$, then one can see that even
though all vertices in $\tilde{T}_i$ have even degree, one can find examples where $\tilde{T}_i$ is not connected. In general, one can show that there exist instances such that the cost of a minimal tour which includes all the paths $P_1, \ldots, P_r$ as subpaths is strictly larger than the cost of $T_i$ plus a matching. This shows that Christofides' algorithm cannot trivially be applied in order to improve our approximation guarantee.

3 A 2-Approximation for Constant Intersection Size

We now consider the setting where the required size $q$ of the intersection set is a constant number. We show that there exists a 2-approximation algorithm that can be applied to the more general $k$-St-RECOV-TSP, where $k$ tours $C_1, \ldots, C_k$ with intersection size $q$ need to be constructed.

The corresponding $k$-Stage RECOVST is NP-hard according to [14]. Hence, it is not possible to use the same approach as in Section 2 to obtain a constant factor approximation algorithm for the $K$-St-RECOV-TSP.

**Theorem 2.** For constant $q$ and arbitrary $k$, there exists a 2-approximation algorithm for METRIC $k$-St-RECOV-TSP.

**Proof.** This result is obtained by guessing the optimal intersection of the $k$ tours by checking all $\binom{n}{q}$ possibilities for subsets of pairwise vertex-disjoint paths. These sets of pairwise vertex-disjoint paths can then be extended to spanning trees $T'_1, \ldots, T'_k$ by solving $k$ instances of the minimum spanning tree problem. Then, the algorithm can proceed in a similar way as Algorithm 1, with the only difference that instead of performing each operation twice we now have to perform them $k$ times. By similar arguments as in Lemma 1 it holds that $\sum_{i=1}^k d_i(T'_k) \leq \text{OPT}$. Using this, we can proceed as in Algorithm 1 line 9 to obtain the tours $C_1, \ldots, C_k$, for which the cost can be bounded by $\sum_{i=1}^k d_i(W'_i) \leq 2 \sum_{i=1}^k d_i(T'_i) \leq 2\text{OPT}$. \qed

4 Implications for Recoverable Robust Optimization

We now discuss the implications of our approximation results for recoverable robust optimization problems. Formally, let $X \subseteq \{0, 1\}^E$ denote the set of feasible solutions for some combinatorial optimization problem over ground set $E$, let $U \subseteq \mathbb{R}^E$ denote a set of cost scenarios, and let $X^k(x) = \{y \in X : d(x, y) \leq k\}$ denote the set of second-stage recovery solutions for some given first-stage solution $x$, where $d$ denotes some measure of distance. The recoverable robust problem is to solve $\min_{x \in X} \max_{c \in U} \min_{y \in X^k(x)} \sum_{e \in E} C_e x_e + c_e y_e$, see, e.g., the definition given in [11]. If $U$ is the Cartesian product of intervals, i.e., $U = \{c \in \mathbb{R}^E : c_e \in [l_e, u_e] \forall e \in E\}$, then an optimal solution to the inner maximization problem is to choose all cost coefficients to be at their upper bound $u_e$. This means that the recoverable robust problem considers only a single scenario, which is equivalent to the recoverable problem setting considered in this paper. Therefore, our
approximation results hold for the RECOVERABLE ROBUST TSP with interval uncertainty.

Other uncertainty sets are considered as well, including budgeted uncertainty (see, e.g., [4], where budgeted uncertainty sets were used for the TSP). Budgeted uncertainty sets are essentially interval sets with an additional constraint on the total amount of deviation. Different variants have been proposed in the literature. In [7], the following sets were used:

\[ U_{\Gamma}^1 = \{ c \in \mathbb{R}^E : c_e = \ell_e + (u_e - \ell_e) \delta_e, \, \delta_e \in \{0, 1\} \forall e \in E, \sum_{e \in E} \delta_e \leq \Gamma \}, \]

\[ U_{\Gamma}^2 = \{ c \in \mathbb{R}^E : c_e = \ell_e + \delta_e, \, \delta_e \in [0, u_e - \ell_e] \forall e \in E, \sum_{e \in E} \delta_e \leq \Gamma \}. \]

They showed the following result in the context of the RecovST, which also holds for any other combinatorial optimization problem. If \( \alpha \in (0, 1] \) is such that \( \ell_e \geq \alpha u_e \) for all \( e \in E \), then an optimal solution to the recoverable problem with respect to costs \( \ell_e \) is an \( 1/\alpha \) approximation for the recoverable robust problem with respect to \( U_{\Gamma}^1 \) or \( U_{\Gamma}^2 \). By a straightforward modification of their proof of [7, Lemma 6], one can derive a \( 4/\alpha \)-approximation algorithm for the recoverable robust TSP with budgeted uncertainty if \( q \) is part of the input (and \( 2/\alpha \) if \( q \) is constant) using our results.

5 Conclusions

Recoverable combinatorial optimization problems are a natural generalization of classic problems that arise in the area of robust optimization. In this paper, we considered the RECOVERABLE TRAVELING SALESMAN PROBLEM, where two tours with respect to two distance functions need to be constructed, minimizing the sum of distances, such that the size of their intersection is at least a prescribed number \( q \). Building upon the classic double-tree approximation idea, we showed that it is possible to transform an optimal solution of the RecovST, which can be solved in polynomial time, into a feasible solution for RecovTSP with an objective value that is at most 4 times the optimum. We provided an example that shows that the analysis of this algorithm is tight, and gave an intuition why it is not possible to apply Christofides’ algorithm while using the same algorithmic ideas. Furthermore, we considered the case that \( q \) is a constant, which allows for a stronger and easier 2-approximation algorithm, which can also be applied if more than two tours need to be constructed.

In further research, stronger approximation results are likely to exist. More specialized cases in the distance structure also seem fruitful to consider, such as the planar Euclidean case, or distance matrices with the Monge and anti-Monge property. Finally, it would be of interest to study the approximability of METRIC RECOVERABLE ASSIGNMENT or MATCHING PROBLEMS.

Funding. The authors acknowledge partial support by the Field of Excellence "COLIBRI" at the University of Graz, Deutsche Forschungsgemeinschaft (DFG) through grant GO 2069/1-1 and the Austrian Science Fund (FWF): W1230.
References

1. Bold, M., Goerigk, M.: Recoverable robust single machine scheduling with interval uncertainty. arXiv preprint arXiv:2107.09310 (2021)
2. Büsing, C.: Recoverable robustness in combinatorial optimization. Cuvillier Verlag (2011)
3. Büsing, C.: Recoverable robust shortest path problems. Networks 59(1), 181–189 (2012)
4. Chassein, A., Goerigk, M.: On the recoverable robust traveling salesman problem. Optimization Letters 10(7), 1479–1492 (2016)
5. Christofides, N.: Worst-case analysis of a new heuristic for the travelling salesman problem. Tech. rep., Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group (1976)
6. Fischer, D., Hartmann, T.A., Lendl, S., Woeginger, G.J.: An investigation of the recoverable robust assignment problem. arXiv preprint arXiv:2010.11456 (2020)
7. Hradovich, M., Kasperski, A., Zieliński, P.: Recoverable robust spanning tree problem under interval uncertainty representations. Journal of Combinatorial Optimization 34(2), 554–573 (2017)
8. Iwamasa, Y., Takazawa, K.: Optimal matroid bases with intersection constraints: Valuated matroids, m-convex functions, and their applications. Mathematical Programming pp. 1–28 (2021)
9. Johnson, D., Papadimitriou, C.: Performance guarantees for heuristics. In: Lawler, E., Lenstra, J., Kan, A.R., Shmoys, D. (eds.) The Traveling Salesman Problem: a guided tour of Combinatorial Optimization. Wiley, Chichester (1985)
10. Kärlin, A.R., Klein, N., Gharan, S.O.: A (slightly) improved approximation algorithm for metric TSP. In: Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing, pp. 32–45 (2021)
11. Kasperski, A., Zieliński, P.: Robust discrete optimization under discrete and interval uncertainty: A survey. In: Robustness analysis in decision aiding, optimization, and analytics, pp. 113–143. Springer (2016)
12. Kasperski, A., Zieliński, P.: Robust recoverable and two-stage selection problems. Discrete Applied Mathematics 233, 52–64 (2017)
13. Lachmann, T., Lendl, S., Woeginger, G.J.: A linear time algorithm for the robust recoverable selection problem. Discrete Applied Mathematics 303, 94–107 (2021)
14. Lendl, S., Peis, B., Timmermans, V.: Matroid bases with cardinality constraints on the intersection. Mathematical Programming pp. 1–24 (2021)
15. Liebchen, C., Lübbebecke, M., Möhring, R., Stiller, S.: The concept of recoverable robustness, linear programming recovery, and railway applications. In: Robust and online large-scale optimization, pp. 1–27. Springer (2009)
16. Rosenkrantz, D.J., Stearns, R.E., Lewis, II, P.M.: An analysis of several heuristics for the traveling salesman problem. SIAM journal on computing 6(3), 563–581 (1977)