RAIRO-Theor. Inf. Appl. 48 (2014) 479–494
DOI: 10.1051/ita/2014021

Available online at:
www.rairo-ita.org

IMPROVED APPROXIMATIONS FOR ORDERED TSP ON NEAR-METRIC GRAPHS *, **

Hans-Joachim Böckenhauer[1] and Monika Steinová[1]

Abstract. The traveling salesman problem is one of the most important problems in operations research, especially when additional precedence constraints are considered. Here, we consider the well-known variant where a linear order on k special vertices is given that has to be preserved in any feasible Hamiltonian cycle. This problem is called Ordered TSP and we consider it on input instances where the edge-cost function satisfies a β-relaxed triangle inequality, *i.e.*, where the length of a direct edge cannot exceed the cost of any detour *via* a third vertex by more than a factor of $\beta > 1$. We design two new polynomial-time approximation algorithms for this problem. The first algorithm essentially improves over the best previously known algorithm for almost all values of k and $\beta < 1.087889$. The second algorithm gives a further improvement for $2n \geq 11k + 7$ and $\beta < \sqrt[3]{4/3}$, where n is the number of vertices in the graph.

Mathematics Subject Classification. 68Q25, 68R10.

1. Introduction

One of the most prominent problems in operations research is the famous Traveling Salesman Problem (TSP). This problem is studied under many generalizations and modifications [14]. It is well known that TSP is very hard to approximate, even very restricted special cases remain $\mathcal{APX}$-hard. If no restrictions are given

Keywords and phrases. Approximation algorithm, near-metric ordered TSP, relaxed triangle inequality.

* *This work was partially supported by SNF grant 200021-132510/1.*

** *An extended abstract of this work has been presented at SOFSEM 2013 [6].*

[1] Department of Computer Science, ETH Zurich, Switzerland.
{hjb,monika.steinova}@inf.ethz.ch

Article published by EDP Sciences

© EDP Sciences 2015

https://doi.org/10.1051/ita/2014021 Published online by Cambridge University Press

on the edge-cost function, the problem is not approximable by any polynomial (dependent on the input size), see for instance [15]. On the other hand, if the edge costs satisfy the standard triangle inequality, the problem becomes approximable within a factor 1.5 due to Christofides' algorithm [11]. If the metrics is even more restricted and the vertices are placed into a fixed-dimension Euclidean space, the problem turns out to have a PTAS [3]. For a brief overview of the different notions of hardness and approximability of problems, we refer, *e.g.*, to [15].

In many applications, additional restrictions are given for the set of feasible solutions, one of the most common type of such restrictions are precedence constraints, *i.e.*, demanding that certain vertices have to be visited before certain others. In general, such precedence constraints can be expressed by an arbitrary partial ordering on the vertices. However, in this paper, we focus on instances where a linear order on a subset of k vertices is given as a part of the input and every feasible solution has to contain these special vertices in the prescribed order. We call this variant of the problem Ordered TSP or k-OTSP for short and its study was initialized by [8].

As we have discussed above, the properties of the edge-cost function (called $cost(\cdot)$) can influence the hardness of a problem. Here, we consider the β-relaxed triangle inequality that can be formulated as $cost(uv) \leq \beta(cost(uw) + cost(wv))$, where u, v, w are arbitrary vertices of the graph and $\beta \geq 1$ is a constant. For $\beta = 1$, this yields the well-known standard triangle inequality. If $\beta > 1$, the inequality is referred to as β-relaxed triangle inequality. The instances where the standard or relaxed triangle inequality holds are called *metric* or *near-metric* instances, respectively. By the relaxed triangle inequality, we model how close the edge-cost function is to being metric. The cases where β is close to 1 (and hence the edge-cost function almost satisfies a triangle inequality) are most interesting for the following two reasons. First of all, many real world instances tend to have β equal or close to 1 which supports the study of these cases. Secondly, from the theoretical viewpoint, the study of such instances allows us to better understand how the triangle inequality influences the hardness of problems and the stability of approximation algorithms [13, 15].

As TSP is one of the most prominent problems in complexity theory, its near-metric instances were studied in [1, 2, 5, 7]. In the metric case, the best known approximation algorithm for k-OTSP with approximation ratio $2.5 - 2/k$ is due to [9, 10]. In this paper, we focus on the version of k-OTSP where a β-relaxed triangle inequality holds which is denoted as k-Δ_βOTSP. In [8], the first polynomial-time approximation algorithm solving k-Δ_βOTSP with ratio

$$(k+1) \cdot \beta^{1+\lceil \log_2(k-1) \rceil} \cdot \min\{4\beta, 1.5\beta^2, \beta^2 + \beta\}$$

is presented. A slightly different approach was used later in [9] to improve the approximability of k-Δ_βOTSP to ratio $k\beta^{\lceil \log_2(3k-3) \rceil}$ and further to $k\beta^{\lceil \log_2(\lfloor 3k/2 \rfloor + 1) \rceil}$ in the journal version [10]. Note that, in the original results in these papers, the ceiling function was omitted, but to be exact, the function has to be used.

https://doi.org/10.1051/ita/2014021 Published online by Cambridge University Press

In this paper, we introduce several new observations that allow us to improve the original approximation algorithm for k-Δ_βOTSP from [8]. Our improved polynomial-time approximation algorithm yields a ratio of $3/2 \cdot \lceil k/2 \rceil \cdot \beta^{2+\lceil \log_2(2\lceil k/2 \rceil + k - 3) \rceil}$ which is better than the currently best known approximation algorithm for all instances with $\beta \leq (3/4)^{1/\log_2(3/32)} < 1.0878898$, except for those with very small k, see Figure 3. The pairs (β, k) for which we improved the approximation ratio are depicted in Figure 3. By a more involved analysis, for instances where the number of vertices is at least $5.5k + 3.5$, we achieve a further improvement even for $\beta < \sqrt[3]{4/3} \approx 1.12651$. We believe that some of our observations can be generalized to other problems where the β-relaxed triangle inequality holds.

The paper is organized as follows. In Section 2, we give the formal definition of the problem and fix our notation; in Section 3 we present our approximation algorithm for k-Δ_βOTSP. Section 4 discusses the more involved analysis of our algorithm and in Section 5 we conclude the paper with open problems.

2. Preliminaries

In this paper, we use standard notions of graph theory [12,16]. For an undirected graph $G = (V, E)$, the edge representation $\{u, v\}$ is simplified into an unordered pair uv $(u, v \in V)$. The set of vertices and the set of edges of a graph G are denoted as $V(G)$ and $E(G)$, respectively. A list of vertices $v_0, \ldots, v_n$, where each vertex is used at most once and such that $v_i v_{i+1} \in E$ $(0 \leq i < n)$, is called a *path* with the *endpoints* v_0 and v_n. The length of such path is n. Once the endpoints of a path are identical, *i.e.*, $v_0 = v_n$, we speak about a *cycle* of length n. A *Hamiltonian cycle* of a graph G is a cycle of length $|V(G)|$. A graph consisting of all possible edges is referred to be *complete*. In a graph G, a *subpath*[2] of a path $P = v_0, \ldots, v_n$ is a path $v_{i_0}, \ldots, v_{i_k}$, where $i_0 < i_1 < \ldots < i_k$ and $v_{i_j} v_{i_{j+1}} \in E(G)$ for all $0 \leq j < k$. We refer to the edges $v_{i_j} v_{i_{j+1}}$ of a subpath to be a *bypass* or a *jump* and the vertices between v_{i_j} and $v_{i_{j+1}}$ (exclusively) in the path P to be *bypassed* by the edge $v_{i_j} v_{i_{j+1}}$.

The *cost* of a subgraph of a graph is the sum of the costs of its edges. For simplicity, we write *cost*(H) when we mean the cost of a subgraph H. The graphs which we refer to through out this paper are always complete and weighted.

The problem of finding a minimum-cost Hamiltonian cycle in a near-metric complete undirected graph is called *Near-Metric Traveling Salesman Problem* and is referred to as Δ_β-TSP. If we fix, in addition, an ordered sequence of k *special vertices* that have to be present in the minimum-cost Hamiltonian cycle in the given order, we speak about the *Ordered Traveling Salesman Problem*. If the underlying graph is near-metric, we refer to the problem as the *Ordered Near-metric Traveling Salesman Problem* and we denote it as k-Δ_βOTSP.

[2]Note that our notion of a subpath is a bit non-standard. The sequence of vertices in our subpath is a (not necessarily contiguous) subsequence of vertices in the path.

https://doi.org/10.1051/ita/2014021 Published online by Cambridge University Press

Note that k-Δ_βOTSP with $k \leq 3$ turns into the standard Near-metric Traveling Salesman Problem. Hence, in this paper, we assume that $k \geq 4$.

For the Near-metric Traveling Salesman Problem, we know three different approximation algorithms, each is best for certain values of β.

Theorem 2.1 (Andreae [1], Bender and Chekuri [5], Böckenhauer *et al.* [7])**.** *There is a polynomial-time approximation algorithm solving Δ_β-TSP with approximation ratio*

$$M := \min\left\{\frac{3}{2}\beta^2, \beta^2+\beta, 4\beta\right\}.$$

Furthermore,

$$M = \begin{cases} \frac{3}{2}\beta^2 & \text{if} \quad \beta \leq 2, \\ \beta^2+\beta & \text{if} \quad 2 \leq \beta \leq 3, \\ 4\beta & \text{if} \quad \beta \geq 3. \end{cases}$$

To be able to estimate the cost of bypasses of a path in near-metric complete graphs, we recall a very basic lemma.

Lemma 2.2 (Bandelt *et al.*, [4])**.** *Let G be a complete graph with edge-cost function cost that satisfies the β-relaxed triangle inequality. Let $W = v_0, \ldots, v_\ell$ be a path in G. For $0 = a_0 < a_1 < \ldots < a_q = \ell$, where $1 \leq q \leq \ell$ holds, let $m := \max_{0 \leq i < q}\{a_{i+1} - a_i\}$. Then*

$$\sum_{i=0}^{q-1} cost(v_{a_i}, v_{a_{i+1}}) \leq \beta^{\lceil \log_2 m \rceil} \cdot cost(W).$$

Intuitively, this lemma says that the length of a jump over a path of length ℓ can be bounded from above by $\beta^{\lceil \log_2 \ell \rceil}$ times the cost of the bypassed path.

The algorithm that we present in the following section builds on the first known approximation algorithm for k-Δ_βOTSP designed by Böckenhauer *et al.* in [8] as Algorithm 2. Its general idea is as follows. First, connect all vertices into a Hamiltonian cycle. Secondly, color the non-special vertices clockwise as they appear in the Hamiltonian cycle cyclically by $k+1$ colors. Then connect pairs of the consecutive special vertices by clockwise traversing the Hamiltonian cycle and by including the vertices of just one fixed color (for each pair of vertices a different one) in between the two special vertices to a subpath. Use another subpath to connect the last special vertex s_k with a vertex x which was not picked to any subpath yet and is the last vertex located on the Hamiltonian cycle before the first special vertex s_1. Connect these vertices *via* a subpath composed of vertices of color k. Finally, start the last subpath in the vertex x and traverse the entire cycle to the first special vertex s_1 by using the remaining vertices not yet included to the subpaths. In this paper we refer to this algorithm as HC-Algorithm.

A couple of years after the HC-Algorithm was introduced, the currently best known approximation algorithm was presented in [9] and a slightly improved version in [10]. The algorithm is based on a different underlying structure. Instead of

https://doi.org/10.1051/ita/2014021 Published online by Cambridge University Press

a Hamiltonian cycle, a cheaper minimum spanning tree is used as an underlying structure. The vertices of the tree are colored again by k colors and the special vertices are then connected by the unique paths in the tree *via* vertices of fixed colors. The vertices not lying on these paths are in the end attached to the cycle using some special rules and a feasible solution is obtained. In the original paper a ceiling function was omitted and hence we restate the achieved result in its correct form in the following theorem.

Theorem 2.3 (Böckenhauer *et al.*, [10]). *There exists a* $k \cdot \beta^{\lceil \log_2(\lfloor 3k/2 \rfloor + 1)\rceil}$*-approximation algorithm that solves* k*-*Δ_βOTSP *in polynomial time.*

3. Building on an underlying Hamiltonian cycle

As a stepping stone to improve the approximability of k-Δ_βOTSP ($k \geq 4$), we use the HC-Algorithm. We modify this algorithm by incorporating three new observations, which yields Algorithm 1.

The general idea of both HC-Algorithm and Algorithm 1 is the same. We first build a Hamiltonian cycle C that does not respect the given order of the special vertices. The cycle is then used to construct paths between consecutive special vertices that are then merged together into a cycle where the special vertices are present in the right order. However, during the process of the path construction, some vertices might be omitted and hence are not present anymore in the new cycle. These vertices are in the end added into the cycle and the required Hamiltonian cycle is obtained.

The three observations that are improving HC-Algorithm are as follows.

- We can always connect three special vertices present on a cycle in the prescribed order by a single path – the path starts in the first vertex and traverses, either clockwise or counterclockwise, the cycle through the second prescribed vertex to the last vertex.
- The vertices that are bypassed by a subpath, but not included into any subpath, can be just included to the bypassing subpath. This does not increase the estimated cost of the solution as in our calculations the costs of all bypasses are estimated by the cost of the underlying path, and thus longer bypasses are always estimated as more expensive due to the relaxed triangle inequality.
- Let H be a Hamiltonian cycle on a complete weighted graph and let $L_0, \ldots, L_{p-1}$ be some paths on H that can be sequentially merged in their endpoints into a walk W, *i.e.*, for all $0 \leq i < p$, paths L_i and $L_{(i+1) \bmod p}$ share one endpoint. Then the vertices of H that are not covered by W (if there are any) form a single path Y of H. Moreover, the adjacent vertex of an endpoint of Y in H is a vertex in which two paths L_i and L_{i+1} are merged. The path Y split into two subpaths can be used to extend the paths L_i and L_{i+1} such that all vertices of H are covered. (The situation on the path named as L'_x is depicted in the upper part of Fig. 2).

https://doi.org/10.1051/ita/2014021 Published online by Cambridge University Press

Algorithm 1. Approximation algorithm for k-Δ_βOTSP.

Input: A complete graph $G = (V, E)$ with edge cost function $cost : E \to \mathbb{Q}^+$ that satisfies the β-relaxed triangle inequality ($\beta > 1$), and a sequence of special vertices $(s_1, \ldots, s_k)$ from V ($k \geq 4$).

1: Using some constant-approximation algorithm, construct an approximate Δ_β-TSP solution C on $(G, cost)$, disregarding the order on $(s_1, \ldots, s_k)$.
2: Let P be one of the two paths that may be obtained by removing one of the edges incident with s_1 in C, and let $W = (w_1, \ldots, w_{n-k})$ be the sequence of the non-special vertices in P, beginning with the non-special vertex closest to s_1 in P. Let f be a $\lceil k/2 \rceil$-cyclic coloring of non-special vertices of C defined as follows: $f : V \to \{-1, 0, \ldots, \lceil k/2 \rceil - 1\}$ where $f(s_i) := -1$, for all $1 \leq i \leq k$, and $f(w_{i+1}) := i \bmod \lceil k/2 \rceil$, for all $0 \leq i < n - k$.
3: For $1 \leq i \leq \lceil k/2 \rceil - 1$, let L_{i-1} be the subpath in C (either clockwise or counterclockwise) from s_{2i-1} to s_{2i} and then to s_{2i+1}, restricted to s_{2i-1}, s_{2i}, and s_{2i+1} plus all vertices w with $f(w) = i - 1$ in between s_{2i-1} and s_{2i+1}.
4: If k is odd, let $L_{\lceil k/2 \rceil - 1}$ be the subpath in C from s_k to s_1 that is restricted to s_k, s_1 and the non-special vertices w that have color $f(w) = \lceil k/2 \rceil - 1$.
If k is even, let $L_{\lceil k/2 \rceil - 1}$ be the subpath in C from s_{k-1} through s_k to s_1, restricted to s_{k-1}, s_k, s_1 and the non-special vertices w with $f(w) = \lceil k/2 \rceil - 1$.
5: Create new subpaths $L'_0, \ldots, L'_{\lceil k/2 \rceil - 1}$ from $L_0, \ldots, L_{\lceil k/2 \rceil - 1}$ by including the non-special vertices bypassed by all L_i as shown in Figure 1 and discussed later.
6: Let Y be the path of C containing all the vertices that are not included in any of the L'_i. If such a path does not exist, skip this step. Otherwise, let x and y be the two endpoints of Y and let $s_{2\gamma+3}$ be the special vertex that is neighboring with x in C (see Fig. 2 for details).
Remove the endpoint $s_{2\gamma+3}$ from the subpath L'_γ and extend it by including all the vertices w of Y in the direction from x to y colored by $f(w) = \gamma$ and terminate it by the vertex y (if its color differs from γ).
Extend the subpath $L'_{\gamma+1}$ from $s_{2\gamma+3}$ by including all the vertices of Y from x to y that were not included into L'_γ and terminate the modified subpath in vertex y.
7: Merge the subpaths $L'_0, \ldots, L'_{\lceil k/2 \rceil - 1}$ into a cycle H.

Output The Hamiltonian cycle H.

Note that, in the HC-Algorithm, when the special vertices are connected by paths, the cycle is traversed always in one direction so that the special vertex s_1 is never bypassed. In our algorithm, instead, we connect an ordered triple of special vertices together in one traversal of the cycle. (If k is odd, the last path is connecting just two special vertices – s_k and s_1.) In such a traversal, we cannot always avoid the bypasses of the special vertex s_1 in the cycle.

The advantage of this approach is that connecting special vertices in triples is roughly halving the estimated cost of the solution: In our algorithm we always bound the cost of two paths connecting a pair of special vertices (except possible the last one) by the cost of the underlying structure (the cycle C). This approach of estimating the cost of more structures at once is new in this context and was

https://doi.org/10.1051/ita/2014021 Published online by Cambridge University Press

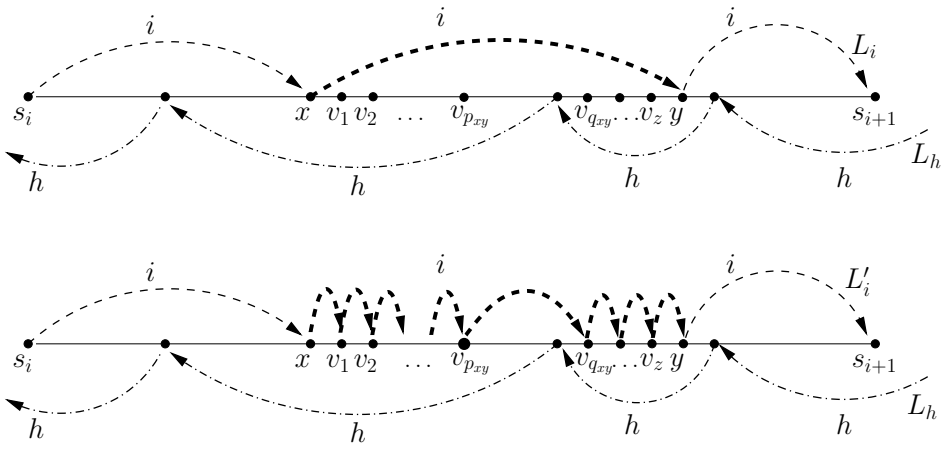


FIGURE 1. Modifying path L_i to path L'_i: All vertices which are bypassed by the edge xy of L_i and not yet included into other subpaths are connected to subpath L_i.

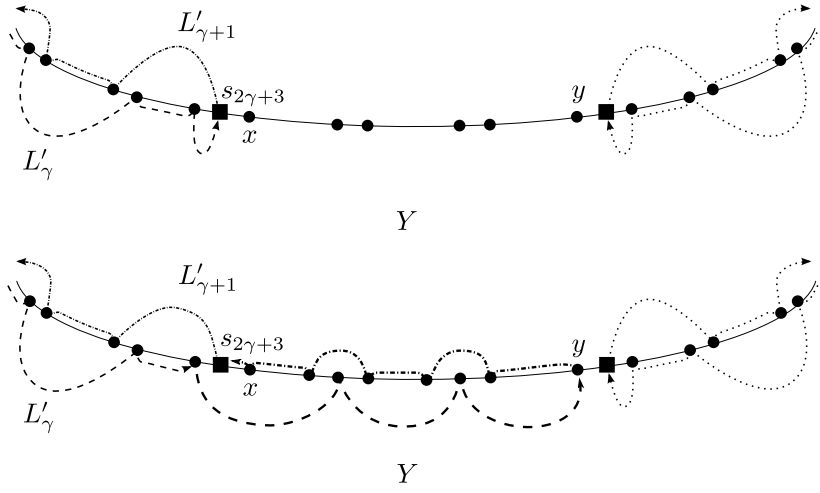


FIGURE 2. An extension of the paths L'_γ and $L'_{\gamma+1}$ by the path Y in Step 6 of Algorithm 1. Squares denote special vertices, dots denote non-special vertices. The terminal vertex $s_{2\gamma+3}$ is removed from L_γ, the subpath L_γ is extended by inclusion of all the vertices of color γ in Y and terminated by the vertex y. The subpath $L'_{\gamma+1}$, in which the special vertex $s_{2\gamma+3}$ is preserved, is extended by inclusion of all the remaining vertices of Y and is terminated by the vertex y.

https://doi.org/10.1051/ita/2014021 Published online by Cambridge University Press

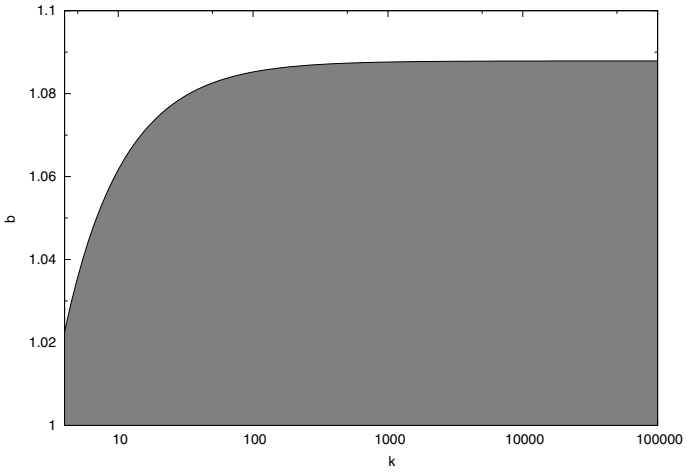


FIGURE 3. The graph shows the pairs (k, β) for which Algorithm 1 has provably better approximation ratio than the best known approximation algorithm.

not applied in any of the previous approximation algorithms for k-Δ_βOTSP. In all the previous approximation algorithms the cost of each path connecting a pair of special vertices was bounded by the cost of the entire underlying structure. Furthermore, the side effect of our approach is that our algorithm needs only $\lceil k/2 \rceil$ vertex-colors instead of $k+1$.

The drawback of this approach is that the bypasses might be slightly longer. More precisely, once we are passing the vertex s_1 in our traversal, we might have a *cut in the coloring*: the segment of the cycle between s_1 and the next vertex of the same color in the other direction than the one used for the coloring might miss vertices of some colors. This happens if the number of non-special vertices in the cycle is not divisible by the number of colors and thus the cycle does not contain the same number of vertices of each color. Here, the algorithm might be forced to bypass at most two vertices of the same color.

Lemma 3.1. *In Steps 3 and 4 of Algorithm 1, at most* $2\lceil k/2 \rceil + k - 3$ *edges of* C *are bypassed between two consecutive vertices of subpath* L_i*, for all* $0 \leq i \leq \lceil k/2 \rceil - 1$*. Furthermore, a bypass of at most* $2\lceil k/2 \rceil + k - 3$ *edges can occur only if the vertex* s_1 *is bypassed. All other bypasses jump over at most* $\lceil k/2 \rceil + k - 2$ *edges.*

Proof. Let x and y be two consecutive vertices of subpath L_i. Since we use a cyclic coloring of the non-special vertices of C, the only color cut can be around the vertex s_1.

Assume that vertex s_1 is bypassed by the edge xy in L_i. There are $\lceil k/2 \rceil - 1$ different colors except i. For each such color, at most two vertices can be present in between x and y in C. In addition, there are at most $k-2$ special vertices that

https://doi.org/10.1051/ita/2014021 Published online by Cambridge University Press

can be present here as well. Hence, the overall number of vertices between two consecutive vertices on each subpath that is bypassing the color cut is at most $2\lceil k/2\rceil + k - 4$.

Assume that vertex s_1 is not bypassed by the edge xy of L_i. Then, since the coloring is cyclic, there is exactly one non-special vertex of every other color between x and y. In addition, there are at most $k-2$ special vertices that can be present here as well. Together, there are at most $\lceil k/2\rceil + k - 3$ vertices that are bypassed by the edge xy, which proves the second part of the claim. □

The second observation is used in Step 5 of Algorithm 1 in the same way as it was used in Step 4 of Algorithm 3 in [9] – including bypassed vertices into any subpath that is bypassing them cannot increase the cost of the subpath. The bottom line why this is so, is that the inclusion of a bypassed vertex preserves the vertex order in the subpath and does not increase the number of vertices which are bypassed. The process of the inclusion of bypassed vertices into subpaths L_i in the construction of the subpaths L'_i is depicted in Figure 1 and is described in detail in [9]. From now on, we omit to write colorings and indexing modulo $\lceil k/2\rceil$, when it is clear from the context.

Lemma 3.2. *The cost of the paths L'_i constructed in Step 5 from paths L_i cannot increase,* i.e.*, for all i, $0 \le i \le \lceil k/2\rceil - 1$, $cost(L'_i) \le cost(L_i)$.*

At this point, if we sequentially merge the subpaths $L'_0, \ldots, L'_{\lceil k/2\rceil-1}$ by their endpoints, we obtain a cycle Q that contains each vertex at most once and that respects the given order on the special vertices. However, some vertices of C might not be present in Q as no subpath is bypassing them. This case is covered by the third observation – see Figure 2 for details. The vertices not included in Q must form a single path Y in C. The path Y (if it exists) is the segment of the cycle C through which none of the L'_Γ subpaths traversed. (Otherwise the bypassed vertices would be included in Step 5 of our algorithm and hence they would not be present in Y.) The vertices of Y are included into two consecutive subpaths as follows. Since Y, with the endpoints x and y, is a path of C, there must exist subpaths L'_γ and $L'_{\gamma+1}$ with an endpoint $s_{2\gamma+3}$ that is adjacent with x. We split Y into two subpaths and extend each of the subpaths L'_γ and $L'_{\gamma+1}$ by one of these split subpaths of Y. The outcome of these extensions is that y becomes the endpoint of both subpaths and each vertex of Y is included into exactly one of the extended subpaths. Notice that, in the extension of the subpath L'_γ, its special vertex $s_{2\gamma+3}$ is removed and then it is continued by inclusion of the vertices of the color γ into the subpath up to the vertex y (inclusively). Since Y was not covered by L'_γ, the vertices of color γ are not already included in any subpath and this operation is sound. The other extension can be evaluated in the same way. The subpath $L'_{\gamma+1}$ (which still contains the special vertex $s_{2\gamma+3}$) is extended by including all the remaining vertices of Y into the subpath and is terminated in y. By the same reasoning as above, this operation is also sound. For our convenience of presenting the statements comprehensively, we call the extended subpaths still L'_γ and $L'_{\gamma+1}$.

https://doi.org/10.1051/ita/2014021 Published online by Cambridge University Press

After the transformation of the subpaths L'_γ and $L'_{\gamma+1}$ in Step 6, observe that two consecutive vertices in the extended subpaths are bypassing at most $2\lceil k/2\rceil + k - 3$ edges in C. This is true as (1) the non-extended parts have this property as discussed above; (2) in the extended part of L'_γ, only the vertices of color γ are used and the distance between the last vertex of the color γ and y, which may be of a different color, can be at most $2\lceil k/2\rceil + k - 3$; (3) the extended part of $L'_{\gamma+1}$ from $s_{2\gamma+3}$ to y contains all the remaining vertices. In particular, this also includes the vertices of the color $\gamma + 1$. Two consecutive vertices of this color are at distance at most $2\lceil k/2\rceil + k - 3$ in the cycle C. Note that this is also true for the distance between the last vertex x of the extended subpath and the vertex $s_{2\gamma+3}$. The inclusion of the vertices of the other colors in between the vertices of color $\gamma + 1$ cannot increase the distance.

Lemma 3.3. *The distance between two consecutive vertices in subpaths L'_i (for $0 \leq i \leq \lceil k/2\rceil - 1$ and $i \notin \{\gamma, \gamma+1\}$) and in the extended subpaths L'_γ and $L'_{\gamma+1}$ is at most $2\lceil k/2\rceil + k - 3$ on the cycle C.*
Furthermore, a bypass of at most $2\lceil k/2\rceil + k - 3$ edges can occur only if the vertex s_1 is bypassed (i.e., *if the bypass occurs on the color cut). All other bypasses jump over at most $\lceil k/2\rceil + k - 2$ edges.*

Proof. The lemma is proven by the discussion above and a similar reasoning as we used in Lemma 3.1. □

From the description above, it is clear that the cycle H computed by our algorithm is a feasible solution, *i.e.*, each vertex of the graph is included in H once and the special vertices are present in the given order. Now we are ready to estimate the approximation ratio of our algorithm.

Theorem 3.4. *For $k \geq 4$, Algorithm 1 computes, in polynomial time, an approximate solution for k-Δ_βOTSP with a ratio of at most*

$$\left\lceil \frac{k}{2} \right\rceil \cdot \beta^{\lceil \log_2(2\lceil k/2\rceil + k - 3)\rceil} \cdot \min\left\{4\beta, \frac{3}{2}\beta^2, \beta^2 + \beta\right\}.$$

Proof. Observe that, in a subpath L'_i, no vertex is present multiple times. Due to the construction of the subpaths, we can always pick a path in C from which we can remove vertices in contiguous blocks of bounded length to obtain the subpath L'_i. (This is true also for the extended subpaths.) Therefore, by Lemma 3.3 we have

$$cost(L'_i) \leq \beta^{\lceil \log_2(2\lceil k/2\rceil + k - 3)\rceil} \cdot cost(C).$$

The cost of the merged subpaths forming the cycle H can be estimated as

$$cost(H) \leq \left\lceil \frac{k}{2} \right\rceil \cdot \beta^{\lceil \log_2(2\lceil k/2\rceil + k - 3)\rceil} \cdot cost(C).$$

https://doi.org/10.1051/ita/2014021 Published online by Cambridge University Press

In Step 1 of the algorithm, we use an approximation algorithm for Δ_β-TSP which influences the overall ratio. The currently best approximation ratio for Δ_β-TSP is

$$\min\left\{4\beta, \frac{3}{2}\beta^2, \beta^2+\beta\right\}$$

and can be achieved by the best out of the currently known approximation algorithms (Thm. 2.1). This gives us the claimed ratio. □

The following theorem characterizes the instances for which our algorithm improves over the $k\beta^{\lceil \log_2(\lfloor 3k/2\rfloor+1)\rceil}$-approximation algorithm of [10].

Theorem 3.5. *For any* $1 < \beta < (3/4)^{\frac{1}{\log_2 3/32}} < 1.08789$*, there exists a fixed* $k_0 \geq 4$ *such that, for all* $k \geq k_0$*, the approximation ratio of Algorithm 1 is better than the ratio achieved by Algorithm 2 of* [10] *for the pairs* (β, k)*. In these cases, the approximation ratio of Algorithm 1 is at most*

$$\left\lceil \frac{k}{2} \right\rceil \cdot \frac{3}{2} \cdot \beta^{2+\lceil \log_2(2\lceil k/2\rceil+k-3)\rceil}.$$

Proof. As the approximation algorithm for building the cycle C in Step 1 of our algorithm we employ the $\frac{3}{2}\beta^2$-approximation algorithm for Δ_β-TSP from [7]. To see for which pairs of (k, β) our algorithm is better, we compare the approximation ratios of the two algorithms.

To simplify the calculations, we bound the estimated approximation ratio of our algorithm from above.

$$\begin{aligned}\frac{3}{2} \cdot \left\lceil \frac{k}{2} \right\rceil \cdot \beta^{2+\lceil \log_2(2\lceil k/2\rceil+k-3)\rceil} &\leq \frac{3}{2} \cdot \frac{k+1}{2} \cdot \beta^{3+\log_2(k+1+k-3)} \\ &\leq \frac{3}{4} \cdot (k+1) \cdot \beta^{\log_2(16k-16)}.\end{aligned}$$

Similarly, we bound the estimated approximation ratio of the algorithm from [10] from below.

$$k \cdot \beta^{\log_2(\frac{3k+1}{2})} \leq k \cdot \beta^{\lceil \log_2(\lfloor 3k/2\rfloor+1)\rceil}.$$

Algorithm 1 has a better approximation ratio if inequality (3.1) with the estimated lower bound of the approximation ratio of the old k-Δ_βOTSP algorithms from [10] is satisfied:

$$\frac{3}{4} \cdot (k+1) \cdot \beta^{\log_2(16k-16)} \leq k \cdot \beta^{\log_2(\frac{3k+1}{2})}. \tag{3.1}$$

After applying a couple of basic operations, the inequality can be modified into its final form

$$\log_2 \beta \leq \frac{\log_2 \frac{3k+3}{4k}}{\log_2 \frac{3k+1}{32k-32}}. \tag{3.2}$$

https://doi.org/10.1051/ita/2014021 Published online by Cambridge University Press

The function on the right-hand side of inequality (3.2) for $k \geq 4$ is increasing. Hence the improvement is obtained for all β such that

$$\log_2 \beta < \lim_{k\to\infty} \frac{\log_2 \frac{3k+3}{4k}}{\log_2 \frac{3k+1}{32k-32}} = \log_{3/32} \frac{3}{4}.$$

This allows us to estimate β to be

$$\beta < \left(\frac{3}{4}\right)^{\frac{1}{\log_2 3/32}} < 1.08788973739.$$

We would like to note that, at this point, we do not have to consider the other two approximation algorithms which can be used for computing the cycle C. Note that we achieved an improvement only for values $\beta < 2$. In these cases, the $3/2\beta^2$-approximation algorithm of Böckenhauer *et al.* [7] is achieving the best approximation ratio out of all three approximation algorithms. Thus, the usage of the other two approximation algorithms for k-Δ_βOTSP in Step 1 of the algorithm cannot bring further improvements. □

Our calculations from the proof of Theorem 3.5 imply that, for arbitrary $1 < \beta < \left(\frac{3}{4}\right)^{\frac{1}{\log_2 3/32}}$, there exists an initial value $k_0 \geq 4$ such that, for all $k \geq k_0$, the approximation ratio of Algorithm 1 is better than the ratio of the previously best known algorithm. The pairs (k, β) for which we obtain an improvement are depicted in Figure 3.

4. Improving the coloring

We can identify three parts of Algorithm 1 which contribute to the overall approximation ratio $3/2 \cdot \lceil k/2 \rceil \cdot \beta^{2+\lceil \log_2(2\lceil k/2\rceil + k - 3)\rceil}$. The factor $3\beta^2/2$ is given by the algorithm computing the cycle C in Step 1. The factor $\lceil k/2 \rceil$ comes from structural properties of a cycle (*i.e.*, the number of times $cost(C)$ is used to estimate the connected special vertices is halved as triples of such vertices can be connected at once). We do not aim here for the improvement of these two factors. The last part produces the factor $\beta^{\lceil \log_2(2\lceil k/2\rceil + k - 3)\rceil}$. Its exponent corresponds to the maximal length of a bypass in the solution and is dependent on the used vertex-coloring procedure. In this section, we introduce another observation which allows us to decrease the length of bypasses and hence to improve the approximation ratio of our algorithm.

All our bypasses in Algorithm 1 that are not bypassing the color cut around s_1 contain at most $\lceil k/2 \rceil + k - 3$ vertices. Only the bypasses that occur around this cut are the bottleneck of our estimations and are increasing the estimated length of the bypasses from $\lceil k/2 \rceil + k - 2$ to $2\lceil k/2 \rceil + k - 3$ in all our calculations. For the ease of the presentation, we call a bypass that bypasses also vertex s_1 (and hence the color cut) a *color-cut bypass*.

https://doi.org/10.1051/ita/2014021 Published online by Cambridge University Press

We estimated that each color-cut bypass contains at most $2 \cdot (\lceil k/2 \rceil - 1) + x$ vertices, where x is the number of special vertices that are bypassed. Hence, the main idea of the improvement is to choose a vertex from which the coloring is started (and which also places the color cut into the cycle) such that each color-cut bypass contains a minimum number of special vertices.

Let d be a positive integer whose precise value is specified later. Let us divide the cycle C into d parts of roughly the same size. From the pigeon-hole principle, one of these parts contains at most $\lfloor k/d \rfloor$ special vertices and at least $\lfloor n/d \rfloor$ of all the vertices of C. If this part of the cycle contains at least $3 \cdot \lceil k/2 \rceil - 1$ non-special vertices, we can place the color cut into this part such that starting and ending vertices of all color-cut bypasses are present in this part. Then, the number of bypassed vertices in each color-cut bypass is at most $2 \cdot (\lceil k/2 \rceil - 1) + \lfloor k/d \rfloor$. Hence, if we choose d to be large (but still $d \leq n$), the number of vertices bypassed by each color-cut bypass is approximately $k + \frac{k}{n}$.

The last part that we have to show is that, indeed, the chosen part contains at least $3 \cdot \lceil k/2 \rceil - 1$ vertices with respect to d and n. We can estimate the number of non-special vertices in the chosen part of C which is at least

$$\left\lfloor \frac{n}{d} \right\rfloor - \left\lfloor \frac{k}{d} \right\rfloor \geq \frac{n-d+1}{d} - \frac{k}{d} = \frac{n-d+1-k}{d}.$$

Hence, if

$$\frac{n-d+1-k}{d} \geq 3 \cdot \left\lceil \frac{k}{2} \right\rceil - 1,$$

we can place the color cut into the chosen part. The inequality above is true if

$$\frac{n-d+1-k}{d} \geq 3 \cdot \frac{k+1}{2} - 1,$$

and hence,

$$2 \cdot (n-d+1-k) \geq 3d \cdot (k+1) - 2d.$$

If we further modify the inequality, we are able to bound d by

$$d \leq \frac{2(n+1-k)}{3k+3}. \tag{4.1}$$

Any positive integer value d satisfying the inequality (4.1) can be used to divide the cycle into d parts, one of these parts contains only $\lfloor k/d \rfloor$ special vertices which allows each color-cut bypass to contain at most $2 \cdot (\lceil k/2 \rceil - 1) + \lfloor k/d \rfloor$ vertices. To minimize the length of the color-cut bypass, we need to minimize $\lfloor k/d \rfloor$ and hence d has to be maximal.

Since our coloring achieves bypasses of $\lceil k/2 \rceil + k - 3$ vertices for each segment of the cycle without the color cut, the partition of the cycle C into many parts does not always help to decrease the length of bypasses.

https://doi.org/10.1051/ita/2014021 Published online by Cambridge University Press

If $d = 2$ (and $n \geq 4k+2$), the color-cut bypasses contain at most $2(\lceil k/2 \rceil - 1) + \lfloor k/2 \rfloor = \lceil k/2 \rceil + k - 2$ vertices which is by 1 vertex more than the number of vertices bypassed in the remainder of the cycle. Therefore, if $n \geq 4k+2$, Algorithm 1 enhanced by the special placement of the color cut has an approximation ratio of

$$\left\lceil \frac{k}{2} \right\rceil \cdot \beta^{\lceil \log_2(\lceil k/2 \rceil + k - 1) \rceil} \min \left\{ 4\beta, \frac{3}{2}\beta^2, \beta^2 + \beta \right\}.$$

If $k \geq 4$, then $\lfloor k/3 \rfloor \leq \lfloor k/2 \rfloor - 1$. Thus, if in addition $d = 3$ (and hence $2n \geq 11k+7$), the color-cut bypasses become at most as long as the non-color-cut bypasses and the enhanced Algorithm 1 is achieving ratio

$$\left\lceil \frac{k}{2} \right\rceil \cdot \beta^{\lceil \log_2(\lceil k/2 \rceil + k - 2) \rceil} \min \left\{ 4\beta, \frac{3}{2}\beta^2, \beta^2 + \beta \right\}.$$

For the latter approximation ratio with bypasses of length $\lceil k/2 \rceil + k - 2$ edges, we can do similar calculations as we did in Section 3 when the $\frac{3}{2}\beta^2$-approximation algorithm for Δ_β-TSP from Böckenhauer *et al.* [7] is plugged into Step 1 which yields the following theorem.

Theorem 4.1. *For any $1 < \beta < \sqrt[3]{\frac{4}{3}}$, there exists a fixed $k_0 \geq 4$ such that, for all $k \geq k_0$ such that $2n \geq 11k + 7$, the approximation ratio of Algorithm 1 with improved coloring is better than the ratio achieved by Algorithm 2 of [10] for the pairs (β, k). In these cases, the approximation ratio of Algorithm 1 is at most*

$$\left\lceil \frac{k}{2} \right\rceil \cdot \frac{3}{2} \cdot \beta^{2 + \lceil \log_2(\lceil k/2 \rceil + k - 2) \rceil}.$$

Proof. The inequality (3.1) is changed to

$$\frac{3}{4} \cdot (k+1) \cdot \beta^{\log_2(12k-12)} \leq k \cdot \beta^{\log_2(\frac{3k+1}{2})},$$

and the inequality (3.2) changes to

$$\log_2 \beta \leq \frac{\log_2 \frac{3k+3}{4k}}{\log_2 \frac{3k+1}{24k-24}}.$$

Again, the function on the right-hand side is increasing for $k \geq 4$ and, hence, an improvement is obtained for all β (and all values of k from some point) such that

$$\log_2 \beta < \lim_{k \to \infty} \frac{\log_2 \frac{3k+3}{4k}}{\log_2 \frac{3k+1}{24k-24}} = \log_{3/24} \frac{3}{4}.$$

This allows us to estimate β to satisfy

$$\beta < \sqrt[3]{\frac{4}{3}}.$$

Again, as we achieved an improvement only for values $\beta < 2$, there is no need to apply other two approximation algorithms for Δ_β-TSP in Step 1. □

https://doi.org/10.1051/ita/2014021 Published online by Cambridge University Press

5. Conclusion

In this paper, we investigated approximation algorithms for the Ordered Near-Metric Traveling Salesman Problem (k-Δ_βOTSP). The first, general approximation algorithm improves the best known approximation ratio for instances with $\beta < 1.087889$ and the last ratio is an improvement for $\beta < \sqrt[3]{4/3}$. In both cases, this is true for all k except for the first finitely many small values, but, in the latter case, k should be at most proportional to n, more precisely $2n \geq 11k + 7$.

Even though the obtained improvement of our algorithm does not cover the entire space of instances, it broadens our knowledge on the approximability of the most interesting instances – those that are close to being metric.

There are still several open questions related to k-Δ_βOTSP and k-OTSP. First of all, we are not aware of any hard instances for approximation algorithms which could suggest whether the estimated ratios are tight or there is still space for improvements. Furthermore, our algorithm, same as the algorithms of [8,9], is using a "static" coloring of vertices that does not reflect the actual situation in the initial Hamiltonian cycle. We believe that the approximation ratio, especially the part that is dependent on the length of bypasses, could be improved by a more sophisticated coloring.

Our enhancement from Section 4 for n-vertex graphs can be applied only if $n \geq 4k + 2$ (or, to be a little bit better, if $2n \geq 11k + 7$), *i.e.*, if the number of special vertices is not too high. However, it seems that the cases where n and k are of the same order of magnitude are the hardest ones. For these cases, one may consider an approximation algorithm that is different in its used structures from all our approaches. We can even generalize the idea and consider the problem on instances where the number of special vertices k is a constant fraction of n.

References

[1] T. Andreae, On the traveling salesman problem restricted to inputs satisfying a relaxed triangle inequality. *Networks* **38** (2001) 59–67.

[2] T. Andreae and H.-J. Bandelt, Performance guarantees for approximation algorithms depending on parametrized triangle inequalities. *SIAM J. Discrete Math.* **8** (1995) 1–16.

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

[4] H.-J. Bandelt, Y. Crama and F.C.R. Spieksma, Approximation algorithms for multidimensional assignment problems with decomposable costs. *Discrete Appl. Math.* **49** (1994) 25–50.

[5] M. Bender and C. Chekuri, Performance guarantees for TSP with a parametrized triangle inequality. *Inf. Proc. Lett.* **73** (2000) 17–21.

[6] H.-J. Böckenhauer and M. Steinová, Improved approximations for ordered TSP on near-metric graphs. In *Proc. of the 39th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2013)*. Vol. 7741 of *Lect. Notes Comput. Sci.* Springer (2013) 157–168.

[7] H.-J. Böckenhauer, J. Hromkovič, R. Klasing, S. Seibert and W. Unger, Towards the notion of stability of approximation for hard optimization tasks and the traveling salesman problem. *Theoret. Comput. Sci.* **285** (2002) 3–24.

https://doi.org/10.1051/ita/2014021 Published online by Cambridge University Press

[8] H.-J. Böckenhauer, J. Hromkovič, J. Kneis and J. Kupke, On the approximation hardness of some generalizations of TSP (extended abstract), in *Proc. of the 10th Scandinavian Workshop on Algorithm Theory (SWAT 2006)*. Vol. 4059 of *Lect. Notes Comput. Sci.* Springer (2006) 184–195.
[9] H.-J. Böckenhauer, R. Klasing, T. Mömke and M. Steinová, Improved approximations for TSP with simple precedence constraints. In *Proc. of the 7th International Conference on Algorithms and Complexity (CIAC 2010)*. Vol. 6078 of *Lect. Notes Comput. Sci.* Springer (2010) 61–72.
[10] H.-J. Böckenhauer, T. Mömke and M. Steinová, Improved approximations for TSP with simple precedence constraints. *J. Discrete Algorithms* **21** (2013) 32–40.
[11] 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 (1976).
[12] R. Diestel, Graph Theory. Springer, 3rd edition (2005).
[13] T.F. Gonzalez, Handbook of Approximation Algorithms and Metaheuristics. *Chapman & Hall/CRC Computer and Information Science Series*. Chapman & Hall/CRC (2007).
[14] G. Gutin and A.P. Punnen, The Traveling Salesman Problem and Its Variations. Combinatorial Optimization. Springer, New York (2007).
[15] J. Hromkovič, Algorithmics for Hard Problems. Introduction to Combinatorial Optimization, Randomization, Approximation, and Heuristics. Springer (2003).
[16] D.B. West, Introduction to Graph Theory. Prentice Hall Inc., Upper Saddle River, NJ, 2nd edition (2000).

Communicated by J. Hromkovic.
Received August 6, 2013. Accepted March 31, 2014.

https://doi.org/10.1051/ita/2014021 Published online by Cambridge University Press