Approximation Algorithms for the Traveling Salesman Problem

DISSERTATION

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Xianghui Zhong

aus

Fujian

Bonn 2020

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Stefan Hougardy
2. Gutachter: Prof. Dr. Heiko Röglin
Tag der Promotion: 16.03.2021
Erscheinungsjahr: 2021

Abstract

The traveling salesman problem (TSP) is probably one of the best-studied problems in discrete optimization. Given a complete weighted graph with n vertices, the task is to find a tour of minimal length that visits every vertex exactly once. In this thesis we will focus on two questions regarding the traveling salesman problem: Finding the approximation ratio of the k-Opt and Lin-Kernighan algorithm and the integrality ratio of the subtour LP.

The k-Opt and Lin-Kernighan algorithm are two of the most important local search approaches for the Metric TSP. Both start with an arbitrary tour and make local improvements in each step to get a shorter tour. In the first part of the thesis we determine the exact approximation ratio $\sqrt{\frac{n}{2}}$ for the 2-Opt algorithm. Then we show that for any fixed $k \geq 3$ the approximation ratio of the k-Opt algorithm for Metric TSP is $O(\sqrt[k]{n})$. Assuming the Erdős girth conjecture, we prove a matching lower bound of $\Omega(\sqrt[k]{n})$. Unconditionally, we obtain matching bounds for $k = 3, 4, 6$ and a lower bound of $\Omega(n^{\frac{2}{3k-3}})$. Our most general bounds depend on the values of a function from extremal graph theory and are tight up to a factor logarithmic in the number of vertices unconditionally. Moreover, all the upper bounds also apply to a parameterized version of the Lin-Kernighan algorithm with appropriate parameters. Furthermore, we show that the approximation ratio of k-Opt for Graph TSP is between $\Omega\left(\frac{\log(n)}{\log\log(n)}\right)$ and $O\left(\left(\frac{\log(n)}{\log\log(n)}\right)^{\log_2(9)+\epsilon}\right)$ for all $\epsilon > 0$. If the vertices of the instance can be embedded into $\mathbb{R}^d$ such that the distances arise from the p-norm, we show that the approximation ratio is $\Omega\left(\frac{\log(n)}{\log\log(n)}\right)$. For the (1,2)-TSP we prove that the exact approximation ratio of the 3-Opt algorithm is $\frac{11}{8}$. We introduce a modified version of the k-Opt algorithm for the (1,2)-TSP and show that it has for $k = 3$ an exact approximation ratio of $\frac{4}{3}$. The k-improv algorithm is the currently best approximation algorithm with respect to approximation ratio for the (1,2)-TSP. We give a lower bound of $\frac{11}{10}$ for the k-improv algorithm for arbitrarily fixed k. This lower bound also carries over to the k-Opt algorithm for the (1,2)-TSP.

Another useful tool to approximate the TSP is the subtour LP. Many approximation algorithms use the optimal solution of the well-known LP relaxation. Although the exact integrality ratio of the subtour LP is still unknown, it is conjectured to be $\frac{4}{3}$. In the second part of the thesis we compute the exact integrality ratio for Rectilinear TSP with up to 10 vertices. Based on the computation results we give lower bounds depending on the number of vertices for several TSP variants and show that some of them are tight under certain assumptions. We also investigate the concept of local optimality with respect to integrality ratio and develop several algorithms to find instances with high integrality ratio for Euclidean TSP. Moreover, we improve the upper bound on the integrality ratio for $s-t$ Path TSP to 1.5273.

Acknowledgements

I would like to express my gratitude towards everyone who supported me in the past few years. First and foremost I want to thank both of my advisors Stefan Hougardy and Heiko Röglin. Whenever I had a problem they were ready to help and give me valuable advice. Through their joint effort and commitment I was granted funding for my research. Additionally I would like to acknowledge the support they gave me throughout my Bachelor and Master studies.

Also it was a pleasure to collaborate with Fabian Zaiser. Moreover, I would like to thank Fabian Henneke and Yvonne Omlor for reading parts of the thesis. Last but not least I would like to thank my colleagues with whom I had many discussions on mathematics and other topics. They made my time as Ph.D. student a lot more enjoyable.

I was financially supported by the Bonn International Graduate School.

Contents

1 Introduction **7**
1.1 Previous Work 8
1.1.1 Approximation Ratio of the k-Opt and Lin-Kernighan Algorithm . 8
1.1.2 Integrality Ratio of the Subtour LP 9
1.2 New Results 10
1.2.1 Approximation Ratio of the k-Opt and Lin-Kernighan Algorithm . 10
1.2.2 Integrality Ratio of the Subtour LP 12
1.3 Structure of the Thesis 14
1.4 Prelimilaries 15
1.4.1 Basic Definitions and Properties 15
1.4.2 Traveling Salesman Problem 17
1.4.3 k-Opt Algorithm 17
1.4.4 k-Improv Algorithm 18
1.4.5 Lin-Kernighan Algorithm 19
1.4.6 Girth and Ex 21
1.4.7 Linear Programming 21
1.4.8 Subtour LP and Integrality Ratio 22
1.4.9 Structure of Euclidean Tours 23
1.4.10 Karamata's inequality 23

2 Approximation Ratio of the k-Opt and Lin-Kernighan Algorithm **24**
2.1 Exact Approximation Ratio of 2-Opt for METRIC TSP 24
2.1.1 Upper Bound 24
2.1.2 Lower Bound 26
2.2 Approximation Ratio of k-Opt and Lin-Kernighan for METRIC TSP . . . 28
2.2.1 Lower Bound 28
2.2.2 Outline of Upper Bound 30
2.2.3 Upper Bound 32
2.2.4 Comparing the Lower and Upper Bound 41
2.3 Approximation Ratio of k-Opt for GRAPH TSP 43
2.3.1 Lower Bound 43
2.3.2 Outline of Upper Bound 45
2.3.3 Upper Bound 47
2.4 Approximation Ratio of k-Opt for EUCLIDEAN TSP 51
2.5 Approximation Ratio of k-Opt for (1,2)-TSP 57
2.5.1 Lower Bound on the Approximation Ratio of the 2-Opt Algorithm 57
2.5.2 Approximation Ratio of the 3-Opt Algorithm 59
2.5.3 Approximation Ratio of the 3-Opt++ Algorithm 66
2.5.4 Lower Bound on the Approximation Ratio of the k-Improv and k-Opt Algorithm 69
2.6 A Polynomial Time Local Search Algorithm for GRAPH TSP 73

3 Integrality Ratio of the Subtour LP **76**
3.1 Construction of Instances with Integrality Ratio Converging to $\frac{4}{3}$ 76
3.1.1 Construction 76
3.1.2 Applications 78
3.2 Computing the Exact Integrality Ratio for Rectilinear TSP 79
3.3 Integrality Ratio for Rectilinear TSP 80
3.3.1 Structure of the Fractional Tours 81
3.3.2 Structure of the Optimal Tours 81
3.3.3 The Instance $I^2_{i,j,k}$ 83
3.3.4 Length of the Optimal Tours for $I^2_{i,j,k}$ 85
3.3.5 The Integrality Ratio of $I^2_{i,j,k}$ 86
3.4 Integrality Ratio for Metric TSP 87
3.5 Integrality Ratio for Multidimensional Rectilinear TSP 90
3.6 Local Optimality 92
3.6.1 A Criterion for Local Optimality 92
3.6.2 Local Optimality for the p-Norm 95
3.6.3 A Local Search Algorithm 95
3.7 Integrality Ratio for Euclidean TSP 96
3.7.1 The Ellipse Construction Algorithm 97
3.7.2 Results of the Ellipse Construction Algorithm 101
3.8 Comparing Integrality Ratio 102
3.9 Hard to Solve Instances 103
3.10 Integrality Ratio for Metric $s-t$ Path TSP 105

4 Conclusion and Open Problems **109**
4.1 Approximation Ratio of the k-Opt and Lin-Kernighan Algorithm 109
4.2 Integrality Ratio of the Subtour LP 110

Bibliography **110**

1 Introduction

The traveling salesman problem (TSP) is probably the most well-known problem in discrete optimization. The problem can be formulated as follows: A salesman in his home town wants to make a *tour* visiting a given set of cities and return to his home town in the end. The question is in which order he should visit them to minimize the travel distance. The problem has various applications in practice, for example in chip design or logistics. This thesis addresses two questions about the traveling salesman problem.

In the most general form the distances between the cities could be arbitrary non-negative numbers. In practice however certain restrictions to the distance function often apply. For example the triangle inequality for the distances: If we travel from city A directly to city C, it cannot be farther than traveling via city B. The TSP variant where the distances satisfy the triangle inequality is called METRIC TSP. Another special case is when the cities lie in a Euclidean plane and the distance function arises from the Euclidean distance. This variant is called the EUCLIDEAN TSP.

Unfortunately, it is quite hard to find an optimal solution efficiently in general. The TSP and even the variants above are known to be NP-hard [30, 42, 52, 29]. That means an efficient algorithm solving this problem would imply an efficient algorithm on a various number of problems. It is not expected that such an efficient algorithm exists.

In order to speed up the calculation of a good tour in practice, several approximation algorithms are considered. Instead of computing the optimal solution they calculate a solution which is at most by some factor larger than the optimal solution. This factor is also called the *approximation ratio* of the algorithm. The approximation ratio is one way to compare approximation algorithms. The previous best approximation algorithm in terms of approximation ratio for METRIC TSP was independently developed by Christofides and Serdjukov with an approximation ratio of $\frac{3}{2}$ [18, 59]. Despite huge efforts this factor could not be improved in 40 years. Recently, the approximation ratio was improved by Karlin, Klein and Oveis Gharan to $\frac{3}{2} - \epsilon$ for some $\epsilon > 10^{-36}$ [41].

However, the approximation ratio only describes the worst-case behavior but not the average case behavior or the behavior on real world instances. In fact, in practice other algorithms are usually easier to implement and have better performance and runtime [8, 39, 55]. One natural approach is the k-Opt algorithm which is based on local search. It starts with an arbitrary tour and replaces at most k edges by new edges such that the resulting tour is shorter. It stops if the procedure cannot be applied anymore. The behavior of this algorithm was not well understood, not even in the worst case.

One of the best practical heuristics by Lin and Kernighan is based on k-Opt [49]. The Lin-Kernighan algorithm, like the k-Opt algorithm, modifies the tour locally to obtain a new tour. Instead of replacing arbitrary k edges with new edges, which results in a high runtime for large k, it searches for specific changes: Changes, where the edges to be added and deleted are alternating in a closed walk, a so-called closed alternating walk. No non-trivial approximation guarantee on this algorithm was known.

Another approach to approximate the TSP is using the so-called *subtour linear pro-*

gram (subtour LP): We introduce for every connection between two cities a variable whose value indicates if the tour uses this connection or not. If the value is 1, the connection is in the tour, otherwise the value is 0 and the connection is not in the tour. Certain additional conditions ensure that a valid assignment of the values corresponds to a tour. If we omit the condition that the value of the variables has to be either 0 or 1 and instead just require it to be between 0 and 1, we get the subtour LP. It turns out that the subtour LP can be solved in polynomial time. However, the optimal solution of the subtour LP which we call the optimal *fractional tour* is not necessarily a tour. It has more freedom to choose the values of the variables and could be cheaper than the optimal tour. Thus, it is important to find the maximal ratio of the length of the optimal tour to that of the optimal fractional tour. This ratio is also called the *integrality ratio* of the subtour LP. Its exact value is still unknown, only lower and upper bounds of $\frac{4}{3}$ [67] and $\frac{3}{2}$ [68] exist, respectively.

The first part of the thesis will focus on the approximation ratio of the k-Opt and the Lin-Kernighan algorithm on various TSP variants. We give new and improved bounds on the approximation ratio of the k-Opt algorithm for various TSP variants. This gives us a good insight into the behavior of the approximation ratio of the k-Opt algorithm.

The second part of the thesis deals with the integrality ratio of the subtour LP. We will mainly focus on the lower bounds of the integrality ratio. We compute the exact integrality ratio in the rectilinear case with a small fixed number of vertices. Based on these results we construct families of instances with a high integrality ratio for various TSP variants. We show that under certain assumptions some of the instances we describe maximize the integrality ratio among all instances with the same number of vertices in the particular TSP variant. We investigate several properties of these instances and develop algorithms that find instances with high integrality ratio. Moreover, we improve the upper bound on the integrality ratio for the $s-t$ Path TSP.

This thesis is partially based on work that is published in [36, 70, 71, 72].

1.1 Previous Work

1.1.1 Approximation Ratio of the k-Opt and Lin-Kernighan Algorithm

For the 2-Opt algorithm in the metric case Plesník showed that there are infinitely many instances with approximation ratio $\sqrt{\frac{n}{8}}$, where n is the number of vertices [53]. Chandra, Karloff and Tovey showed that the approximation ratio of 2-Opt is at most $4\sqrt{n}$ [17]. Levin and Yovel observed that the same proof yields an upper bound of $\sqrt{8n}$ [48].

For general $k > 2$ Chandra, Karloff and Tovey gave a lower bound of $\frac{1}{4}\sqrt[2k]{n}$ for Metric TSP [17], no non-trivial upper bound is known so far.

Apart from the upper bounds for the Metric TSP, which also apply to the special case of Graph TSP, only a lower bound of $2(1-\frac{1}{n})$ on the approximation ratio of the k-Opt algorithm for Graph TSP is known so far: Rosenkrantz, Stearns and Lewis describe a Metric TSP instance with this ratio that is also a Graph TSP instance [56].

For the Euclidean TSP Chandra, Karloff and Tovey showed that the approximation ratio of the 2-Opt algorithm is asymptotically between $\Omega\left(\frac{\log n}{\log\log(n)}\right)$ and $O(\log(n))$. The upper bound was improved by Brodowsky and Hougardy to $O\left(\frac{\log(n)}{\log\log(n)}\right)$ which imply a tight asymptotic approximation ratio of $\Theta\left(\frac{\log(n)}{\log\log(n)}\right)$ for the 2-Opt algorithm [15].

The currently best approximation ratio for the (1,2)-TSP is achieved by the k-improv algorithm by Berman and Karpinski with an approximation ratio of $\frac{8}{7}$ [9]. The k-improv algorithm is an improved version of the k-Opt algorithm that is based on a local search approach. Adamaszek, Mnich and Paluch proposed another algorithm with approximation ratio $\frac{8}{7}$ [1].

For the (1,2)-TSP it is known that the approximation ratio of the 2-Opt algorithm is at most $\frac{3}{2}$ [43]. It was noted in the same paper that this ratio can be proven to be tight. However, to our best knowledge, no explicit construction and proof for a lower bound was given so far.

Since the Lin-Kernighan algorithm uses a superset of the modification rules of the 2-Opt algorithm, the same upper bounds as for 2-Opt also apply. Apart from this, no other upper bound was known.

Beyond the worst-case analysis there are also results about the average case behavior of the algorithm. For example the smoothed analysis of the 2-Opt algorithm by Englert, Röglin and Vöcking [21]. In their model each vertex of the TSP instance is a random variable distributed in the d dimensional unit cube by a given probability density function $f_i : [0,1]^d \to [0,\phi]$ bounded from above by a constant $1 \leq \phi < \infty$ and the distances are given by the p-norm. They show that in this case the expected approximation ratio is bounded by $O(\sqrt[d]{\phi})$ for all p. In the model where any instance is given in $[0,1]^d$ and perturbed by Gaussian noise with standard deviation σ the approximation ratio was improved to $O(\log(\frac{1}{\sigma}))$ by Künnemann and Manthey [46].

1.1.2 Integrality Ratio of the Subtour LP

The exact integrality ratio of the subtour LP is still unknown. For the Metric TSP the currently best lower and upper bounds are $\frac{4}{3}$ [67] and $\frac{3}{2}$ [68], respectively. For the Euclidean case the same upper bound applies and Hougardy gave a lower bound of $\frac{4}{3}$ [35]. The belief is that the exact integrality ratio is $\frac{4}{3}$, this is also known as the $\frac{4}{3}$-*Conjecture*. The conjecture was proven for instances whose optimal fractional solutions satisfy certain properties [13].

Benoit and Boyd computed the exact integrality ratio for Metric TSP instances with a small fixed number of vertices [6]. For $6 \leq n \leq 10$ vertices they determined these instances with computer assistance and discovered that the instances achieving the maximum integrality ratio are unique and have certain structures. Later, Boyd and Elliott-Magwood could further decrease the computation time significantly by exploiting more structure of the subtour polytope. The computational results could be extended to $n = 12$ [14].

Hougardy and Zhong introduced a family of Euclidean TSP instances called the *tetrahedron instances* that has a different structure as the instances from [6] with integrality ratio converging to $\frac{4}{3}$ [38]. They also investigate the runtime of the currently fastest TSP solver `Concorde` [4] to solve the tetrahedron instances in practice. More precisely, they compare the runtime of the tetrahedron instances with the family of instances proposed in [35] and instances from the TSPLIB, a library of TSP instances [54]. It turned out that the tetrahedron instances are significantly harder to solve than the other instances in practice: `Corcorde` needs up to 1,000,000 more time to solve the instances compared to TSPLIB instances of similar size.

Recent research shows that the Metric $s-t$ Path TSP can be approximated within a factor of $\frac{3}{2}+\epsilon$ and $\frac{3}{2}$ [61, 69]. Moreover, it was shown that any α-approximation algorithm for the standard TSP problem implies an $(\alpha + \epsilon)$-approximation algorithm in the $s-t$ PATH TSP version [63]. The best currently known lower bound for the integrality ratio of the subtour LP for the METRIC $s-t$ PATH TSP is $\frac{3}{2}$. This value is achieved by a simple standard example. A recent series of work improves the upper bound towards the conjectured optimal value of $\frac{3}{2}$.

Hoogeveen adapted Christofides' algorithm for the standard TSP [18] (which was independently developed by Serdjukov [59]) to the $s-t$ PATH TSP [34]. A parity correction vector is added to a minimum spanning tree to obtain a tour. This leads to an integrality ratio of $\frac{5}{3}$ for the path version. An, Kleinberg and Shmoys suggested the best-of-many Christofides' algorithm for $s-t$ PATH TSP [3]. Instead of using the minimum spanning tree they decompose the optimal LP solution into a convex combination of spanning trees. Then, they sample the spanning tree according to the convex combination, add a parity correction vector and output the best result. With this approach the upper bound on the integrality ratio was improved to $\frac{1+\sqrt{5}}{2}$. Sebő improved and simplified this approach to obtain a ratio of $\frac{8}{5}$ [57]. In [64] Vygen chose the convex combination in a particular way. This idea was further improved by Gottschalk and Vygen by a generalization of the Gao trees [32]. For an upper bound of $\frac{3}{2} + \frac{1}{34}$, Sebő and Van Zuylen delete the so-called lonely edges of the spanning trees before adding the parity correction vector based on the underlying idea that the parity correction vector will likely reconnect the tour. If this is not the case, they add two copies of lonely edges to reconnect the tour afterwards [58]. The analysis was improved by Traub and Vygen by choosing the weights of the spanning trees in a non-standard way. This improves the ratio to $1 + \frac{1}{1+4\ln(\frac{5}{4})}$ [62].

1.2 New Results

1.2.1 Approximation Ratio of the k-Opt and Lin-Kernighan Algorithm

For the 2-Opt algorithm we determine in a joint work with Stefan Hougardy and Fabian Zaiser the exact approximation ratio:

Theorem 1.2.1. *The 2-Opt algorithm for* METRIC TSP *instances with n vertices has approximation ratio $\sqrt{\frac{n}{2}}$ and this result is tight.*

For fixed $k \geq 3$, we show that the approximation ratio of the k-Opt algorithm is related to the extremal graph theoretic problem of maximizing the number of edges in a graph with fixed number of vertices and no short cycles. Let $\mathrm{ex}(n, 2k)$ be the largest number of edges in a graph with n vertices and girth at least $2k$, i.e. it contains no cycles with less than $2k$ edges. For instances with n vertices we show for METRIC TSP that:

Theorem 1.2.2. *For all fixed k if $\mathrm{ex}(n, 2k) \in O(n^c)$ for some $c > 1$, the approximation ratio of k-Opt for* METRIC TSP *is $O(n^{1-\frac{1}{c}})$ where n is the number of vertices.*

Theorem 1.2.3. *For all fixed k if $\mathrm{ex}(n, 2k) \in \Omega(n^c)$ for some $c > 1$, the approximation ratio of k-Opt for* METRIC TSP *is $\Omega(n^{1-\frac{1}{c}})$ where n is the number of vertices.*

Using known upper bounds on $\mathrm{ex}(n, 2k)$ in [2] we can conclude:

Corollary 1.2.4. *The approximation ratio of k-Opt for* METRIC TSP *is in $O(\sqrt[k]{n})$ for all fixed k where n is the number of vertices.*

If we further assume the Erdős girth conjecture [24], i.e. $\mathrm{ex}(n, 2k) \in \Theta(n^{1+\frac{1}{k-1}})$, we have:

Corollary 1.2.5. *Assuming the Erdős girth conjecture, the approximation ratio of k-Opt for* Metric TSP *is in $\Omega(\sqrt[k]{n})$ for all fixed k where n is the number of vertices.*

Using known lower bounds on $\mathrm{ex}(n, 2k)$ from [22, 23, 16, 7, 60, 65, 47] we obtain:

Corollary 1.2.6. *The approximation ratio of k-Opt for* Metric TSP *is in $\Omega(\sqrt[k]{n})$ for $k = 3, 4, 6$ and in $\Omega(n^{\frac{2}{3k-4+\epsilon}})$ for all fixed k where $\epsilon = 0$ if k is even and $\epsilon = 1$ if k is odd and n is the number of vertices.*

Comparing our upper and lower bounds we obtain:

Theorem 1.2.7. *Our most general upper bound of*

$$\sum_{l=0}^{l^*} \frac{4\,\mathrm{ex}(4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil, 2k)}{(\frac{4k-4}{4k-5})^l}$$

on the approximation ratio of the k-Opt algorithm for Metric TSP *is tight up to a factor of $O(\log(n))$ where n is the number of vertices and $l^* := \min\{j \in \mathbb{N} | \sum_{l=0}^{j} 4\,\mathrm{ex}(4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil, 2k) \geq n\}$.*

The upper bounds can be carried over to a parameterized version of the Lin-Kernighan algorithm we will describe in detail later. In contrast to the original version of the algorithm proposed by Lin and Kernighan two parameters determine the depth that the algorithm searches for improvement.

Theorem 1.2.8. *The same upper bounds from Theorem 1.2.2 of $O(n^{1-\frac{1}{c}})$ if $\mathrm{ex}(n, 2k) \in O(n^c)$ and Theorem 1.2.4 of $O(\sqrt[k]{n})$ hold for a parameterized version of the Lin-Kernighan algorithm with appropriate parameters.*

Although the Lin-Kernighan algorithm only considers special changes, namely changes by augmenting a closed alternating walk, we are able to show the same upper bound as for the general k-Opt algorithm. For the original version of Lin-Kernighan we get an improved upper bound of $O(\sqrt[3]{n})$. Our results solve two of the four open questions in [17], namely:

- Can the upper bounds given in [17] be generalized to the k-Opt algorithm, i.e. for increasing k the performance guarantee improves?
- Can we show better upper bounds for the Lin-Kernighan algorithm than the upper bound obtained from the 2-Opt algorithm?

We also bound the approximation ratio of the k-Opt algorithm for Graph TSP.

Theorem 1.2.9. *The approximation ratio of the k-Opt algorithm with fixed $k \geq 2$ for* Graph TSP *is $\Omega\left(\frac{\log(n)}{\log\log(n)}\right)$ where n is the number of vertices.*

Theorem 1.2.10. *The approximation ratio of the 2-Opt algorithm for* Graph TSP *is $O\left(\left(\frac{\log(n)}{\log\log(n)}\right)^{\log_2(9)+\epsilon}\right)$ for all $\epsilon > 0$ where n is the number of vertices.*

<table>
<tr><th></th><th>2-Opt</th><th>k-Opt</th></tr>
<tr><td>METRIC TSP</td><td>$\sqrt{\frac{n}{2}}$</td><td>$\Omega(n^{\frac{2}{3k-3}}) \cap O(\sqrt[k]{n})$</td></tr>
<tr><td>GRAPH TSP</td><td colspan="2">$\Omega\left(\frac{\log(n)}{\log\log(n)}\right) \cap O\left(\left(\frac{\log(n)}{\log\log(n)}\right)^{3.17}\right)$</td></tr>
<tr><td>EUCLIDEAN TSP</td><td>$\Theta\left(\frac{\log(n)}{\log\log(n)}\right)$ [1,2]</td><td>$\Omega\left(\frac{\log(n)}{\log\log(n)}\right) \cap O\left(\frac{\log(n)}{\log\log(n)}\right)$ [2]</td></tr>
<tr><td>(1,2)-TSP</td><td>$\frac{3}{2}$ [3]</td><td>between $\frac{11}{10}$ and $\frac{11}{8}$</td></tr>
</table>

Table 1.1: The currently best bounds on the approximation ratio of the 2-Opt and k-Opt algorithm for $k > 2$. The red results are from this thesis.

Note that the same upper bound also applies to the k-Opt algorithm and the Lin-Kernighan algorithm since they produce 2-optimal tours. Hence, up to a constant factor of at most $\log_2(9)$ in the exponent the k-Opt algorithm does not achieve asymptotically better performance than the 2-Opt algorithm in contrast to the metric case.

If the vertices of the instance can be embedded into $\mathbb{R}^d$ such that the distance function arises from the p-norm, we show:

Theorem 1.2.11. *The approximation ratio of the k-Opt Algorithm for the* RECTILINEAR *and* EUCLIDEAN TSP *is* $\Omega\left(\frac{\log n}{\log\log n}\right)$ *where n is the number of vertices.*

For the (1,2)-TSP we give a lower bound of $\frac{3}{2}$. A matching upper bound of $\frac{3}{2}$ was given in [43] and it was noted that this bound can be shown to be tight. Nevertheless, no lower bound was given explicitly. Moreover, we show that the exact approximation ratio of the 3-Opt algorithm is $\frac{11}{8}$ for the (1,2)-TSP.

Theorem 1.2.12. *The exact approximation ratio of the 3-Opt algorithm for* (1,2)-TSP *is* $\frac{11}{8}$.

We introduce the k-Opt++ algorithm, a slightly modified version of the k-Opt algorithm for (1,2)-TSP, and analyze the exact approximation ratio of the 3-Opt++ algorithm.

Theorem 1.2.13. *The exact approximation ratio of the 3-Opt++ algorithm for* (1,2)-TSP *is* $\frac{4}{3}$.

Furthermore, we show a lower bound on the approximation ratio for the k-Opt and k-improv algorithm.

Theorem 1.2.14. *The k-Opt and k-improv algorithm with arbitrary fixed k have an approximation ratio of at least* $\frac{11}{10}$ *for the* (1,2)-TSP.

Moreover, we give a polynomial time local search algorithm for GRAPH TSP with constant approximation factor.

1.2.2 Integrality Ratio of the Subtour LP

We describe a procedure to construct families of EUCLIDEAN TSP instances whose integrality ratios converge to $\frac{4}{3}$. These instances can have a different structure than the known instances from the literature.

We use the same approach as Benoit and Boyd to compute the exact integrality ratio for Rectilinear TSP with $6 \leq n \leq 10$ vertices. Using the results of the computations we define the instances $I_6^2, \ldots, I_{10}^2$.

Theorem 1.2.15. *The instances I_n^2 maximize the integrality ratio for* Rectilinear TSP *for $n \leq 10$.*

The instances I_n^2 show the same structure as the instances maximizing the integrality ratio in the metric case described in [6]. Based on this we state the following conjecture.

Conjecture 1.2.16. *The instances maximizing the integrality ratio among all instances with a fixed number of vertices have the following structure: An optimal fractional solution x^* of the subtour LP satisfies $x^*(e) = \frac{1}{2}$ for all edges e of two disjoint triangles and $x^*(e) = 0$ or $x^*(e) = 1$ for all other edges e.*

We analyze the structure of I_n^2 and generalize the family of instances to arbitrary numbers of vertices. We compute the integrality ratio of the family and show that the integrality ratios of the family converge to $\frac{4}{3}$ as $n \to \infty$.

Moreover, we also investigate the integrality ratio for Metric TSP and Multidimensional Rectilinear TSP.

Theorem 1.2.17. *Assuming Conjecture 1.2.16 the* Metric TSP *instances given in [6] maximize the integrality ratio.*

For the Multidimensional Rectilinear TSP we define a family of instances $(I_n^3)_{n \in \mathbb{N}}$ and show

Theorem 1.2.18. *Assuming Conjecture 1.2.16 the family of instances I_n^3 maximizes the integrality ratio for* Multidimensional Rectilinear TSP.

We investigate local optima of instances that can be embedded into $\mathbb{R}^d$ such that the distances arise from a totally differentiable norm. Such an instance is a local optimum if we cannot increase the integrality ratio by moving the vertices slightly. A criterion is given to detect local optima. Based on that we give a local search algorithm that computes a local optimum.

For the Euclidean TSP we use the local search algorithm to find instances with high integrality ratio. The results have similar structures as I_n^2 in the rectilinear case and the instances from [6] in the metric case. Based on these we give an efficient algorithm generating instances having this structure. Using this algorithm, we were able to generate instances with high integrality ratio for Euclidean TSP.

Furthermore, we investigate the runtime of the `Concorde` TSP solver on slightly modified I_n^3. We observe that the runtimes of `Concorde` for these instances are much higher than for the hard to solve instances given in [38].

For the Metric $s-t$ Path TSP we improve the previous best upper bound on the integrality ratio of $1 + \frac{1}{1+4\ln(\frac{5}{4})} > 1.5283$ by Traub and Vygen in [62] to 1.5273.

Theorem 1.2.19. *The integrality ratio of the standard LP relaxation for the* Metric $s-t$ Path TSP *is at most 1.5273.*

1.3 Structure of the Thesis

First, we briefly summarize in Section 1.4 the preliminaries we need for this thesis.

In Chapter 2 we consider the k-Opt algorithm on several TSP variants. We want to bound the approximation ratio in the worst case. We start by determining the exact approximation ratio of the 2-Opt algorithm in Section 2.1. Then, we improve the existing lower bound for the Metric TSP by weakening the condition for the construction of bad instances given in [17]. After that, we show in Section 2.2 the upper bound of the approximation ratio for Metric TSP. For that we assume that the optimal tour and the output of the k-Opt or Lin-Kernighan algorithm with the largest approximation ratio are given. Our aim is to show that the output of the algorithm does not have too many long edges compared to the optimal tour. To achieve this, we first divide the edges into length classes, such that the longest edge from each class is at most a constant times longer than the shortest. Then, we construct with help of the optimal tour a graph containing at least $\frac{1}{4}$ of the edges in a length class. We show that this graph has a high girth and use results from extremal graph theory to bound the number of its edges, which implies that the length class does not contain too many edges. In the last section, we compare the lower and upper bound we got from the previous sections and show that they differ asymptotically only by a logarithmic factor even if the exact behavior of $\mathrm{ex}(n, 2k)$ is unknown.

Furthermore, we give lower and upper bounds on the approximation ratio of the k-Opt algorithm for Graph TSP in Section 2.3. For the lower bound we construct an instance and a k-optimal tour with the appropriate approximation factor again using results from extremal graph theory. To show the upper bound, starting with a worst-case instance we iteratively decompose the current graph into smaller graphs with small diameter and contract these smaller graphs into single vertices. We show that a certain subset of the vertices, the so-called active vertices, shrinks by a factor exponential in the approximation ratio after a sufficient number of iterations. Moreover, we show that after that many iterations we still have at least one active vertex. We conclude that the number of active vertices and hence the number of vertices in the beginning depends exponentially on the approximation ratio.

Next, in Section 2.4 we show that if the vertices of the TSP instance can be embedded into $\mathbb{R}^d$ such that the distances function arises from the p-norm for some $p \geq 1$, then the approximation ratio of k-Opt is $\Omega\left(\frac{\log(n)}{\log\log(n)}\right)$. In Section 2.5 we give a lower bound of $\frac{3}{2}$ on the approximation ratio of 2-Opt for the (1,2)-TSP and show that the approximation ratio of the 3-Opt algorithm is $\frac{11}{8}$. We introduce a modified version of the k-Opt algorithm for (1,2)-TSP and show that its approximation ratio for $k = 3$ is $\frac{4}{3}$. Moreover, for the (1,2)-TSP we give a lower bound on the k-improv and k-Opt algorithm for arbitrary fixed k of $\frac{11}{10}$.

At the end of this chapter in Section 2.6 we give a polynomial time local search algorithm for Graph TSP that has in contrast to the k-Opt algorithm a constant approximation ratio.

In Chapter 3 we investigate the integrality ratio of the subtour LP. First, we start in Section 3.1 by describing a procedure of generating families of instances for Euclidean TSP whose integrality ratios converge to $\frac{4}{3}$. These instances can have a different structure than the currently known families of instances whose integrality ratios converge to $\frac{4}{3}$. In Section 3.2 we compute the exact integrality ratio for Rectilinear TSP instances with a small fixed number of vertices. In the following Section 3.3 we generalize the instances maximizing the integrality ratio we found in the previous section to arbitrary numbers of vertices. In Section 3.4 and 3.5 we identify the instances maximizing the integrality ratio assuming Conjecture 1.2.16 for the METRIC and MULTIDIMENSIONAL RECTILINEAR TSP, respectively.

Then, we give in Section 3.6 a criterion that certifies local optimality with respect to integrality ratio. Based on this criterion we describe a local search algorithm to find local optima. In Section 3.7 we give a more efficient algorithm to generate a family of instances for the EUCLIDEAN TSP that was found by the local search algorithm for small number of vertices and has similar structure to the instances maximizing the integrality ratio for other TSP variants. We compare the lower bounds we found for the various TSP variants in Section 3.8. Then, we observe in Section 3.9 that `Concorde` needs significantly more running time to solve slightly modified instances we found than to solve the hard to solve instances from the literature. At the end of the chapter we improve in Section 3.10 the upper bound on the integrality ratio for the METRIC $s-t$ PATH TSP to 1.5273.

Last, in Chapter 4 we give an outlook to the open problems unsolved in this thesis.

1.4 Prelimilaries

In this section we briefly introduce the basic notation, definitions and some previously known results used in this thesis. For a more detailed description of these topics we refer to standard textbooks about discrete mathematics and combinatorial optimization, for example [45].

1.4.1 Basic Definitions and Properties

Graphs and Graph Algorithms

An *undirected graph* $G = (V(G), E(G))$ consists of a set of *vertices* $V(G)$ and a set of *edges* $E(G) \subseteq \{\{v,w\}|v,w \in V(G), v \neq w\}$. It is called *directed* if the edge set instead satisfies $E(G) \subseteq \{(v,w) \in V(G) \times V(G)|v \neq w\}$. The complete graph K_n is an undirected graph with n vertices and all $\binom{n}{2}$ edges between each pair of vertices. A graph is called *weighted* if in addition a weight function $c : E(G) \to \mathbb{R}$ is given. For convenience we abbreviate $c(\{u,v\})$ by $c(u,v)$ for all $\{u,v\} \in E(G)$.

For an edge $e = \{v,w\}$ or $e = (v,w)$ the vertices v and w are also called the *endpoints* of e. For an edge $e = (v,w)$ of a directed graph v and w are called the *head* and *tail* of e, respectively. Two vertices v,w are called *adjacent* if $\{v,w\} \in E(G)$. A vertex v is called *incident* to an edge e if $v \in e$. Two edges are called *incident* if they share a common endpoint.

For a set $X \subseteq V(G)$ we define $\delta(X) := \{\{v,w\} \in E(G)|v \in X, w \notin X\}$ in the undirected case and $\delta^+(X) := \{(v,w) \in E(G)|v \in X, w \notin X\}$, $\delta^-(X) := \{(w,v) \in E(G)|v \in X, w \notin X\}$ in the directed case. If $X = \{v\}$ contains only one element, we simply write $\delta(v), \delta^+(v), \delta^-(v)$. For a vertex $v \in V(G)$ the numbers $|\delta(v)|, |\delta^+(v)|$ and $|\delta^-(v)|$ are also called the *degree*, *outdegree* and *indegree* of the vertex v, respectively. A *d-regular* graph G is an undirected graph with $|\delta(v)| = d$ for all $v \in V(G)$.

An undirected graph or directed graph with *self-loops* can contain additional edges where both endpoints are the same vertex. If we further allow multiple copies of the same edge, we get a *multigraph*. Given a directed graph G the *underlying undirected graph* is a multigraph with the vertex set $V(G)$ and the edge multiset $\{\{u,v\}|(u,v) \in E(G)\}$. The *support graph* of a weighted graph G is the graph with the same vertex set and whose edge set consists of the edges with positive weight in G.

A *walk* in a graph G is a sequence of its vertices $v_0, v_1, \dots, v_l$ such that $\{v_i, v_{i+1}\} \in E(G)$ or $(v_i, v_{i+1}) \in E(G)$ for all $i \in \{0, \dots, l-1\}$. A walk is called *closed* if in addition $v_0 = v_l$. A *path* is a walk with $v_i \neq v_j$ for all $i \neq j$ and $i, j \in \{0, \dots, l\}$. Similarly, a *cycle* is a closed walk with $v_i \neq v_j$ for all $i \neq j$ and $i, j \in \{0, \dots, l-1\}$. A graph is called *connected* if there is a path between every pair of vertices.

A *Eulerian walk* of a graph G is a closed walk using every edge of G exactly once. A graph is called *Eulerian* if it has a Eulerian walk. The following well-known theorem from Euler characterizes Eulerian graphs:

Theorem 1.4.1 (Euler's Theorem [25])**.** *A connected undirected graph is Eulerian if and only if all of its vertices have even degree.*

Given a graph G and an even subset of vertices $T \subseteq V(G)$ a *T-join* J is a set of edges such that $|J \cap \delta(v)|$ is odd if and only if $v \in T$. Therefore, if T is the set of vertices with odd degree in a connected graph G we can add J to $E(G)$ to make G Eulerian.

We need the following result on the number of colors to color the edges of a bipartite graph such that no two incident edges have the same color.

Theorem 1.4.2 (Kőnig [44])**.** *Every bipartite graph with maximal degree Δ can be Δ-edge colored such that no two incident edges have the same color.*

We can find in polynomial time a negative cycle in a given weighted directed graph or determine that none exists.

Theorem 1.4.3 (Moore, Bellman, Ford [51, 5, 27])**.** *Given a directed weighted graph G, the Moore-Bellman-Ford algorithm finds a negative cycle or returns that there is no negative cycle in $O(|V(G)||E(G)|)$ time.*

Asymptotic Notation

Given two real valued functions f, g, we write

$$
\begin{aligned}
f \in O(g) \qquad & \text{if} \qquad \limsup_{x\to\infty} \left|\frac{f(x)}{g(x)}\right| < \infty, \\
f \in \Omega(g) \qquad & \text{if} \qquad \liminf_{x\to\infty} \left|\frac{f(x)}{g(x)}\right| > 0, \\
f \in \Theta(g) \qquad & \text{if} \qquad f \in O(g) \text{ and } f \in \Omega(g).
\end{aligned}
$$

p-Norm

The p-norm $\|\cdot\|_p$ for $p \geq 1$ in $\mathbb{R}^d$ is defined for all $x = (x_1, \dots, x_d) \in \mathbb{R}^d$ by $\|x\|_p := \sqrt[p]{\sum_{i=1}^d |x_i|^p}$, where $\sqrt[1]{z} := z$. The 1-norm and 2-norm are also called the *Manhatten* and *Euclidean* norm, respectively. For two points $x, y \in \mathbb{R}^d$ we denote the distance of x and y according to the p-norm by $\text{dist}_p(x, y)$, i.e. $\text{dist}_p(x, y) := \|x - y\|_p$.

1.4.2 Traveling Salesman Problem

An instance of Metric TSP is given by a complete weighted graph (K_n, c) where the costs are non-negative and satisfy the triangle inequality: $c(x,z) + c(z,y) \geq c(x,y)$ for all $x, y, z \in V(K_n)$. A *tour* is a cycle that visits every vertex exactly once. For a tour T, let the *length* of the tour be defined as $c(T) := \sum_{e \in T} c(e)$. The task is to find a tour of minimal length. We fix an *orientation* of the tour, i.e. we consider the edges of the tour as directed edges such that the tour is a directed cycle. From now on, let n denote the number of vertices of the instance.

Graph TSP is a special case of the Metric TSP. Each instance arises from an unweighted, undirected connected graph G. To construct a TSP instance (K_n, c), we set $V(K_n) = V(G)$. The cost $c(u,v)$ of the edge connecting any two vertices $u, v \in V(G)$ is given by the length of the shortest u-v-path in G.

Other special cases of the TSP are the Rectilinear TSP and the Euclidean TSP. In these cases the vertices of the instances can be embedded into $\mathbb{R}^2$ such that the distances are induced by the Manhattan and Euclidean norm, respectively. Instead of embedding the vertices into $\mathbb{R}^2$ we can also consider instances that can be embedded into the general space $\mathbb{R}^d$ to get the multidimensional variant of these problems.

For the (1,2)-TSP the distances between the cities are restricted to be equal to 1 or 2. Note that this variant of the TSP is metric since three edges of length 1 or 2 always satisfy the triangle inequality.

The $s - t$ Path TSP is a generalization of the TSP where two vertices s and t are specified and the task is to find a shortest path starting in s and ending in t visiting all other vertices exactly once. If s is equal to t, then we have the standard TSP.

An algorithm A for the traveling salesman problem has *approximation ratio* $\alpha(n) \geq 1$ if for every TSP instance with n vertices it finds a tour that is at most $\alpha(n)$ times as long as a shortest tour and this ratio is achieved by an instance for every n. Note that we require here the sharpness of the approximation ratio deviating from the standard definition in the literature to express the approximation ratio in terms of the Landau symbols. Nevertheless, the results also hold for the standard definition with more complicated notation.

1.4.3 k-Opt Algorithm

A *k-move* replaces at most k edges of a given tour by other edges to obtain a new tour. It is called *improving* if the resulting tour is shorter than the original one. A tour is called *k-optimal* if there is no improving k-move (Algorithm 1).

Algorithm 1 k-Opt Algorithm

Input: Instance of TSP (K_n, c)
Output: Tour T

```
Start with an arbitrary tour T
while ∃ improving k-move for T do
    Perform an improving k-move on T
return T
```

For the 2-Opt algorithm recall the following well-known fact: Given a tour T with a fixed orientation, it stays connected if we replace two edges of T by the edge connecting their heads and the edge connecting their tails, i.e. if we replace edges $(a, b), (c, d) \in T$ by (a, c) and (b, d).

1.4.4 k-Improv Algorithm

In this section we describe the k-improv algorithm, which is an improved version of the k-Opt algorithm for the (1,2)-TSP introduced by Berman and Karpinski in [9]. In the same paper it was shown that this algorithm has an approximation ratio of $\frac{8}{7}$ for $k = 15$ which is the currently best approximation ratio for the (1,2)-TSP.

A *2-matching* is the union of disjoint paths and cycles. A *k-improv-move* deletes and adds in total at most k edges of a 2-matching to obtain a new 2-matching. Note that in contrast to a k-move the number of removed and added edges do not have to be equal. A k-improv-move is called *improving* if the result $\widetilde{T}'$ after performing the k-improv-move on $\widetilde{T}$ satisfies the following conditions:

1. $\widetilde{T}'$ only contains edges with cost 1.

2. One of the following properties hold:
   - $\widetilde{T}'$ contains less connected components than $\widetilde{T}$.
   - $\widetilde{T}'$ contains the same number of connected components as $\widetilde{T}$, but more cycles than $\widetilde{T}$.
   - $\widetilde{T}'$ contains the same number of connected components and cycles as $\widetilde{T}$, but less *singletons*, i.e. vertices with degree 0, than $\widetilde{T}$.

The algorithm is also a local search algorithm. It starts with an arbitrary tour and removes all edges with cost 2 to obtain a 2-matching consisting of edges with cost 1. During each iteration of the algorithm we perform an improving k-improv-move. Note that this way we maintain a 2-matching in every iteration. We call a 2-matching *k-improv-optimal* if there are no improving k-improv-moves. If this is the case, we remove an arbitrary edge from every cycle in $\widetilde{T}$ and after that connect the paths in $\widetilde{T}$ arbitrarily to a tour T (Algorithm 2).

Algorithm 2 k-Improv Algorithm

Input: Instance of (1,2)-TSP (K_n, c)
Output: Tour T

```
Start with an arbitrary tour T
Let T̃ be the 2-matching we obtain by removing all edges of cost 2 from T
while ∃ improving k-improv-move for T̃ do
    Perform an improving k-improv-move on T̃
Remove an arbitrary edge from each cycle in T̃
Connect the paths in T̃ arbitrarily to a tour T
return T
```

We note that the k-improv algorithm for fixed k runs in polynomial time as shown in [9]. Moreover, by the procedure of the algorithm we can see a correspondence between 2-matchings consisting of edges with cost 1 and tours: The *corresponding 2-matching* of a tour T is obtained by removing all edges with cost 2. The set of *corresponding tours* of a 2-matching is obtained by removing an arbitrary edge from each cycle of the 2-matching and connect the paths arbitrarily to a tour.

1.4.5 Lin-Kernighan Algorithm

We use a parameterized version of the Lin-Kernighan algorithm described in Section 21.3 of [45] for the analysis. In this version two parameters p_1 and p_2 specify the depth the algorithm is searching for improvement.

An *alternating walk* of a tour T is a walk where exactly one of two consecutive edges is in T. An edge of the alternating walk is called *tour edge* if it is contained in T, otherwise it is called *non-tour edge.* A *closed alternating walk* and *alternating cycle* are alternating walks whose edges form a closed walk and cycle, respectively. The symmetrical difference of two sets A and B is the set $A \triangle B := (A \cup B) \setminus (A \cap B)$. When we *augment* T by an augmenting cycle C we get the result $T \triangle C$. Moreover, C is called *improving* if $T \triangle C$ is a shorter tour than T. By $(x_1, x_2, \ldots, x_j) := \cup_{i=1}^{j-1} (x_i, x_{i+1})$ we denote the walk that visits the vertices $x_1, x_2 \ldots, x_j$ in this order. We define the *gain* g of an alternating walk starting with a tour edge by

$$g((x_0, x_1, \ldots, x_{2m})) := \sum_{i=0}^{m-1} c(x_{2i}, x_{2i+1}) - c(x_{2i+1}, x_{2i+2}).$$

An alternating walk $(x_0, x_1, \ldots, x_{2m})$ is *proper* if it starts with a tour edge and $g((x_0, x_1, \ldots, x_{2i})) > 0$ for all $i \leq m$.

The following theorem by Lin and Kernighan allows performance improvements of the Lin-Kernighan algorithm by only looking for proper alternating walks without changing the quality of the result.

Theorem 1.4.4 ([49])**.** *For every improving closed alternating walk P there exists a proper closed alternating walk Q with $E(P) = E(Q)$.*

Now, we state the generalized version of the Lin-Kernighan algorithm with parameters p_1 and p_2 (Algorithm 3):

- The algorithm starts with an arbitrary tour and searches for an improving closed alternating walk in every iteration by a depth-first search.
- At depth zero the list of *candidate vertices* consists of all vertices of the instance.
- At each depth it chooses a vertex from the list of candidate vertices, computes the list of candidate vertices for the next depth and increases the depth.
- The list of candidate vertices consists of all vertices forming with the vertices already chosen in previous iterations an alternating walk starting with a tour edge and having a positive gain.
- At each depth it checks if connecting the endpoints of the alternating walk results in an improving closed alternating walk.
- When the depth is higher than p_2 and even we further require the vertices of the candidate vertices list satisfying the following condition: After choosing any vertex from the candidate vertices list in the next iteration and connecting the endpoints of the resulting alternating walk, we get an improving closed alternating walk.
- When no candidates are available at depth i anymore, it backtracks to the depth $\min\{p_1, i-1\}$ and chooses the next candidate at that depth.

- It terminates if no improving closed alternating walk is found. Otherwise, it improves the current tour by augmenting the improving closed alternating walk with the highest gain it found and repeats the process.

Algorithm 3 Lin-Kernighan Algorithm

Input: Instance of TSP (K_n, c), Parameters $p_1, p_2 \in \mathbb{N}$
Output: Tour T

Start with an arbitrary tour T
Set $X_0 := V(K_n), i := 0$ and $g^* := 0$
while $i \geq 0$ **do**
  if $X_i = \emptyset$ **then**
    if $g^* > 0$ **then**
      Set $T := T \triangle P^*$
      Set $X_0 := V(K_n), i := 0$ and $g^* := 0$
    else
      Set $i := \min\{i - 1, p_1\}$
  else
    Choose $x_i \in X_i$, set $X_i := X_i \backslash \{x_i\}$
    Set $P := (x_0, x_1, \ldots, x_i)$
    if i is odd **then**
      if $i \geq 3, T \triangle (P \cup (x_i, x_0))$ is a tour, $g(P \cup (x_i, x_0)) > g^*$ **then**
        Set $P^* := P \cup (x_i, x_0)$ and $g^* := g(P^*)$
      Set $X_{i+1} := \{x \in V(K_n) \backslash \{x_0, x_i\} : \{x, x_0\} \notin T \cup P, T \triangle (P \cup (x_i, x, x_0))$ is a tour, $g(P \cup (x_i, x)) > g^*\}$
    if i is even **then**
      if $i \leq p_2$ **then**
        Set $X_{i+1} := \{x \in V(K_n) : \{x_i, x\} \in T \backslash P\}$
      else
        Set $X_{i+1} := \{x \in V(K_n) : \{x_i, x\} \in T \backslash P, \{x, x_0\} \notin T \cup P, T \triangle (P \cup (x_i, x, x_0))$ is a tour$\}$
    $i := i + 1$
return T

In the original paper Lin and Kernighan described the algorithm with fixed parameters $p_1 = 5, p_2 = 2$.

Definition 1.4.5. We call the Lin-Kernighan algorithm with parameter $p_1 = 2k - 1$ and $p_2 = 2k - 4$ the *k-Lin-Kernighan algorithm.* A tour is *k-Lin-Kernighan optimal* if it is the output of the k-Lin-Kernighan algorithm for some initial tour.

Note that the original version of the Lin-Kernighan algorithm is the 3-Lin-Kernighan algorithm. By the description of the algorithm it is easy to see that all local changes of the Lin-Kernighan algorithm are augmentations of an improving closed alternating walk and:

Lemma 1.4.6. *The length of any improving alternating cycle in a k-Lin-Kernighan optimal tour is at least $2k + 1$.*

Obviously, this property also holds for the output of the Lin-Kernighan algorithm with parameters $p_1 \geq 2k - 1, p_2 \geq 2k - 4$ and our results carry over in this case.

1.4.6 Girth and Ex

Definition 1.4.7. The *girth* of a graph is the length of the shortest cycle contained in the graph if it contains a cycle and infinity otherwise. Let $\mathrm{ex}(n, 2k)$ be the maximal number of edges in a graph with n vertices and girth at least $2k$. Moreover, define $\mathrm{ex}^{-1}(m, 2k)$ as the minimal number of vertices of a graph with m edges and girth at least $2k$.

There are previous results in the extremal graph theory on the behavior of the function $\mathrm{ex}(n, 2k)$.

Theorem 1.4.8 ([2])**.** *We have*

$$\mathrm{ex}(n, 2k) < \frac{1}{2^{1+\frac{1}{k-1}}} n^{1+\frac{1}{k-1}} + \frac{1}{2}n.$$

Theorem 1.4.9 ([47])**.** *We have*

$$\mathrm{ex}(n, 2k) = \Omega(n^{1+\frac{2}{3k-6+\epsilon}}),$$

where $k \geq 3$ is fixed, $\epsilon = 0$ if k is even, $\epsilon = 1$ if k is odd and $n \to \infty$.

Theorem 1.4.10 (Polarity Graph in [22, 23, 16]; Construction by Benson and by Singleton [7, 60]; Construction by Benson and by Wenger [7, 65])**.** *For $k = 3, 4, 6$ we have*

$$\mathrm{ex}(n, 2k) = \Omega(n^{1+\frac{1}{k-1}}).$$

Theorem 1.4.11 (Theorem 1.4' in Section III of [11])**.** *Let $\delta, g \geq 3$ and*

$$m \geq \frac{(\delta - 1)^{g-1} - 1}{\delta - 2}$$

be integers. Then, there exists a δ-regular graph with $2m$ vertices and girth at least g.

1.4.7 Linear Programming

A *linear program* (LP) is given by a matrix $A \in \mathbb{R}^{m \times n}$ and vectors $b \in \mathbb{R}^m, c \in \mathbb{R}^n$. The task is to find a vector $x \in \mathbb{R}^n$ maximizing $c^\top x$ such that $Ax \leq b$, decide that no such x exists or decide that for all $\alpha \in \mathbb{R}$ there is such a x with $c^\top x > \alpha$. The corresponding *dual program* is given by minimizing $y^\top b$ such that $y^\top A = c^\top, y \geq 0$ for some $y \in \mathbb{R}^m$. The original LP is also called the *primal LP*. If we further require x to be integral we get an *integer program.*

An important result shows the relation between the optimal primal and dual solutions:

Theorem 1.4.12 (Complementary Slackness)**.** *Let a primal dual LP pair $\{\max c^\top x : Ax \leq b\}$ and $\{\min y^\top b : y^\top A = c^\top, y \geq 0\}$ with feasible solutions x and y be given. The following statements are equivalent:*

- *x and y are optimal solutions*
- $c^\top x = y^\top b$
- $y^\top (b - Ax) = 0.$

1.4.8 Subtour LP and Integrality Ratio

One of the most common linear relaxation of the TSP is the *subtour LP* [19]. For a given TSP instance (K_n, c) the subtour LP is given by:

$$\begin{aligned}
\min \sum_{e\in E(K_n)} c(e)x_e & & \\
\sum_{e\in\delta(v)} x_e = 2 & \qquad \text{for all } v \in V(K_n) & (1.1)\\
\sum_{e\in E(\delta(X))} x_e \geq 2 & \qquad \text{for all } \emptyset \subset X \subset V(K_n). & (1.2)\\
0 \leq x_e \leq 1 & \qquad \text{for all } e \in E(K_n) &
\end{aligned}$$

The constraints (1.1) are called the *degree constraints* and the constraints (1.2) are called the *subtour elimination constraints.* Although this LP has an exponential number of constraints it can be solved in polynomial time by the ellipsoid method since the separation problem can be solved efficiently [33]. A solution to the subtour LP is also called a *fractional tour.*

Cunningham [50], Goemans and Bertsimas [31] showed that in the metric case the optimal solution does not change if we omit the degree constraints, i.e. the optimal solutions of the following LP is equal to the optimal solutions of the subtour LP:

$$\begin{aligned}
\min \sum_{e\in E(K_n)} c(e)x_e & \\
\sum_{e\in E(\delta(X))} x_e \geq 2 & \qquad \text{for all } \emptyset \subset X \subset V(K_n)\\
0 \leq x_e \leq 1 & \qquad \text{for all } e \in E(K_n)
\end{aligned}$$

This above LP is an LP relaxation for the 2-Edge Connected Spanning Subgraph problem where the task is to find a 2-edge connected spanning subgraph of a given graph.

Theorem 1.4.13 (Cunningham [50], Goemans and Bertsimas [31])**.** *If the costs satisfy the triangle inequality, the optimal solutions of the* 2-Edge Connected Spanning Subgraph *LP is the same as that of the subtour LP.*

The subtour LP can be modified to apply for the $s-t$ Path TSP.

$$\begin{aligned}
\min \sum_{e\in E(K_n)} c(e)x_e & \\
\sum_{e\in\delta(v)} x_e = 2 & \qquad \text{for all } v \in V(K_n)\backslash\{s,t\}\\
\sum_{e\in\delta(v)} x_e = 1 & \qquad \text{for all } v \in \{s,t\}\\
\sum_{e\in\delta(X)} x_e \geq 2 & \qquad \text{for all } \emptyset \subset X \subseteq V(K_n)\backslash\{s,t\}\\
\sum_{e\in\delta(X)} x_e \geq 1 & \qquad \text{for all } \{s\} \subseteq X \subseteq V(K_n)\backslash\{t\}\\
x_e \geq 0 & \qquad \text{for all } e \in E(K_n).
\end{aligned}$$

Let $OPT(I)$ and $OPT_{LP}(I)$ be the values of the optimal integral solution and optimal fractional solution of an instance I, then the *integrality ratio* of I is defined as $\frac{OPT(I)}{OPT_{LP}(I)}$. The *integrality ratio* of the LP is the supremum of the ratio between the value of the optimal integral solution and that of the optimal fractional solution, i.e. $\sup_I \frac{OPT(I)}{OPT_{LP}(I)}$. Let $|I|$ be the number of vertices of the instance I. The *integrality ratio* of instances with n vertices is defined as $\sup_{|I|=n} \frac{OPT(I)}{OPT_{LP}(I)}$.

1.4.9 Structure of Euclidean Tours

A well-known result about optimal tours for Euclidean TSP is that they do not intersect themself unless all vertices lie on a line.

Lemma 1.4.14 (Flood 1956 [26])**.** *Unless all vertices lie on one line, an optimal tour of a* Euclidean TSP *instance is a simple polygon.*

An important consequence of Lemma 1.4.14 is the following result:

Lemma 1.4.15 ([20], page 142)**.** *An optimal tour of a* Euclidean TSP *instance visits the vertices on the boundary of the convex hull of all vertices in their cyclic order.*

1.4.10 Karamata's inequality

Definition 1.4.16. A sequence of real numbers $x_1, \dots, x_n$ *majorizes* another sequence $y_1, \dots, y_n$ if

$$\begin{aligned} x_1 \geq x_2 \cdots &\geq x_n \\ y_1 \geq y_2 \cdots &\geq y_n \\ \sum_{i=1}^{j} x_i &\geq \sum_{i=1}^{j} y_i \qquad \forall j < n \\ \sum_{i=1}^{n} x_i &= \sum_{i=1}^{n} y_i. \end{aligned}$$

Theorem 1.4.17 (Karamata's inequality [40])**.** *Let I be an interval of real numbers and $f : I \to \mathbb{R}$ be a convex function. Moreover, let $x_1, \dots, x_n$ and $y_1, \dots, y_n$ be sequences of numbers in I such that $(x_1, \dots, x_n)$ majorizes $(y_1, \dots, y_n)$. Then*

$$\sum_{i=1}^{n} f(x_i) \geq \sum_{i=1}^{n} f(y_i).$$

2 Approximation Ratio of the k-Opt and Lin-Kernighan Algorithm

2.1 Exact Approximation Ratio of 2-Opt for Metric TSP

In this section we show that the exact approximation ratio of the 2-Opt algorithm for Metric TSP is $\sqrt{\frac{n}{2}}$ by improving the existing upper and lower bound. This section is based on joint work with Stefan Hougardy and Fabian Zaiser, which appeared in [36]. In Fabian Zaiser's master's thesis supervised by Stefan Hougardy the same lower bound was developed for $n = 2k^2$ for k a power of 2. I modified the instance and removed the condition that k needs to be a power of 2. Moreover, I developed the matching upper bound for the approximation ratio.

2.1.1 Upper Bound

Chandra, Karloff, and Tovey [17] proved in 1999 that the 2-Opt algorithm has an approximation ratio of $4\sqrt{n}$ for Metric TSP. In 2013, Levin and Yovel [48] observed that their proof yields the upper bound $2\sqrt{2n}$. Here we present a new proof which improves this bound by a factor of 4:

Theorem 2.1.1. *The approximation ratio of the 2-Opt algorithm on* Metric TSP *is at most* $\sqrt{\frac{n}{2}}$.

Proof. Let $G = (V(G), E(G))$ with $c : E(G) \to \mathbb{R}_{\geq 0}$ and $|V(G)| = n$ be a Metric TSP instance and let T be an optimal tour. We may assume that T has length 1. We fix an orientation of the tour T and choose two vertices $p, q \in V(G)$ arbitrarily. For each vertex $v \in V(G)$, let $i_p(v)$ be the length taken mod 1 of the unique shortest directed p-v path starting in p and using only edges of T. By our assumption, we have $i_p : V(G) \to [0, 1)$ and we define i_q similarly. For the following, it helps to think of $[0, 1)$ as the circle with circumference 1 and of i_p as an embedding of the optimal tour into this circle such that the arc distance of two consecutive vertices on the circle is the length of the edge between them.

Define the following metric d on the interval $[0, 1)$, interpreted as a circle: $d(x, y)$ is the length of the shorter of the two arcs between x and y on the circle, i.e., $d(x, y) := \min\{|x - y|, 1 - |x - y|\}$. For any points $x, y, z \in [0, 1)$ we have $d(x, y) + d(y, z) \geq d(x, z)$ since combining the two shortest arcs between x, y and y, z and deleting the overlap results in an arc between x, z.

Let T' be a 2-optimal tour. As usual, we assume that it is directed. Now, consider for each edge (u, v) of T' the set

$$S_{p,q}(u, v) = \{(x, y) \in [0, 1) \times [0, 1) \mid d(x, i_p(u)) + d(y, i_q(v)) < c(u, v)\},$$

as shown in Figure 2.1. We claim that all these sets are pairwise disjoint for distinct edges $(u_1, v_1), (u_2, v_2) \in E(T')$. Suppose that $S_{p,q}(u_1, v_1)$ and $S_{p,q}(u_2, v_2)$ intersect in

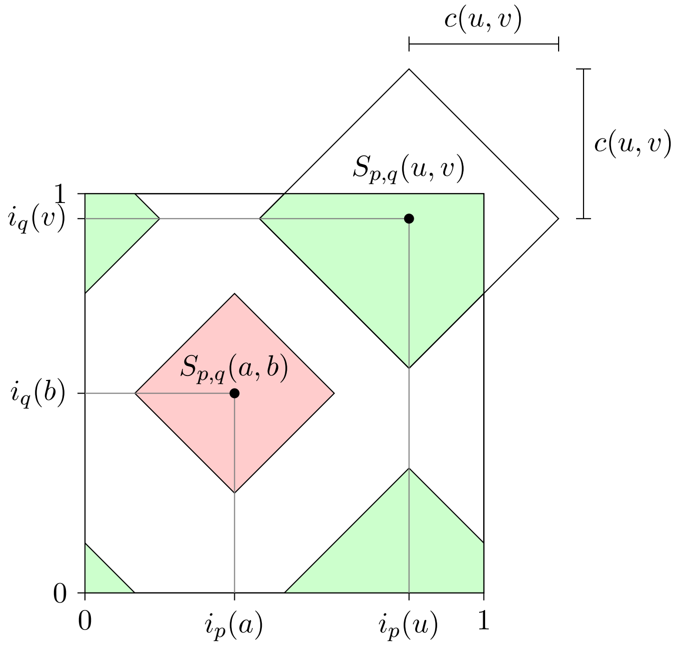


Figure 2.1: The sets $S_{p,q}(a,b)$ (red) and $S_{p,q}(u,v)$ (green) assigned to the edges (a,b) and (u,v) of a 2-optimal tour. The sets are taken modulo the unit square and thus may consist of up to four parts.

(x,y). Then, by the triangle inequality for c and d, we have

$$\begin{aligned} c(u_1,u_2)+c(v_1,v_2) &\le d(i_p(u_1),i_p(u_2))+d(i_q(v_1),i_q(v_2)) \\ &\le d(i_p(u_1),x)+d(x,i_p(u_2))+d(i_q(v_1),y)+d(y,i_q(v_2)) \\ &< c(u_1,v_1)+c(u_2,v_2). \end{aligned}$$

This contradicts the 2-optimality of T'. Hence, all these sets $S_{p,q}(u,v)$ are disjoint.

Next, we want to show that the area of each set is independent of the choice of p and q. Let p' and q' be a different choice. Note that for all vertices u, we have $i_{p'}(u) = i_{p'}(p)+i_p(u) \bmod 1$. In particular, we find $d(x,i_p(u)) = d(x+i_{p'}(p) \bmod 1, i_{p'}(u))$ because both points are shifted by $i_{p'}(p)$ on the circle $[0,1)$. By the definition of $S_{p,q}(u,v)$, this means that the map

$$\begin{aligned} t : [0,1)\times[0,1) &\to [0,1)\times[0,1) \\ (x,y) &\mapsto (x+i_{p'}(p) \bmod 1, y) \end{aligned}$$

bijectively sends $S_{p,q}(u,v)$ to $S_{p',q}(u,v)$. In other words, we obtain $S_{p',q}(u,v)$ from $S_{p,q}(u,v)$ by cutting the unit square vertically at $1-i_{p'}(p) = i_p(p')$ into two rectangles and reassembling them, as described by the following two translations:

$$\begin{aligned} t_1 : [0,i_p(p'))\times[0,1) &\to [i_{p'}(p),1)\times[0,1) \\ (x,y) &\mapsto (x+i_{p'}(p),y) \\ t_2 : [i_p(p'),1)\times[0,1) &\to [0,i_p(p'))\times[0,1) \\ (x,y) &\mapsto (x-i_p(p'),y) \end{aligned}$$

Since they have disjoint domains and disjoint images, their union $t = t_1 \cup t_2$ is a bijection $[0,1) \times [0,1) \to [0,1) \times [0,1)$; sends $S_{p,q}(u,v)$ bijectively to $S_{p',q}(u,v)$; and preserves the area of this set because it consists of translations. Analogously, we can cut the square horizontally at $i_q(q')$ to obtain $S_{p',q'}(u,v)$ from $S_{p',q}(u,v)$, again preserving its area. We conclude that the area of $S_{p,q}(u,v)$ is independent of the choice of p and q.

Now, we want to show that the area of $S_{p,q}(u,v)$ is $2c(u,v)^2$ for any edge $(u,v) \in E(T')$. By the previous paragraph, we can choose $p = u$ and $q = v$. Then $S_{u,v}(u,v) = \{(x,y) \in [0,1) \times [0,1) \mid d(x,0) + d(y,0) < c(u,v)\}$. This set consists of four congruent isosceles right-angled triangles whose legs have length $c(u,v)$. Note that they do not overlap because the metric property ensures $c(u,v) \leq \frac{1}{2}$. Hence we have: $\text{area}(S_{p,q}(u,v)) = 4 \cdot \frac{c(u,v)^2}{2} = 2c(u,v)^2$.

Since the sets $S_{p,q}(u,v)$ for $(u,v) \in E(T')$ are pairwise disjoint, their combined area cannot exceed that of the unit square:

$$2 \sum_{e \in E(T')} c(e)^2 = \sum_{(u,v) \in E(T')} \text{area}(S_{p,q}(u,v)) \leq \text{area}([0,1) \times [0,1)) = 1.$$

Then the inequality of arithmetic and quadratic means implies

$$\frac{\sum_{e \in E(T')} c(e)}{n} \leq \sqrt{\frac{\sum_{e \in E(T')} c(e)^2}{n}} \leq \frac{1}{\sqrt{2n}}.$$

Hence, the length of the 2-optimal tour T' satisfies $\sum_{e \in E(T')} c(e) \leq \sqrt{\frac{n}{2}}$. □

2.1.2 Lower Bound

To prove a lower bound α on the approximation ratio of the 2-Opt algorithm for the Metric TSP, one has to show that for infinitely many n, there exists a Metric TSP instance with n cities that contains a 2-optimal tour which is α times longer than a shortest tour.

In 1999, Chandra, Karloff, and Tovey [17] provided such a construction for all n of the form $4 \cdot k^2$ for positive integers k, which shows a lower bound of $\frac{1}{4}\sqrt{n}$. Several years earlier, Plesník [53] had given another construction without explicitly stating a lower bound. It turns out that his construction yields a lower bound of $\frac{1}{\sqrt{8}}\sqrt{n}$ and works for all n of the form $8 \cdot k^2 - 8 \cdot k + 3$ for positive integers k.

The following result improves Plesník's lower bound by a factor of 2, and yields the tight result stated in Theorem 1.2.1.

Theorem 2.1.2. *The approximation ratio of the 2-Opt algorithm on the* Metric TSP *is at least* $\sqrt{\frac{n}{2}}$.

Proof. Let G be a complete graph on $n := 2 \cdot k^2$ nodes with vertex set $V(G) := \{v_{i,j} \mid 1 \leq i, j \leq k\} \cup \{w_{i,j} \mid 1 \leq i, j \leq k\}$. For each i with $1 \leq i \leq k$, we call $V_i := \{v_{i,j} \mid 1 \leq j \leq k\}$ and $W_i := \{w_{i,j} \mid 1 \leq j \leq k\}$ a *section* of $V(G)$ and the v-vertices and w-vertices the two *halves* of $V(G)$.

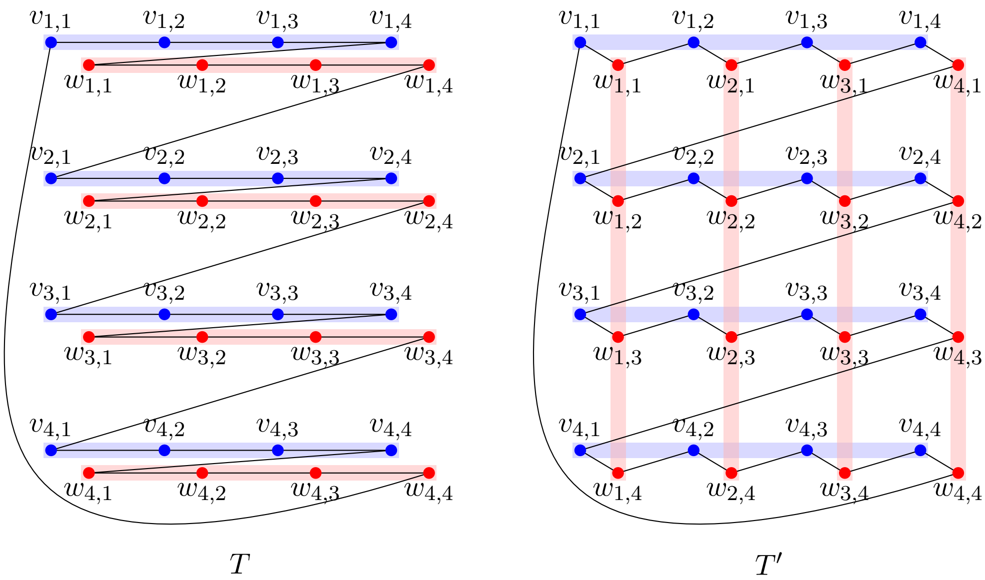


Figure 2.2: The optimal tour T (left) and the 2-optimal tour T' (right) for $k = 4$. Note that the w-vertices on the right are mirrored at the diagonal compared to the w-vertices on the left. Thus, on the left, vertices within the sections V_i and W_i are in a row. On the right, the vertices in the sections V_i are in a row while the vertices in a section W_i are within a column. The colored bars contain the vertices belonging to the same section.

We define a distance function $c : E(G) \to \mathbb{R}_{\geq 0}$ as follows:

$$c(v_{i,j}, w_{i',j'}) = \quad 1 \qquad \text{for all } 1 \leq i, i', j, j' \leq k$$

$$c(v_{i,j}, v_{i',j'}) = \begin{cases} 0 & i = i' \\ 2 & i \neq i' \end{cases} \qquad \text{for all } 1 \leq j, j' \leq k$$

$$c(w_{i,j}, w_{i',j'}) = \begin{cases} 0 & i = i' \\ 2 & i \neq i' \end{cases} \qquad \text{for all } 1 \leq j, j' \leq k$$

It is not hard to see that the function c satisfies the triangle inequality: Let u, v, w be any three vertices in $V(G)$. We want to show that $c(u, w) \leq c(u, v) + c(v, w)$. As c takes only the values $0, 1, 2$, this is obvious if $c(u, v) \geq 1$ and $c(v, w) \geq 1$. Otherwise, without loss of generality, we may assume that $c(u, v) = 0$. i.e., u and v are in the same section of $V(G)$. But then the definition of c implies $c(u, w) = c(v, w)$ and the triangle inequality is satisfied. Therefore, the graph G with cost function c is a Metric TSP instance.

In the following, we will construct two special tours in G, which are depicted in Figure 2.2. Let T be the tour consisting of the edges

$$\begin{aligned} T := \; & \{(v_{i,j}, v_{i,j+1}) \mid 1 \leq i \leq k, 1 \leq j < k\} \cup \\ & \{(w_{i,j}, w_{i,j+1}) \mid 1 \leq i \leq k, 1 \leq j < k\} \cup \\ & \{(v_{i,k}, w_{i,1}) \mid 1 \leq i \leq k\} \cup \\ & \{(w_{i,k}, v_{i+1,1}) \mid 1 \leq i < k\} \cup \\ & \{(w_{k,k}, v_{1,1})\}. \end{aligned}$$

The edges in the first two sets have length 0; the $2k$ edges in the other three sets have length 1. Therefore, we have $c(T) = 2k$. This tour is optimal because any tour has to visit all $2k$ sections of $V(G)$ and the distance of two vertices from different sections is at least 1.

Next we consider the tour T' with

$$\begin{aligned} T' := \ &\{(v_{i,j}, w_{j,i}) \mid 1 \le i, j \le k\} \cup \\ &\{(w_{j,i}, v_{i,j+1}) \mid 1 \le i \le k, 1 \le j < k\} \cup \\ &\{(w_{k,i}, v_{i+1,1}) \mid 1 \le i < k\} \cup \\ &\{(w_{k,k}, v_{1,1})\}. \end{aligned}$$

Each edge of T' has length 1. Thus we have $c(T') = 2k^2$. We claim that the tour T' is 2-optimal. Assume by contradiction that T' is not 2-optimal. Consider a pair of edges $(a, b), (x, y)$ that allows an improving 2-change to $(a, x), (b, y)$. Hence $c(a, x) + c(b, y) < c(a, b) + c(x, y) = 2$ and one of $c(a, x)$ or $c(b, y)$ must be zero. This means a and x or b and y must be in the same section. But since a and b are in opposite halves of $V(G)$ (just like x and y), this means that a and x are in one half of $V(G)$ and b and y in the other. Hence $c(a, x), c(b, y) \in \{0, 2\}$. For an improving 2-change, we must have $c(a, x) = c(b, y) = 0$. This implies that a and x lie in the same section of $V(G)$ and b and y lie in the same section of $V(G)$. Thus there must exist indices i and j with $1 \le i, j \le k$ such that $a, x \in V_i$ and $b, y \in W_j$ or such that $a, x \in W_i$ and $b, y \in V_j$. This implies that there must exist two different edges from V_i to W_j or from W_i to V_j. However, this is a contradiction as by definition of T', for any pair i, j with $1 \le i, j \le k$, there exists exactly one edge directed from V_i to W_j (namely the edge $(v_{i,j}, w_{j,i})$) and exactly one edge directed from W_j to V_i. This proves the 2-optimality of T'.

Combining the above findings we get

$$\frac{c(T')}{c(T)} = \frac{2k^2}{2k} = k = \sqrt{\frac{2k^2}{2}} = \sqrt{\frac{n}{2}}.$$

$\square$

2.2 Approximation Ratio of k-Opt and Lin-Kernighan for Metric TSP

In this section we investigate the approximation ratio of the k-Opt and Lin-Kernighan algorithm for Metric TSP. We show that assuming the Erdős girth conjecture the approximation ratio of the k-Opt algorithm is in $\Theta(\sqrt[k]{n})$. More generally, we prove an upper bound on the approximation ratio that is tight up to a factor of $O(\log(n))$ and depends on functions from extremal graph theory. The upper bounds can be carried over to a parameterized version of the Lin-Kernighan algorithm for appropriate parameters. This section is based on the work that appeared in [70, 71].

2.2.1 Lower Bound

In this section, we improve the lower bound of the k-Opt algorithm using the following theorem.

Theorem 2.2.1 (Lemma 3.6 in [17]). *Suppose there exists a Eulerian unweighted graph $G_{k,n,m}$ with n vertices and m edges, having girth at least $2k$. Then, there is a* Metric TSP *instance with m vertices and a k-optimal tour T such that $\frac{c(T)}{c(T^*)} \geq \frac{m}{2n}$, where T^* is the optimal tour of the instance.*

For the previous lower bound the theorem was applied to regular Eulerian graphs with high girth. Instead, we show that for every graph there is a Eulerian subgraph with similar edge vertex ratio and apply the theorem to the Eulerian subgraphs of dense graphs with high girth to get the new bound. Before we start, we make the following observation.

Lemma 2.2.2. *The approximation ratio of the k-Opt algorithm for* Metric TSP *instances with n vertices is monotonically increasing in n.*

Proof. Given an instance I we can increase the number of vertices of I without decreasing the approximation ratio by constructing an instance I' as follows: Make a copy v' of an arbitrary vertex v and set the costs $c(v, v') := 0$, $c(v', w) := c(v, w) \quad \forall w \neq v$. It is easy to see that I' still satisfies the triangle inequality. To prove that the approximation ratio does not decrease we need to show that the optimal tour of I is at least as long as that of I' and the longest k-optimal tour of I' is at least as long as that of I. To show this, observe that we can transform a tour of I to a tour of I' by visiting v' directly after visiting v and leaving the order of the other vertices unchanged. The transformed tour has the same cost as the old tour. Given the optimal tour of I, the above transformation gives us a tour of I' with the same cost. Thus, the optimal tour of I is at least as long as that of I'.

Let T be a k-optimal tour of I. It remains to show that the transformed tour T' is still k-optimal. Assume that there is an improving k-move, apply it on T' to get T'_2. If the edge $\{v, v'\}$ is contained in T'_2, we can contract the vertices v and v' and deleting the self-loop at v to get a shorter tour of I than T. Observe that this tour arises by performing the same k-move on T, contradicting the k-optimality of T. So assume that $\{v, v'\}$ is not contained in T'_2. When we contract the vertices v and v' from T'_2 we get a connected Eulerian graph T_2, where the degree of v is four and the degree of every other vertex is two. Hence, I contains at least two vertices. Now, start at an arbitrary vertex other than v and traverse the graph on a Eulerian walk. Let $\{a_1, v\}, \{v, a_2\}, \{b_1, v\}, \{v, b_2\}$ be the order the edges incident to v are traversed. Since there are exactly two edges incident to v in T, T contains either at most one edge in $\{a_1, v\}, \{v, a_2\}$ or $\{b_1, v\}, \{v, b_2\}$. W.l.o.g. let T contain at most one edge of $\{a_1, v\}$ and $\{v, a_2\}$. We get a tour of I with less or equal length than T by shortcutting $\{a_1, v\}$ and $\{v, a_2\}$ to $\{a_1, a_2\}$ in T_2. To obtain this tour from T, we deleted and add the same edges as the improving k-move for T' except: Instead of deleting $\{v, v'\}$ we delete the edges $T \cap \{\{a_1, v\}, \{v, a_2\}\}$, which is at most one edge. Instead of adding $\{\{a_1, v\}, \{v, a_2\}\} \backslash T$, which is at least one edge, we add the edge $\{a_1, a_2\}$. Hence, this tour arises from T by performing a k-move, again contradicting the k-optimality of T. Therefore, the longest k-optimal tour of I' is at least as long as that of I. □

Lemma 2.2.3. *For every graph G there exists a Eulerian subgraph G' such that $\frac{|E(G')|}{|V(G')|} \geq \frac{|E(G)|+1}{|V(G)|} - 1$.*

Proof. We construct a new graph by deleting cycles successively from G and adding them to an empty graph G_0 with $V(G_0) = V(G)$ until there are no cycles left. After the deletion of cycles, the remaining graph will be a forest with at most $|V(G)| - 1$ edges. Hence, we added at least $|E(G)| - |V(G)| + 1$ edges to G_0. Let $S_1, S_2 \dots, S_u$ be the connected components of G_0. We have

$$\sum_{i=1}^{u} \frac{|V(S_i)|}{|V(G)|} \frac{|E(S_i)|}{|V(S_i)|} = \sum_{i=1}^{u} \frac{|E(S_i)|}{|V(G)|} \geq \frac{|E(G)| - |V(G)| + 1}{|V(G)|} = \frac{|E(G)| + 1}{|V(G)|} - 1.$$

Since $\sum_{i=1}^{u} \frac{|V(S_i)|}{|V(G)|} = 1$, there has to be a connected component S_i with $\frac{|E(S_i)|}{|V(S_i)|} \geq \frac{|E(G)|+1}{|V(G)|} - 1$. Moreover, by construction, S_i is Eulerian and we can set $G' := S_i$. $\square$

Theorem 2.2.4. *The approximation ratio of k-Opt is $\Omega\left(\frac{n}{\mathrm{ex}^{-1}(n,2k)}\right)$ for* Metric TSP *where n is the number of vertices.*

Proof. Take a graph G with girth $2k$, $ex^{-1}(n, 2k)$ vertices and n edges. By Lemma 2.2.3 there is a Eulerian subgraph G' with $\frac{|E(G')|}{|V(G')|} \geq \frac{n+1}{\mathrm{ex}^{-1}(n,2k)} - 1$. Clearly, this subgraph has girth at least $2k$. By Theorem 2.2.1 we can construct an instance with $|E(G')| \leq n$ vertices and an approximation ratio of $\Omega\left(\frac{n+1}{\mathrm{ex}^{-1}(n,2k)} - 1\right) = \Omega\left(\frac{n}{\mathrm{ex}^{-1}(n,2k)}\right)$ since by Theorem 1.4.9 $\lim_{n\to\infty} \frac{n}{\mathrm{ex}^{-1}(n,2k)} = \infty$. The statement follows from the fact that the approximation ratio is monotonically increasing by Lemma 2.2.2. $\square$

Theorem 2.2.5. *If $\mathrm{ex}(n, 2k) \in \Omega(n^c)$ for some $c > 0$, then the approximation ratio of k-Opt is $\Omega(n^{1-\frac{1}{c}})$ for* Metric TSP *where n is the number of vertices.*

Proof. If $\mathrm{ex}(n, 2k) \in \Omega(n^c)$, then $\mathrm{ex}^{-1}(n, 2k) \in O(n^{\frac{1}{c}})$ and by Theorem 2.2.4 we can construct an instance with approximation ratio $\Omega\left(\frac{n}{n^{\frac{1}{c}}}\right) = \Omega(n^{1-\frac{1}{c}})$. $\square$

Together with Theorem 1.4.9 and 1.4.10, we conclude:

Corollary 2.2.6. *For* Metric TSP *the approximation ratio of k-Opt is $\Omega(\sqrt[k]{n})$ for $k = 3, 4, 6$ and $\Omega(n^{\frac{2}{3k-4+\epsilon}})$ for all other k where $\epsilon = 0$ if k is even and $\epsilon = 1$ if k is odd and n is the number of vertices.*

2.2.2 Outline of Upper Bound

In this subsection we briefly summarize the ideas for the analysis of the upper bound for the Metric TSP given by Theorem 1.2.2.

For a fixed k assume that an instance is given with a k-optimal tour T. We fix an orientation of T and assume w.l.o.g. that the length of the optimal tour is 1. To bound the approximation ratio it is enough to bound the length of T. Our general strategy is to construct an auxiliary graph depending on T and bound its girth. More precisely, we show that if this graph has a short cycle this would imply the existence of an improving k-move contradicting the k-optimality of T. Moreover, the auxiliary graph contains many long edges of T so the bound on its girth also bounds the number of long edges in the tour and hence the approximation ratio.

Let the graph G consist of the vertices of the instance and the edges of T, i.e. $G := (V(K_n), T)$. We first divide the edges of T in length classes such that the lth length class consists of the edges with length between c^{l+1} and c^l for some constant $c < 1$, we call these edges *l-long*. For each $l \in \mathbb{N}_0$ we want to get an upper bound on the number of l-long edges that depends on the number of vertices.

If we performed the complete analysis on G, we would get a bad bound on the number of l-long edges since G contains too many vertices. To strengthen the result we first construct an auxiliary graph containing all l-long edges for some fixed l but fewer vertices and bound the number of l-long edges in that graph: We partition $V(G)$ into classes with help of the optimal tour such that in each class any two vertices have small distance to each other. We contract the vertices in each class to one vertex and delete self-loops to get the multigraph G_1^l. We can partition $V(G)$ in such a way that G_1^l contains all the l-long edges. Note we did not delete parallel edges in G_1^l and hence every edge in G_1^l has a unique preimage in G.

Unfortunately, we cannot directly bound the girth of G_1^l since the existence of a short cycle would not necessarily imply an improving k-move for T. For that we need a property of the cycles in the graph: The common vertex of consecutive edges in any cycle has to be head of both or tail of both edges according to the orientation of T. Therefore, we construct the auxiliary graph G_2^l from G_1^l as follows: We start with G_2^l as a copy of G_1^l and color the vertices of G_2^l red and blue. We only consider l-long edges in G_2^l from a red vertex to a blue vertex according to the orientation of T and delete all other edges. We can show that the coloring can be done in such a way that at least $\frac{1}{4}$ of the l-long edges remain in G_2^l.

We claim that the underlying undirected graph of G_2^l has girth at least $2k$. Note that by construction the graph is bipartite and hence all cycles have even length. Assume that there is a cycle C with $2h < 2k$ edges. We call the preimage of the edges of C in G the *C-edges*. Our aim is to construct a tour T' with the assistance of C that arises from T by an improving k-move.

For every common vertex w of two consecutive edges e_1, e_2 of C in G_2^l we consider the preimage e_1^{-1}, e_2^{-1} of e_1, e_2 in G. Then there have to be endpoints $u \in e_1^{-1}$ and $v \in e_2^{-1}$ such that the images of u and v after the contraction in G_2^l are both w. We will call the edge $\{u, v\}$ a *short edge*. In fact since both endpoints of a short edge are mapped to the same vertex in G_1^l after the contraction and we contracted vertices that have a small distance to each other, they are indeed short. Furthermore, we can show that the total length of all the short edges is shorter than that of any single C-edge. The number of the short edges is equal to the number of C-edges which is $2h$. Now, observe that the cycle C defines an alternating cycle in G in a natural way: Let the preimages of C in G be the tour edges and that of the short edges be the non-tour edges.

To construct a new tour T' from T we start by augmenting the alternating cycle. Afterward, the tour may split into at most $2h$ connected components. A key property is that the coloring of the vertices in G_2^l ensures that every connected component contains at least two short edges. Since there are $2h$ short edges, we know that after the augmentation we actually get at most h connected components. To reconnect and retain the degree condition we add twice a set L of at most $h-1$ different C-edges, i.e. in total at most $2h-2$ edges. In the end we shortcut to the new tour T' in a particular way without decreasing $|T \cap T'|$.

Note that the number of the C-edges in the original tour T is $2h$, thus T' contains at least 2 fewer C-edges than T. The additional short edges that T' contains are cheap, therefore T' is cheaper than T. Moreover, T' arises from T by replacing at most $2h - |L|$ C-edges since we deleted the C-edges and added twice the set L consisting of C-edges. Therefore, we know that T' arises from T by a $2h - |L| \leq 2h$-move. By the k-optimality of T we have $2h > k$ or $2h \geq k+1$. This already gives us a lower bound of $k+1$ for the girth of the graph G_2^l as C contains $2h$ edges.

In the next step we use the previous result to show that there is actually a cheaper tour T' that arises by an $h+1$-move. This implies that $h+1>k$ or $2h \geq 2k$, i.e. the girth of G_2^l is at least $2k$. As we have seen above the number of edges we have to replace to obtain T' from T depends on $|L|$, the number of C-edges T' contains. Therefore, we modify T' iteratively such that the number of C-edges in T' increases by 1 after every iteration while still maintaining the property that T' is cheaper than T. We stop when the number of C-edges in T' is $h-1$ as then T' would arise from T by a $2h-(h-1)=h+1$-move.

To achieve this we start with the constructed tour T' and iteratively perform 2-moves that are not necessarily improving but add one more C-edge to T'. In every iteration we consider C-edges e not in the current tour T'. We can show that there is an edge in $T' \backslash T$ incident to each of the endpoints of e. Let the two edges be f_1 and f_2. We want to replace f_1 and f_2 in T' by e_1 and the edge connecting the endpoints of f_1 and f_2 not incident to e. To ensure the connectivity after the 2-move we need to find edges e such that the corresponding edges f_1, f_2 fulfill the following condition: Either both heads or both tails of f_1 and f_2 have to be endpoints of e. It turns out that we can find such edges e in enough iterations to construct T' with the desired properties.

In the end we notice that a lower bound on the girth of G_2^l gives us an upper bound on the number of edges in G_2^l by previous results on extremal graph theory. This implies an upper bound on the number of l-long edges as G_2^l contains at least $\frac{1}{4}$ of the l-long edges in T. That gives us an upper bound on the length of T and thus also an upper bound on the approximation ratio as we assumed that the optimal tour has length 1.

2.2.3 Upper Bound

In this section we give an upper bound on the approximation ratio of the k-Opt and k-Lin-Kernighan algorithm. We bound the length of any k-optimal or k-Lin-Kernighan optimal tour T compared to the optimal tour. To show the bound we first divide the edges of T in classes such that the length of two edges in the same class differ by at most a constant factor. For each of these classes we construct with the help of the optimal tour a graph containing at least $\frac{1}{4}$ of the edges in the class and show that this graph has a high girth. Thus, we can use results from extremal graph theory to bound the number of edges in the length class.

Fix a $k>2$ and assume that a worst-case instance with n vertices is given. Let T be a k-optimal or k-Lin-Kernighan optimal tour of this instance. We fix an orientation of the optimal tour and T. Moreover, let w.l.o.g. the length of the optimal tour be 1. We divide the edges of T into length classes.

Definition 2.2.7. An edge e is *l-long* if $(\frac{4k-5}{4k-4})^{l+1} < c(e) \leq (\frac{4k-5}{4k-4})^l$. Let $\{q_l\}_{l\in\mathbb{N}_0}$ be the sequence of the number of l-long edges in T.

Note that the shortest path between every pair of vertices has length at most $\frac{1}{2}$ since the optimal tour has length 1. Thus, by the triangle inequality every edge with positive length in T has length at most $\frac{1}{2}$ and is l-long for exactly one l. For every l we want to bound the number of l-long edges. Let us consider from now on a fixed l. In the following we define three auxiliary graphs we need for the analysis and show some useful properties of them. Our general aim is to show that the girth of an auxiliary graph containing many l-long edges is high since otherwise there would exist an improving k-move contradicting the assumption. This would imply a bound on the number of l-long edges depending on the number of vertices.

Definition 2.2.8. We view the optimal tour as a circle with circumference 1. Let the vertices of the instance lie on that circle in the order of the oriented tour where the arc distance of two consecutive vertices is the length of the edge between them. Divide the optimal tour circle into $4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil$ consecutive arcs of length $\frac{1}{4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil}$. Two vertices are called *near* to each other if they lie on the same arc.

Definition 2.2.9. Let the directed graph $G := (V(K_n), T)$ consist of the vertices of the instance and the oriented edges of T (an example is shown in Figure 2.3, the colors of the edges will be explained later). The directed multigraph G_1^l arises from G by contracting all vertices near each other to a vertex and deleting self-loops (Figure 2.4).

Note that G_1^l may contain parallel edges. By construction G_1^l contains fewer vertices than G and we will later show that the definition of near ensures that G_1^l contains all the l-long edges. Thus, an upper bound on the girth of G_1^l would give a better upper bound on the number of l-long edges than a bound on the girth of G. Unfortunately, we can not bound the girth of G_1^l since the existence of a short cycle in G_1^l would not necessarily lead to an improving k-move. For that we need the property that the common vertex of consecutive edges of a cycle in the graph is head of both or tail of both edges according to the orientation of T. To ensure this, we further modify in the next step G_1^l to the graph G_2^l.

Lemma 2.2.10. *There exists a coloring of the vertices of G_1^l with two colors such that at least $\frac{1}{4}$ of the l-long edges in G_1^l go from a red vertex to a blue vertex according to the fixed orientation of T.*

Proof. The proof is similar to the standard proof that a maximal cut of a graph contains at least $\frac{1}{2}$ of the edges (see for example Theorem 5.3 in [66]).

By coloring the vertices uniformly at random each l-long edge has a probability of $\frac{1}{4}$ of going from a red vertex to a blue vertex. Hence, the expected number of l-long edges satisfying this condition is $\frac{1}{4}$ of the original number. This implies that there is a coloring where at least $\frac{1}{4}$ of the l-long edges satisfies the condition. □

Definition 2.2.11. We obtain the directed multigraph G_2^l by coloring the vertices of G_1^l red and blue according to Lemma 2.2.10 and deleting all edges that are not l-long edges from a red vertex to a blue vertex according to the fixed orientation of T (Figure 2.5, the colors of the edges will be explained later).

Now, we claim that the underlying undirected graph of G_2^l has girth at least $2k$. In particular, it is a simple graph. Assume the contrary, then there has to be a cycle C with $2h < 2k$ edges since G_2^l is bipartite by construction. We call the preimages in G of the edges in C the *C-edges*. Note that the preimages are unique since we do not delete parallel edges after the contraction.

Definition 2.2.12. Let the *connecting paths* be the connected components of the graph $(V(K_n), T\backslash C)$, i.e. the paths in T between consecutive heads and tails of C-edges (the colored edges in Figure 2.3 and 2.5). Define *head* and *tail* of a path p as the head of the last edge and the tail of the first edge of p according to the orientation of T, respectively. The head and tail of a connecting path are also called the *endpoints* of the connecting path.

Note that the number of connecting paths is equal to that of C-edges which is $2h$.

Lemma 2.2.13. *The two endpoints of a connected path are not near each other. In particular, every connected path contains at least one edge.*

Proof. Observe that the head and tail of a connecting path is a tail and head of a C-edge, respectively. Hence, the corresponding vertices of the heads and tails of the connecting paths in G_2^l are colored red and blue, respectively. Therefore, the two endpoints are not near each other. Since the relation near is reflexive, we can conclude that every connecting path contains at least one edge. □

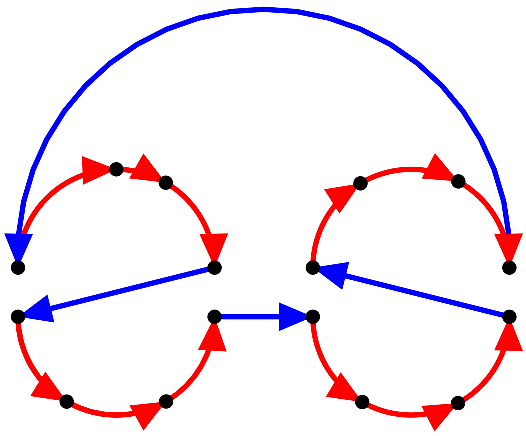


Figure 2.3: An example instance with a k-optimal tour, i.e. the directed graph G. The blue and red edges are the C-edges and connecting path edges that arise from the chosen cycle in G_2^l in Figure 2.5, respectively.

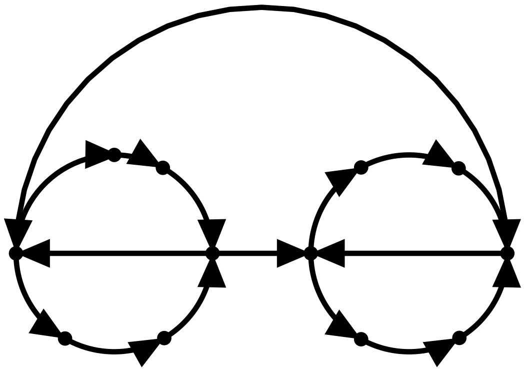


Figure 2.4: The directed multigraph G_1^l: We contracted vertices that lie near each other in the optimal tour. Note that the optimal tour is not drawn here, so it is not clear from the figure which vertices to contract.

Definition 2.2.14. For any two endpoints v_1, v_2 of C-edges in G which are near each other we call the edge $\{v_1, v_2\}$ a *short edge*.

Lemma 2.2.15. *There are exactly $2h$ short edges forming an alternating cycle with the C-edges. Moreover, every short edge connects either two heads or two tails of connecting paths.*

Proof. For any endpoint of a C-edge in G there is exactly one other endpoint of a C-edge which is near to it since the C-edges in G are the preimage of a cycle in G_2^l. By definition every near pair of such endpoints is connected by a short edge and no other short edges exist. Note that there are $2h$ C-edges, so we get $2h$ short edges which form a set of alternating cycles with the C-edges. Using the fact again that after the contraction we get a single cycle C in G_2^l we see that the C-edges form with the short edges a single alternating cycle. Since the vertices of C are colored either red or blue in G_2^l, the short edges connect two heads or two tails of C-edges and hence also two heads or two tails of connecting paths. □

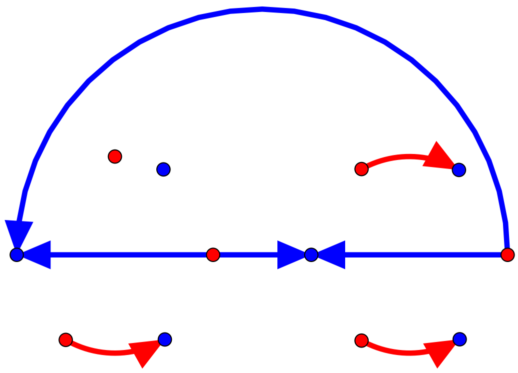


Figure 2.5: The directed multigraph G_2^l: Coloring the vertices and only considering the l-long edges from red to blue. In this example the upper left edge is not l-long and hence not drawn. The blue edges form the undirected cycle C, the red edges are the remaining edges of the connecting paths corresponding to this cycle.

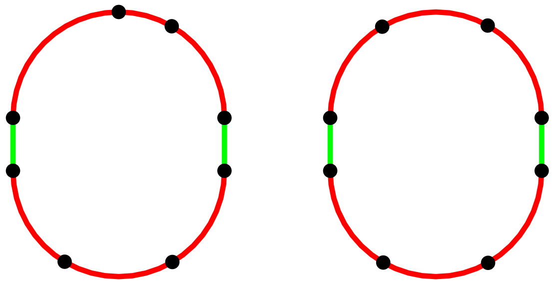


Figure 2.6: The graph $G_3^{l,C}$: The green edges are the short edges, the red edges are the connecting paths.

Definition 2.2.16. We construct the graph $G_3^{l,C}$ as follows: The vertex set of $G_3^{l,C}$ is that of G and the edge set consists of the connecting paths and the short edges (Figure 2.6).

Lemma 2.2.17. *$E(G_3^{l,C})$ is the union of at most h disjoint cycles.*

Proof. By the definition of connecting path every endpoint of a connecting path is an endpoint of a C-edge and vice versa. By Lemma 2.2.15 every endpoint of a C-edge is an endpoint of a short edge and vice versa. Hence, every vertex in G is either an endpoint of a connecting path and a short edge or none of them. Thus, the edges of $G_3^{l,C}$ form disjoint cycles. Note that every connected component in $G_3^{l,C}$ contains at least two connecting paths since the two endpoints of a connecting path are not near each other by Lemma 2.2.13 and hence they cannot be connected by a short edge. Thus, there are at most h connected components. □

Before we start with the actual analysis we show that the total length of all short edges is smaller than that of any C-edge.

Lemma 2.2.18. *Let S be the set of short edges. We have*

$$\sum_{e \in S} c(e) \leq \frac{1}{2}\left(\frac{4k-5}{4k-4}\right)^l.$$

Proof. By Lemma 2.2.15 there are $2h \leq 2(k-1)$ short edges. Each of them connects two vertices which are near each other. By the triangle inequality, each of the short edges has length at most $\frac{1}{4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil}$. Hence the total length of short edges is at most $2h\frac{1}{4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil} \leq 2(k-1)\frac{1}{4(k-1)(\frac{4k-4}{4k-5})^l} = \frac{1}{2}(\frac{4k-5}{4k-4})^l$. □

Lemma 2.2.19. *Let S be the set of short edges, B_1 and B_2 be sets of C-edges with $|B_1| < |B_2| \leq 2h$, then*

$$\sum_{e\in B_1} c(e) + \sum_{e\in S} c(e) \leq \sum_{e\in B_2} c(e).$$

Proof. Let $l_1 := |B_1|,\ l_2 := |B_2|$. We have

$$\frac{l_1 + \frac{1}{2}}{l_1 + 1} = \frac{2l_1 + 1}{2l_1 + 2} \leq \frac{2\cdot(2h-1)+1}{2\cdot(2h-1)+2} \leq \frac{4h-1}{4h} \leq \frac{4(k-1)-1}{4(k-1)} = \frac{4k-5}{4k-4}.$$

Combined with Lemma 2.2.18 and the fact that C-edges are edges of G_2^l and hence l-long we get

$$\begin{aligned}\sum_{e\in B_1} c(e) + \sum_{e\in S} c(e) &\leq l_1\left(\frac{4k-5}{4k-4}\right)^l + \frac{1}{2}\left(\frac{4k-5}{4k-4}\right)^l \\ &= \left(l_1+\frac{1}{2}\right)\left(\frac{4k-5}{4k-4}\right)^l \leq (l_1+1)\left(\frac{4k-5}{4k-4}\right)^{l+1} \\ &\leq l_2\left(\frac{4k-5}{4k-4}\right)^{l+1} < \sum_{e\in B_2} c(e).\end{aligned}$$ □

Now, we show that the existence of C implies that there is an improving k-move or improving alternating cycle of length at most $2k$ contradicting the k-optimality or k-Lin-Kernighan optimality of T.

Lemma 2.2.20. *There is a tour T' containing the connecting paths, $u-1$ C-edges and at least $2h-2u+2$ short edges, where u is the number of connected components of $G_3^{l,C}$. Moreover, $T \triangle T'$ is an alternating cycle of T.*

Proof. We construct such a tour T'. Start with a graph G' with the same vertex set and edge set as $G_3^{l,C}$. First, add a set of C-edges to $E(G')$ that makes the graph connected. This is possible since T consists of the C-edges and connecting paths and is connected. We call these C-edges the *fixed C-edges*. Next, add another copy of the fixed C-edges (they do not belong to the fixed C-edges, Figure 2.7). We will call the connecting path edges and the fixed C-edges the *fixed* edges. Since every vertex of G' is incident to at least one connecting path edge, G' is by construction connected.

We can decompose $E(G')$ into cycles: for every connected component in $G_3^{l,C}$ we get a cycle by Lemma 2.2.17 and for every fixed C-edge and the copy of it a cycle with two edges. Moreover, every vertex b with degree greater two has degree four and is the intersection point of two cycles C_1 and C_2. Note that there are two fixed edges incident to b, a connecting path edge and a fixed C-edge, one lying on C_1 and the other on C_2. This implies that there are also two non-fixed edges incident to b, one lying on C_1 and the other on C_2, we call this property the *transverse* property. Now, we can iteratively shortcut $E(G')$ to a tour: In every step we shortcut two cycles intersecting at vertex b to one cycle by shortcutting the two non-fixed edges $\{a, b\}$ and $\{b, c\}$ to $\{a, c\}$ and

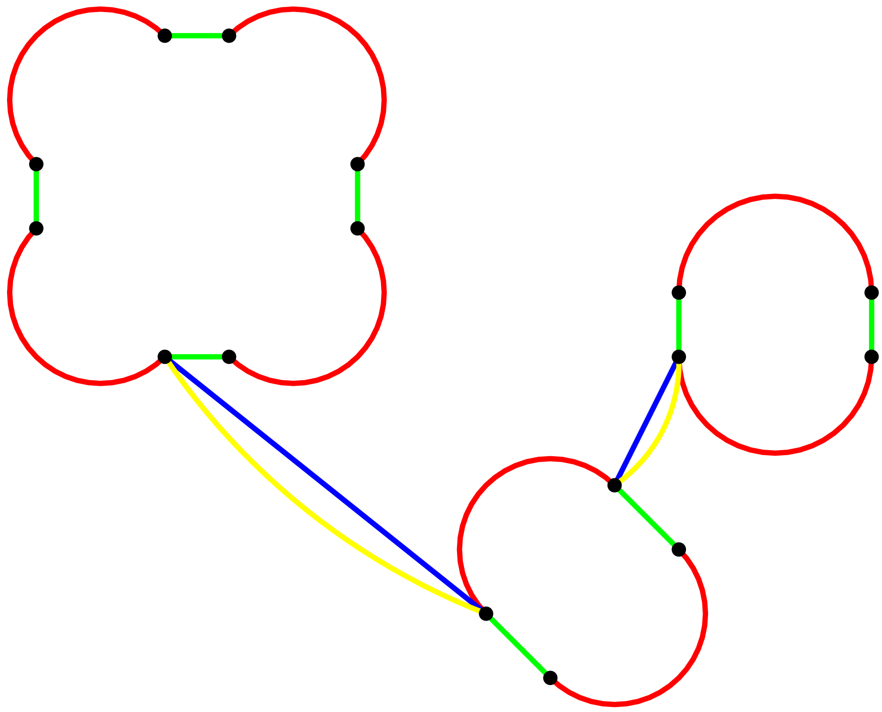


Figure 2.7: Sketch for Lemma 2.2.20. The red curves represent the connecting paths. The green edges are the short edges, the blue edges are the fixed C-edges and the yellow edges are the copies of the fixed C-edges. The tour T' results from shortcutting the green and yellow edges while leaving the other edges fixed.

decrease the number of vertices with degree greater two. Note that each shortcut does not affect the transverse property at other intersection points. When this procedure is not possible anymore, every vertex has degree two and since G' was connected we get a tour that contains all the fixed edges.

The final tour contains by construction $u-1$ fixed C-edges. Moreover, to connect the $2h$ connecting paths T' contains besides of the $u-1$ fixed C-edges and shortcuts of the $u-1$ copies of them also $2h-2u+2$ short edges.

It remains to prove that $T \triangle T'$ is an alternating cycle. By Lemma 2.2.15 we know that $T \triangle E(G_3^{l,C})$ is an alternating cycle. Note that the C-edges are the tour edges in this cycle. By adding two copies of the fixed C-edges we need to add a copy of them instead of deleting a copy of them to obtain $E(G')$. Hence, we change the fixed C-edges from tour to non-tour edges in the alternating cycle. After that, the alternating cycle does not have to be alternating anymore. When we shortcut the Eulerian walk to a tour we shortcut short edges and copies of fixed C-edges, thus consecutive non-tour edges to a non-tour edge. Assume that after the shortcutting step there are still two consecutive non-tour edges. Consider the common vertex v of these two edges. Note that every vertex of the alternating cycle is an endpoint of a connecting path that is not contained as a tour edge in the cycle. Hence the degree of v in T' has to be at least three, contradicting the definition of tour. Therefore, in the end the cycle is alternating again. $\square$

Remark 2.2.21. The last lemma already gives us a bound on the girth of G_2^l: The length of T' can be bounded by the length of the connecting paths plus $2(u-1) < 2h$ C-edges and all short edges. Thus, by Lemma 2.2.19 T' is shorter than T. The alternating cycle $T \triangle T'$ consists of $2h-(u-1) \leq 2h$ tour edges, which are the C-edges we remove. If $2h \leq k$, this would contradict the k-optimality or k-Lin-Kernighan optimality of T, hence G_2^l has girth at least $k+1$.

Next, we use T' to get an improved result: G_2^l has girth at least $2k$.

Definition 2.2.22. Given a tour T' containing the connecting paths. An *ambivalent 2-move* replaces two non-connecting path edges of T' to obtain a new tour containing at least one more C-edge.

Definition 2.2.23. Fix an orientation of T', we call a connecting path p *wrongly oriented* if the orientation of p in T' is opposite to the orientation in T. Otherwise, it is called *correctly oriented.*

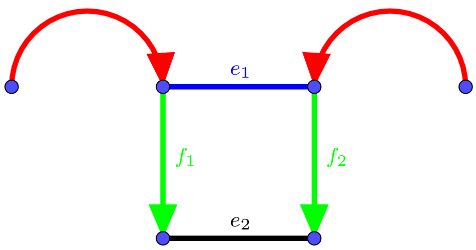


Figure 2.8: Sketch for Lemma 2.2.24. The drawn orientation is that of T'. The red curves represent oppositely oriented connecting paths connected by a C-edge e_1. The green edges f_1 and f_2 are the non-connecting path edges of T' incident to e_1. The edge e_2 connects the other two endpoints of f_1 and f_2 not incident to e_1.

Lemma 2.2.24. *If a tour T' contains a short edge and all connecting paths, then there is an ambivalent 2-move that increases the length of the tour by at most two C-edges.*

Proof. By Lemma 2.2.15 every short edge e always connects either two heads or two tails of connecting paths. If in addition $e \in T'$, one of them is correctly oriented and the other one is wrongly oriented. Thus, as long as there is a short edge in T', there has to be at least one correctly oriented and one wrongly oriented connecting path. In this case there has to be a C-edge e_1 connecting two oppositely oriented connecting paths since the C-edges connect the connecting paths to the tour T. By definition every C-edge connects a head and a tail of two connecting paths. If $e_1 \in T'$, the incident connecting paths would be both correctly or both wrongly oriented. Thus, e_1 is not contained in T'. Let the two non-connecting path edges in T' that share an endpoint with e_1 be f_1 and f_2. Since the two connecting paths are oppositely oriented, either both tails of f_1 and f_2 according to the orientation of T' are endpoints of e_1 or both heads. Assume w.l.o.g. that they share their tails with e_1, let e_2 be the edge connecting the heads of f_1 and f_2 (Figure 2.8). Now, we can make a 2-move replacing f_1, f_2 by e_1 and e_2 to obtain a new tour with the additional C-edge e_1. The tour stays connected since e_1 and e_2 connect the tails and heads of f_1 and f_2, respectively. By Lemma 2.2.13 every connecting path contains at least one edge, hence there are no two adjacent C-edges. Thus, f_1 and f_2 are not C-edges and the new tour contains at least one more C-edge.

Moreover, by the triangle inequality we have $c(e_2) \le c(f_1) + c(e_1) + c(f_2)$ and thus each of the 2-moves increases the length of the tour by at most two C-edges. $\square$

Lemma 2.2.25. *The given tour vertex T is not $h+1$-optimal and $h+1$-Lin-Kernighan optimal.*

Proof. Let u be the number of connected components of $G_3^{l,C}$. By Lemma 2.2.20, we can construct a tour T' using the connecting paths, $u-1$ C-edges and $2h-2u+2$ short edges. We modify T' iteratively. In every iteration fix an orientation of T'. If T' contains a short edge, we perform an ambivalent 2-move by Lemma 2.2.24. Note that there are $2h-2u+2$ short edges in the beginning and with each of these 2-moves, we replaced at most two short edges. Therefore, we can perform by Lemma 2.2.17 $h-u \ge 0$ iterations. After that, we get a tour with $h-1$ C-edges and all connecting paths. Thus, the resulting tour arises by an $h+1$-move from T. In the beginning the length of T' can be bounded by the length of the connecting paths, $2(u-1)$ C-edges and copies of C-edges and the short edges. In every iteration the cost increases by at most two C-edges. Hence, in the end the cost of T' is bounded from above by the cost of the connecting paths, $2h-2$ C-edges and the cost of the short edges. By Lemma 2.2.19 T' is shorter than T which contains $2h$ C-edges.

It remains to show that the $h+1$-move can be performed by augmenting a closed alternating walk. We prove by induction over the iterations that $T \triangle T'$ is an alternating cycle. In the beginning, by Lemma 2.2.20 $T \triangle T'$ is an alternating cycle of T. Assume that $T \triangle T'$ is an alternating cycle in the beginning of an iteration. Let f_1 and f_2 be replaced by e_1 and e_2 during the iteration, where e_1 is a C-edge. Note that f_1, e_1 and f_2 share endpoints on the alternating cycle. Moreover, a cycle visits every vertex by definition at most once, hence f_1, e_1 and f_2 are consecutive edges of it. With the 2-move we shortcut the three consecutive non-tour, tour and non-tour edges of the cycle by the non-tour edge e_2, hence it remains an alternating cycle. This completes the proof. $\square$

Since $h < k$, this is a contradiction to the assumption that T is k-optimal or k-Lin-Kernighan optimal. Hence, such a cycle C with less than $2k$ edges cannot exist and this gives us a bound on the number of l-long edges:

Corollary 2.2.26. *We have $q_l \le 4\,\mathrm{ex}(4(k-1)\lceil (\frac{4k-4}{4k-5})^l \rceil, 2k)$ where q_l is the number of l-long edges in T.*

Proof. By definition, G contains q_l l-long edges. By the triangle inequality, any two vertices which are near each other have distance at most $\frac{1}{4(k-1)\lceil (\frac{4k-4}{4k-5})^l \rceil} \le \frac{1}{4(k-1)(\frac{4k-4}{4k-5})^l} = \frac{(\frac{4k-5}{4k-4})^l}{4(k-1)}$ which is shorter than the length of any l-long edge. Hence, G_1^l has also q_l l-long edges. Since we have chosen a coloring according to Lemma 2.2.10, G_2^l has at least $\frac{1}{4} q_l$ edges. By the k-optimality or k-Lin-Kernighan optimality and Lemma 2.2.25, G_2^l has girth at least $2k$ and thus at most $\mathrm{ex}(|V(G_2^l)|, 2k) \le \mathrm{ex}(4(k-1)\lceil (\frac{4k-4}{4k-5})^l \rceil, 2k)$ edges. Therefore, $q_l \le 4\,\mathrm{ex}(4(k-1)\lceil (\frac{4k-4}{4k-5})^l \rceil, 2k)$. $\square$

Lemma 2.2.27. *If the number of l-long edges $q_l \le f(l)$ for some function f and all l and $l^* := \min\{j \in \mathbb{N} \mid \sum_{l=0}^{j} f(l) \ge n\}$, then*

$$c(T) \le \sum_{l=0}^{l^*} f(l) \left(\frac{4k-5}{4k-4} \right)^l .$$

Proof. By the definition of l-long edges, we have

$$c(T) \leq \sum_{l=0}^{\infty} q_l \left(\frac{4k-5}{4k-4}\right)^l.$$

Since every edge with positive cost is l-long for some l, we have $\sum_{l=0}^{\infty} q_l \leq n$. Moreover, $(\frac{4k-5}{4k-4})^l$ is monotonically decreasing in l, hence the right hand side is maximized if q_l is maximal for small l. Thus, we get an upper bound by assuming that $q_l = f(l)$ for $l \leq l^*$ and $q_l = 0$ for $l > l^*$, where $l^* := \min\{j \in \mathbb{N} | \sum_{l=0}^{j} f(l) \geq n\}$. $\square$

Corollary 2.2.28. *For* $l^* := \min\{j \in \mathbb{N} | \sum_{l=0}^{j} 4\,\mathrm{ex}(4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil, 2k) \geq n\}$ *we have*

$$c(T) \leq \sum_{l=0}^{l^*} \frac{4\,\mathrm{ex}(4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil, 2k)}{(\frac{4k-4}{4k-5})^l}.$$

Proof. By Corollary 2.2.26 and Lemma 2.2.27, we get an upper bound by assuming $q_l = 4\,\mathrm{ex}(4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil, 2k)$ for $l \leq l^*$ and $q_l = 0$ else. $\square$

Theorem 2.2.29. *If* $\mathrm{ex}(x, 2k) \in O(x^c)$ *for some* $c > 1$*, the approximation ratio of the* k*-Opt and* k*-Lin-Kernighan algorithm are* $O(n^{1-\frac{1}{c}})$ *for* Metric TSP *where* n *is the number of vertices.*

Proof. Let d be a constant such that $\mathrm{ex}(x, 2k) \leq dx^c$. By Corollary 2.2.26, we have $q_l \leq 4\,\mathrm{ex}(4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil, 2k) \leq 4d\left(8(k-1)\left(\frac{4k-4}{4k-5}\right)^l\right)^c$. By Lemma 2.2.27, we get an upper bound by assuming that $q_l = 4d\left(8(k-1)\left(\frac{4k-4}{4k-5}\right)^l\right)^c$ for $l \leq l^*$ and $q_l = 0$ for $l > l^*$, where $l^* := \min\{j \in \mathbb{N} | \sum_{l=0}^{j} 4d\left(8(k-1)\left(\frac{4k-4}{4k-5}\right)^l\right)^c \geq n\}$. Hence

$$\begin{aligned}
c(T) &\leq \sum_{l=0}^{l^*} q_l \left(\frac{4k-5}{4k-4}\right)^l \leq \sum_{l=0}^{l^*} \frac{4d\left(8(k-1)\left(\frac{4k-4}{4k-5}\right)^l\right)^c}{(\frac{4k-4}{4k-5})^l} \\
&\leq 4d\,(8(k-1))^c \sum_{l=0}^{l^*} \left(\frac{4k-4}{4k-5}\right)^{(c-1)l} = 4d\,(8(k-1))^c \frac{\left(\frac{4k-4}{4k-5}\right)^{(c-1)(l^*+1)} - 1}{\left(\frac{4k-4}{4k-5}\right)^{(c-1)} - 1}.
\end{aligned}$$

By definition,

$$\sum_{l=0}^{l^*-1} q_l = \sum_{l=0}^{l^*-1} 4d\left(8(k-1)\left(\frac{4k-4}{4k-5}\right)^l\right)^c = 4d\,(8(k-1))^c \frac{\left(\frac{4k-4}{4k-5}\right)^{cl^*} - 1}{\left(\frac{4k-4}{4k-5}\right)^c - 1} < n.$$

Thus, $\left(\frac{4k-4}{4k-5}\right)^{cl^*} < \frac{\left(\left(\frac{4k-4}{4k-5}\right)^c-1\right)n}{4d(8(k-1))^c} + 1$ and we get

$$c(T) \leq 4d\,(8(k-1))^c \, \frac{\left(\frac{4k-4}{4k-5}\right)^{(c-1)(l^*+1)} - 1}{\left(\frac{4k-4}{4k-5}\right)^{(c-1)} - 1}$$

$$= \frac{4d\,(8(k-1))^c}{\left(\frac{4k-4}{4k-5}\right)^{(c-1)} - 1} \left(\left(\frac{4k-4}{4k-5}\right)^{(c-1)} \left(\left(\frac{4k-4}{4k-5}\right)^{cl^*} \right)^{\frac{c-1}{c}} - 1 \right)$$

$$< \frac{4d\,(8(k-1))^c}{\left(\frac{4k-4}{4k-5}\right)^{(c-1)} - 1} \left(\left(\frac{4k-4}{4k-5}\right)^{(c-1)} \left(\frac{\left(\left(\frac{4k-4}{4k-5}\right)^c - 1\right) n}{4d\,(8(k-1))^c} + 1 \right)^{\frac{c-1}{c}} - 1 \right) \in O(n^{1-\frac{1}{c}}).$$

Since we assumed that the length of the optimal tour is 1, we get the result. □

Combined with Theorem 1.4.8 we conclude:

Corollary 2.2.30. *The approximation ratios of the k-Opt and k-Lin-Kernighan algorithm are $O(\sqrt[k]{n})$ for* Metric TSP *where n is the number of vertices.*

Remark 2.2.31. When we do not consider k as a constant the above analysis gives us an upper bound of $O(k\sqrt[k]{n})$.

2.2.4 Comparing the Lower and Upper Bound

In this section we compare the lower and upper bound we got from the previous sections for the k-Opt algorithm. From Corollary 2.2.6 and Corollary 2.2.30 we can directly conclude that

Theorem 2.2.32. *The approximation ratio of the k-Opt algorithm is $\Theta(\sqrt[k]{n})$ for $k = 3, 4, 6$ where n is the number of vertices.*

Now, we want to compare the bounds for other values of k where the exact behavior of $\text{ex}(n, 2k)$ is still unknown.

Lemma 2.2.33. *For all $x \geq 2$ we have* $\text{ex}(2x, 2k) \leq 6\,\text{ex}(x, 2k)$.

Proof. The proof is similar to the standard proof that the maximal cut of a graph contains at least $\frac{1}{2}$ of the edges (see for example Theorem 5.3 in [66])

Take a graph with $2x$ vertices and $\text{ex}(2x, 2k)$ edges and color randomly half of the vertices in red and the other half in blue. For each edge the probability is $\frac{x-1}{2x-1}$ that the endpoints are colored in the same color. So the expected number of edges which endpoints are colored in the same color is $\frac{x-1}{2x-1}\,\text{ex}(2x, 2k) \geq \frac{1}{3}\,\text{ex}(2x, 2k)$. Hence, it is possible to color them in a way such that $\frac{1}{3}\,\text{ex}(2x, 2k)$ of the edges have endpoints colored in the same color. Note that the subgraphs on the red and blue vertices have girth at least $2k$, hence the total number of edges in both subgraphs is at most $2\,\text{ex}(x, 2k)$. Thus $2\,\text{ex}(x, 2k) \geq \frac{1}{3}\,\text{ex}(2x, 2k)$. □

Lemma 2.2.34. *For real numbers $p_1, \ldots, p_n$ with $0 \leq p_j \leq 1$ for all $j \in \{1, \ldots, n\}$ and $\sum_{j=1}^n p_j = 1$ there exists an instance with n vertices and an approximation ratio of k-Opt of $\Omega\left(\sum_{j=1}^n p_j \frac{j}{\text{ex}^{-1}(j,2k)}\right)$.*

Proof. By Theorem 2.2.4, there exists for any $1 \le j \le n$ an instance I_j with at most n vertices and approximation ratio $\Omega\left(\frac{j}{\mathrm{ex}^{-1}(j,2k)}\right)$. We can extend the number of vertices of these instances to n as described in Lemma 2.2.2. Now, construct a random instance which is equal to I_j with probability p_j for all $j \in \{1,\ldots,n\}$. This instance has the expected approximation ratio of $\Omega\left(\sum_{j=1}^{n} p_j \frac{j}{\mathrm{ex}^{-1}(j,2k)}\right)$. Hence, there is a deterministic instance with an approximation ratio of this value. □

Theorem 2.2.35. *The upper bound from Corollary 2.2.28 on the approximation ratio of k-Opt*

$$\sum_{l=0}^{l^*} \frac{4\,\mathrm{ex}(4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil, 2k)}{(\frac{4k-4}{4k-5})^l},$$

where $l^* := \min\{j \in \mathbb{N} | \sum_{l=0}^{j} 4\,\mathrm{ex}(4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil, 2k) \ge n\}$ *and n is the number of vertices, is tight up to a factor of* $O(\log(n))$.

Proof. By Corollary 2.2.28 and Lemma 2.2.33, we get an upper bound for the approximation ratio of the k-Opt algorithm of

$$\begin{aligned}
&\sum_{l=0}^{l^*} \frac{4\,\mathrm{ex}(4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil, 2k)}{(\frac{4k-4}{4k-5})^l} \\
\le &\sum_{l=0}^{l^*-1} \frac{4\,\mathrm{ex}(4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil, 2k)}{(\frac{4k-4}{4k-5})^l} + \frac{4\,\mathrm{ex}(4(k-1)\lceil(\frac{4k-4}{4k-5})\rceil\lceil(\frac{4k-4}{4k-5})^{l^*-1}\rceil, 2k)}{(\frac{4k-4}{4k-5})^{l^*}} \\
\le &\sum_{l=0}^{l^*-1} \frac{4\,\mathrm{ex}(4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil, 2k)}{(\frac{4k-4}{4k-5})^l} + \frac{24\,\mathrm{ex}(4(k-1)\lceil(\frac{4k-4}{4k-5})^{l^*-1}\rceil, 2k)}{(\frac{4k-4}{4k-5})^{l^*}} \\
\le &28 \sum_{l=0}^{l^*-1} \frac{\mathrm{ex}(4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil, 2k)}{(\frac{4k-4}{4k-5})^l}.
\end{aligned}$$

By the definition of l^*, we have $\mathrm{ex}(4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil) < n$ for all $l < l^*$. Hence, by Lemma 2.2.34 we get a lower bound of

$$\Omega\left(\frac{1}{l^*-1} \sum_{l=0}^{l^*-1} \frac{\mathrm{ex}(4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil, 2k)}{4(k-1)\lceil(\frac{4k-4}{4k-5})^l\rceil}\right).$$

The upper and lower bound differ by a factor of $\Theta(l^*)$. By Lemma 1.4.9, there is a constant C such that

$$\begin{aligned}
n &> \sum_{l=0}^{l^*-1} \mathrm{ex}\left(4(k-1)\left\lceil\left(\frac{4k-4}{4k-5}\right)^l\right\rceil, 2k\right) \ge \sum_{l=0}^{l^*-1} C\left(4(k-1)\left\lceil\left(\frac{4k-4}{4k-5}\right)^l\right\rceil\right)^{1+\frac{2}{3k-5}} \\
&\ge C(4(k-1))^{1+\frac{2}{3k-5}} \sum_{l=0}^{l^*-1}\left(\frac{4k-4}{4k-5}\right)^{l(1+\frac{2}{3k-5})} = C(4(k-1))^{1+\frac{2}{3k-5}} \frac{\left(\frac{4k-4}{4k-5}\right)^{l^*(1+\frac{2}{3k-5})} - 1}{\left(\frac{4k-4}{4k-5}\right)^{(1+\frac{2}{3k-5})} - 1}.
\end{aligned}$$

Thus, $l^* \in \Theta(\log(n))$ and the upper bound is tight up to a factor of $O(\log(n))$. □

2.3 Approximation Ratio of k-Opt for Graph TSP

In this section we investigate the approximation ratio of the k-Opt algorithm for Graph TSP. We show that the k-Opt algorithm has an approximation ratio asymptotically between $\Omega\left(\frac{\log(n)}{\log\log(n)}\right)$ and $O\left(\left(\frac{\log(n)}{\log\log(n)}\right)^{\log_2(9)+\epsilon}\right)$ for all $\epsilon > 0$. This section is based on work appeared in [70, 71].

2.3.1 Lower Bound

In this section we show a lower bound of $\Omega\left(\frac{\log(n)}{\log\log(n)}\right)$ on the approximation ratio of the k-Opt algorithm for Graph TSP. For all positive integers f we first construct an instance with at most $4(2f)^{2kf}$ vertices and a k-optimal tour T with an approximation ratio of at least $\frac{f}{4}$.

For the construction, note that we have $\frac{(2f-1)^{2kf-1}-1}{2f-2} \leq (2f)^{2kf}$, hence by Lemma 1.4.11 there exists a $2f$-regular graph with girth at least $2kf$ and $2(2f)^{2kf}$ vertices. Let G be a connected component of this graph. By construction, we know that G is Eulerian. Now, we construct a k-optimal tour T of a graph similar to G.

Definition 2.3.1. Let $W = (v_0, v_1, \ldots, v_{|E(G)|-1})$ be a Eulerian walk of G. Traverse through G according to W starting at v_0 and mark every fth vertex both in G and in W. Whenever we would mark an already marked vertex v in G, we add a new copy v' of v adjacent exactly to the neighbors of v and mark v' instead. Moreover, we replace this occurrence of v in W by v' and mark v'. Let G' be the graph containing G and all the copies of the vertices we made. After the traversal of W, we mark for every unmarked vertex in G' one arbitrary occurrence of it in W. The tour T consists of the edges connecting consecutive marked vertices in W.

We only need the property that every vertex of G' is marked somewhere in W, hence it does not matter which occurrence we mark in W for the unmarked vertices in G'. Note that the number of edges in W is $f|V(G)|$ since G is $2f$ regular. Hence, we added at most $|V(G)| - 1$ copies of vertices to G to obtain G'. Therefore, we have $|V(G')| < 2|V(G)|$. Next, we show that T is a tour with length $f|V(G)|$ and it is k-optimal to conclude the lower bound on the approximation ratio.

Lemma 2.3.2. *T, as defined in Definition 2.3.1, is a tour of G' with length $f|V(G)|$.*

Proof. By construction, we marked every vertex of G' exactly once. Hence, T visits every vertex of G' exactly once and is a tour. It remains to show that the length of T is $f|V(G)|$. For that, we show that every edge of T has the same length as the shorter of the two walks in W between the two consecutive marked endpoints. This implies the statement since W consists of $|E(G)| = f|V(G)|$ edges. First, note that two consecutive marked vertices of W have distance at most f in G' since we marked every fth vertex at the beginning of the construction and two consecutive vertices of the Eulerian walk are connected by an edge in G'. Now, assume that the distance of two consecutive marked vertices u and v is not equal to the length of the shorter walk between these vertices in W. Then, the walk between u and v in W is not the shortest path between them. Hence, there are at least two distinct walks in G' between u and v that are together shorter than $2f$. Now, transfer the two walks to G by mapping the copies of the vertices to the original vertex. The transferred u-v walk in W uses every edge at most once since W is an Eulerian walk of G. Thus, there has to be an edge of the transferred u-v walk that does not occur in the transferred shortest u-v path, otherwise the transferred shortest path between u and v cannot be shorter. Hence, the union of the two has to contain at least one cycle with length less than $2f$ contradicting the girth of the graph G. □

Lemma 2.3.3. *The tour T, as defined in Definition 2.3.1, is k-optimal.*

Proof. This proof is similar to the proof of Theorem 3.5 in [17].

Assume that there is an improving k-move. Then, this k-move can be decomposed into alternating cycles. Since the k-move is improving, at least one alternating cycle has positive gain. Choose such a cycle C, it consists of at most k tour edges. Note that by construction all tour edges of the cycle have length at most f, so the total length of the tour edges is at most kf. Since C has positive gain, the non-tour edges have a total length of less than kf. Recall that we showed in the proof of Lemma 2.3.2 that the shorter of the two walks in W between consecutive marked vertices is a shortest path between them. Now, consider for all tour edges in C the corresponding walk in W between the endpoints and also call these edges in the walk *tour edges*. Similarly, consider for all non-tour edges the shortest path in G' and also call them *non-tour edges*. The union of these edges gives a closed walk of length less than $2kf$ in G'. We map the closed walk to G by mapping the copies of a vertex to the original vertex. Note that every tour edge occurs at most once in this closed walk since W is a Eulerian walk of G. Thus, there has to be a tour edge that does not occur a second time as a non-tour edge, otherwise the cost of the non-tour edges is not strictly less than that of the tour edges. Hence, the closed walk contains a cycle with length less than $2kf$ contradicting the girth of G. □

Lemma 2.3.4. *For all positive integers f there exists an instance of* Graph TSP *with at most $4(2f)^{2kf}$ vertices and approximation ratio of at least $\frac{f}{4}$ for the k-Opt algorithm.*

Proof. By construction (Definition 2.3.1), G' has at most $2|V(G)| \leq 4(2f)^{2kf}$ vertices and by Lemma 2.3.2 and 2.3.3 T is a k-optimal tour of G' with length $f|V(G)|$. By the double tree algorithm (see for example [45]), we can bound the length of the optimal tour by twice the cost of the minimum spanning tree. In the special case of Graph TSP this is at most $2(|V(G')| - 1) < 2(2|V(G)| - 1) < 4|V(G)|$ since the minimum spanning tree consist only of edges of cost 1. Hence, the approximation ratio is at least $\frac{f}{4}$. □

Lemma 2.3.5. *For all positive integers f and $n \geq 4(2f)^{2kf}$ there exists an instance of* Graph TSP *with n vertices and approximation ratio of at least $\frac{f}{8}$ for the k-Opt algorithm.*

Proof. Let G' and T be constructed as above. For nonnegative integers a, b we construct a graph $G'_{a,b}$ from G'. Choose an arbitrary vertex $v \in V(G')$ and let $G'_1, \ldots, G'_a$ be a copies of G' and $v_1 \ldots, v_a$ be the corresponding vertices of v in these copies. Let $V(G'_{a,b})$ be of the union of the vertices in $V(G'_i)$ and b extra vertices $v_{a+1}, \ldots, v_{a+b}$ and $E(G'_{a,b})$ be the union of the edges in $E(G'_i)$ together with the edges $\{v_i, v_{i+1}\}$ for $i \in \{1, \ldots, a+b-1\}$. We call the edges of the form $\{v_i, v_{i+1}\}$ the *connecting edges*. Consider the tour T for each of the graphs $G'_1, \ldots, G'_a$. Assemble the tours together with two copies of the connecting edges and shortcut to a tour for $G'_{a,b}$. The length of the tour is $a|V(G)|f + 2(a + b - 1)$.

Next, we show that this tour is still k-optimal. For every tour and non-tour edge in the k-move we replace it by the corresponding walk according to W possibly with connecting edges and the shortest path, respectively. We also call these edges *tour* and *non-tour* edges, respectively. The union of these edges is a closed walk. Next, we claim that it is not possible that a connecting edge $\{v_i, v_{i+1}\}$ occurs more often as a tour edge than as a non-tour edge. Assume the contrary, since the edges of the k-move have to cross the cut between v_i and v_{i+1} even times, it has to contain $\{v_i, v_{i+1}\}$ twice as tour edge but not as non-tour edge. But then after performing the k-move the tour is not connected between v_i and v_{i+1} anymore. Hence, the total cost of non-tour connecting edges is less or equal to that of the tour connecting edges. Moreover, the alternating cycle splits into a union of cycles such that each of them only uses edges in one copy G'_j of G' for some $j \in \{1, \ldots, a\}$ or connecting edges. Thus, there is a cycle with positive gain shorter than $2kf$ in some G'_j. We get a contradiction by transforming this cycle to G similar as in the proof of Lemma 2.3.3 contradicting its girth.

We choose a, b such that $a|V(G')| + b = n$, $a \geq 1$ and $0 \leq b < |V(G')|$ since $|V(G')| < 2|V(G)| \leq 4(2f)^{2kf} \leq n$. In this case $G'_{a,b}$ has n vertices and the approximation ratio is at least

$$\frac{a|V(G)|f + 2(a+b-1)}{2(a|V(G')| + b)} > \frac{\frac{1}{2}a|V(G')|f + 2(a+b-1)}{2(a|V(G')| + b)} > \frac{a|V(G')|f}{4(a|V(G')| + |V(G')|)}$$
$$= \frac{af}{4(a+1)} \geq \frac{f}{8}.$$

□

Corollary 2.3.6. *The approximation ratio of k-Opt for* Graph TSP *is* $\Omega\left(\frac{\log(n)}{\log\log(n)}\right)$ *where n is the number of vertices.*

Proof. For all positive integers f and n with $4(2f)^{2kf} \leq n < 4(2(f+1))^{2k(f+1)}$ we get an instance with n vertices and approximation ratio at least $\frac{f}{8}$ by Lemma 2.3.5. Let g^{-1} be the inverse function of $g(x) := 4(2x)^{2kx}$ for $x > 0$. Then, by monotonicity $f \leq g^{-1}(n) < f + 1$. Hence, $f \in \Theta(g^{-1}(n))$. Moreover, we have $g(x) = 4(2x)^{2kx} \leq (2kx)^{2kx} =: g_1(x)$. The inverse function of $g_1(x)$ is $g_1^{-1}(x) = \frac{\log(x)}{2kW(\log(x))}$, where W is the Lambert W function. Thus, $g^{-1}(n) \in \Omega\left(\frac{\log(n)}{2kW(\log(n))}\right) = \Omega\left(\frac{\log(n)}{\log\log(n)}\right)$. Therefore, $f \in \Omega\left(\frac{\log(n)}{\log\log(n)}\right)$ and we have an instance with n vertices and approximation ratio $\Omega\left(\frac{\log(n)}{\log\log(n)}\right)$. □

2.3.2 Outline of Upper Bound

This subsection comprises a sketch of the proof of Theorem 1.2.10. Assume that an instance of Graph TSP (K_n, c) is given where c arises from the unweighted graph G. Let a 2-optimal tour T be given for the instance and fix an orientation.

First, note that every edge with length l corresponds to shortest paths with l edges in G between the endpoints of the edges. Now, if the corresponding shortest paths of two edges share a common directed edge, we see that there is an improving 2-move contradicting the assumed 2-optimality of T (Figure 2.9). Hence, the directed edges of the corresponding shortest paths are disjoint. Note that the optimal tour contains n edges and hence has length at least n. Thus, if the approximation ratio is high, we must have many edges in the union of the shortest paths corresponding to the edges in T and hence also in G. The main challenge now is to exploit this fact in a good way since a simple bound of $n(n-1)$ on the number of directed edges in G would only give an upper bound of $O(n)$ on the approximation ratio, which is worse than the upper bound of $O(\sqrt{n})$ for METRIC TSP.

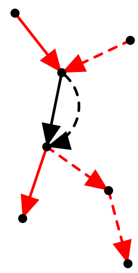


Figure 2.9: The solid and dashed edges are shortest paths that correspond to two edges in T. If they share a directed edge, there exists an improving 2-move replacing these two edges. The cost of the new edges is bounded by the number of the red edges which is less than the total cost of the two original edges.

To get a better result we use the same idea from the analysis of the upper bound for METRIC TSP: We contract vertices and get a graph with fewer vertices and many edges. Instead of contracting once, we iteratively partition the vertices into sets and contract each set to a single vertex to get a new graph. (We note that we actually just contract the vertices and construct the edges of the new graph in a slightly different way. But let us assume for simplicity that the edges of the new graph are images of the contraction of edges in the old graph.) Starting with G in every iteration we ideally want to partition the vertices of the current graph into sets, contract each set to a vertex and delete self-loops such that:

1. The number of vertices decreases much faster than the number of edges.

2. The subgraphs induced by the sets we contract have small diameter.

The first condition ensures that we get a better bound after every iteration. The second condition builds the connection between the approximation ratio and the number of edges in the contracted graph: It ensures that if the shortest paths corresponding to two edges of T share a directed edge in the contracted graph, then they are also not far away in G, so there is an improving 2-move replacing these two edges. This means that a high approximation ratio would imply a high number of edges in the contracted graph.

Unfortunately, it is not easy to ensure both conditions at the same time even if we know that the graph has many edges as the edges are not equally distributed in the graph. There might be many vertices with very small degree. If we contract them while still ensuring that the subgraphs have small diameter, the number of vertices cannot decrease fast enough. Therefore, we consider a subset of vertices we call *active* vertices and only require that the number of active vertices decreases fast. If an active vertex has small degree, we will not contract it and consider it as inactive in future iterations. Initially, all vertices are active and we use the following theorem to find a good partition of the active vertices.

Theorem 2.3.7 (Theorem 6 in [28])**.** *Given $\epsilon > 0$ every graph G on n vertices can be edge partitioned $E = E_0 \cup E_1 \cup \cdots \cup E_l$ such that $|E_0| \leq \epsilon n^2, l \leq 16\epsilon^{-1}$ and for $1 \leq i \leq l$ the diameter of E_i is at most 4.*

In every iteration we apply the theorem to the subgraph induced by the currently active vertices. The vertices only incident to edges in E_0 become inactive after this iteration. For each of the sets $E_1, \dots, E_l$ we contract the vertices incident to an edge in the set to a single vertex. These are the active vertices in the next iteration. By choosing ϵ appropriately, we can ensure that the number of vertices decreases significantly and the number of vertices that become inactive in every iteration is small.

After a fixed number of iterations, we have at least one edge and one active vertex remaining. Since the number of active vertices decreased much faster than the edges, we can conclude that G only contains few edges compared to the number of vertices. This implies a bound on the approximation ratio.

2.3.3 Upper Bound

In this section we show an upper bound of $O\left(\left(\frac{\log(n)}{\log\log(n)}\right)^{\log_2(9)+\epsilon}\right)$ for all $\epsilon > 0$ on the approximation ratio for the 2-Opt algorithm for Graph TSP instances. This implies the same upper bound also for the general k-Opt and Lin-Kernighan algorithm since they also produce 2-optimal tours. To show the bound, we assume that a worst-case instance together with a 2-optimal tour is given and bound the length of the tour compared to the length of the optimal tour. Starting with the given instance we iteratively contract a subset of vertices. We show that the cardinality of a subset of the vertices, the so called *active vertices*, decreases by a factor exponential in the approximation factor after a certain number of iterations. In the end we know that by construction at least one active vertex is remaining. Hence, we can bound the approximation ratio by the number of active vertices at the beginning which is upper bounded by the total number of vertices.

Let an instance (K_n, c) of Graph TSP and a graph $G = (V(K_n), E(G))$ be given such that $c(u, v)$ is the shortest distance between u and v in G. Moreover, let T be a 2-optimal TSP tour of this instance. Fix an orientation of T and define $f := \frac{\sum_{e \in T} c(e)}{n}$. Note that f does not have to be an integer. We may assume that $f > 1$ since otherwise T has length n and is optimal.

Definition 2.3.8. For every edge $(u, v) \in T$ fix a shortest path between u and v in G. We call (u', v') a *subedge* of $(u, v) \in T$ if u' and v' lie on the fixed shortest path between u and v in G and $c(u, u') < c(u, v')$.

Definition 2.3.9. Fix some $0 < \epsilon < 1$ and set $s := n(f^\epsilon - 1) > 0$. Starting with G_0 we construct iteratively directed multigraphs G_{i+1} from G_i:

- Let $V(G_0) := V(G)$, we call all vertices of G_0 *active*. Moreover, let $p_0(v) := v$ for all $v \in V(G)$.
- To construct $E(G_i)$ for all $i \geq 0$ let $l_i := 9^i$. We start with $E(G_i) = \emptyset$. For every subedge (u', v') of $(u, v) \in T$ with $c(u', v') = l_i$ and such that l_i divides $c(u, u')$ we add the edge $(p_i(u'), p_i(v'))$ to G_i (Figure 2.10).
- To construct $V(G_{i+1})$ from G_i consider the underlying undirected graph of G_i and delete parallel edges. We call the resulting graph G'_i. The set of active vertices in G'_i is the same as in G_i.

- Take an edge partition $E_0^i, \dots, E_l^i$ of the subgraph induced by the active vertices in G_i' such that $\epsilon_i := \frac{s}{8n_i^2 2^i}$, $|E_0| \le \epsilon_i n_i^2$, $l \le \frac{16}{\epsilon_i}$ and the diameter of E_j^i is at most 4 for all $j > 0$. By Lemma 2.3.7, such a decomposition exists.
- Define iteratively the sets $V_1^i, \dots, V_l^i$ as follows: $V_j^i := \{v \in V(G_i) | \exists e \in E_j^i, v \in e\} \setminus \left(V_1^i \cup \dots \cup V_{j-1}^i\right)$.
- We contract the vertices in each of the sets V_j^i to a single vertex, which together with the vertices in $V(G_i) \setminus \left(V_1^i \cup \dots \cup V_l^i\right)$ form the vertex set of G_{i+1}.
- We call the contracted vertices of $V_1^i, \dots, V_l^i$ the *active* vertices of G_{i+1}, all other vertices of G_{i+1} are called *inactive*.
- Note that if a vertex is inactive in G_i it is also inactive in G_{i+1}. Let $X_i := V(G_i) \setminus \left(X_1 \cup \dots \cup X_{i-1} \cup V_1^i \cup \dots \cup V_l^i\right)$ be the set of vertices that become inactive the first time in G_{i+1} (Figure 2.11).
- Let $p_{i+1}(v) \in V(G_{i+1})$ for all $v \in V(G)$ be the image of $p_i(v)$ in G_{i+1}.

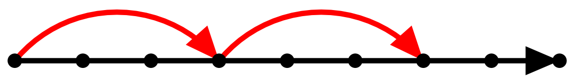


Figure 2.10: Construction of $E(G_i)$: The black edges symbolize the fixed shortest path in G between endpoints of an edge $e \in T$. For illustrative purposes we choose $l_i = 3$. The red edges illustrate the edges we would add between the corresponding endpoints in G_i.

In the following we will show that G_i is a simple directed graph and give a lower bound on the number of edges of G_i depending on the constant ϵ we fixed above.

Lemma 2.3.10. *If $p_i(u) = p_i(v)$, then $c(u, v) < l_i$ for all $u, v \in V(G)$.*

Proof. We prove this statement by induction on i. For $i = 0$ the two vertices u and v have to be identical, hence $c(u, v) = 0 < 1 = l_0$. Now, consider the case $i > 0$. By construction, either $p_{i-1}(u) = p_{i-1}(v)$ or $p_{i-1}(u), p_{i-1}(v) \in V_j^{i-1}$ for some $j > 0$. In the first case we can simply apply the induction hypothesis. In the second case recall that by construction the diameter of E_j^{i-1} is at most 4. Hence, there exists a path of length at most 4 in G_{i-1} connecting $p_{i-1}(u)$ and $p_{i-1}(v)$. W.l.o.g. assume the worst case that the path has length 4. Let $(p_{i-1}(x_j), p_{i-1}(y_j)) \in E(G_{i-1})$ for $j \in \{1, 2, 3, 4\}$ such that $p_{i-1}(y_j) = p_{i-1}(x_{j+1})$ for $j \in \{1, 2, 3\}$, $p_{i-1}(x_1) = p_{i-1}(u)$ and $p_{i-1}(y_4) = p_{i-1}(v)$, i.e. $(p_{i-1}(x_j), p_{i-1}(y_j))$ are the edges of the path (Figure 2.12). We can use the induction hypothesis five times to bound the distance:

$$c(u, v) \le c(u, x_1) + \sum_{j=1}^{4} c(x_j, y_j) + \sum_{j=1}^{3} c(y_j, x_{j+1}) + c(y_4, v) < 9l_i = 9^{i+1} = l_{i+1}$$

$\square$

Lemma 2.3.11. *If there are two subedges (a', b') and (u', v') of different edges (a, b) and (u, v) in T with $c(a', b') + c(u', v') > c(a', u') + c(b', v')$, then T is not 2-optimal.*

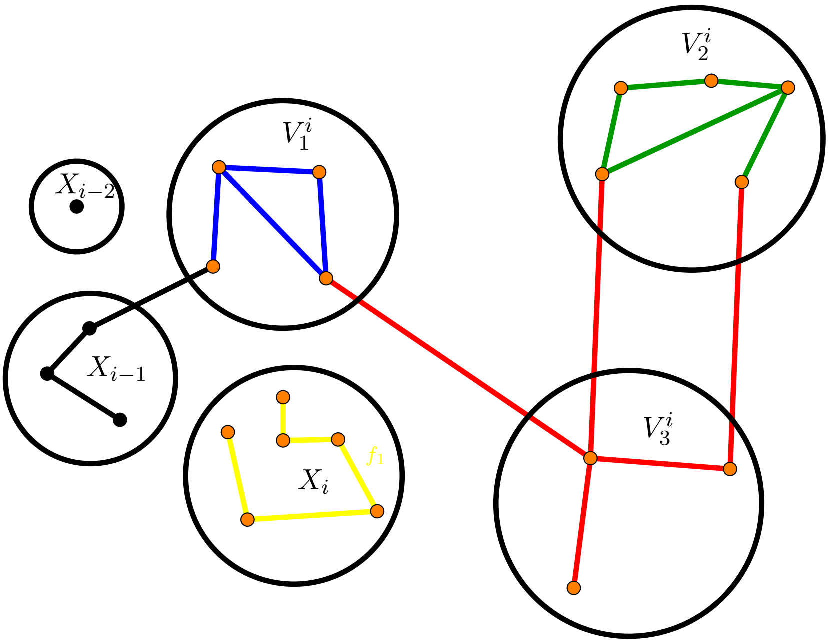


Figure 2.11: Construction of $V(G_{i+1})$: The orange and black vertices are the active and inactive vertices in G_i, respectively. The yellow, blue, green and red edges are the edges of E_0^i, E_1^i, E_2^i and E_3^i, respectively. The black edges have at least an inactive vertex in G_i as endpoint and are hence unassigned. Each of the sets V_1^i, V_2^i and V_3^i will be contracted to a single vertex in G_{i+1}, they will be the active vertices of G_{i+1}.

Proof. We have by the triangle inequality

$$\begin{aligned} c(a,b) + c(u,v) &= c(a,a') + c(a',b') + c(b',b) + c(u,u') + c(u',v') + c(v',v) \\ &> c(a,a') + c(a',u') + c(u',u) + c(b,b') + c(b',v') + c(v',v) \\ &\geq c(a,u) + c(b,v). \end{aligned}$$

Hence, replacing (a,b) and (u,v) by (a,u) and (b,v) is an improving 2-move. $\square$

Lemma 2.3.12. *G_i is a simple directed graph with at least s edges for all $i \leq \log_9(f^{1-\epsilon})$.*

Proof. Assume that there are parallel edges $(p_i(a'), p_i(b'))$ and $(p_i(u'), p_i(v'))$, where $p_i(a') = p_i(u')$ and $p_i(b') = p_i(v')$ for some $a', b', u', v' \in V(G)$. Then, by Lemma 2.3.10 $c(a',u') + c(b',v') < l_i + l_i = c(a',b') + c(u',v')$. If (a',b') and (u',v') are subedges of different edges, there is an improving 2-move by Lemma 2.3.11 contradicting the 2-optimality of T. Otherwise, assume that (a',b') and (u',v') are subedges of an edge $e \in T$. By construction, we can w.l.o.g. assume that a', b', u', v' lie in this order on the fixed shortest path between the endpoints of e according to the orientation of T (with possibly $b' = u'$). We have

$$c(a',v') \leq c(a',u') + c(u',v') \leq c(a',u') + c(b',v') < c(a',b') + c(u',v') \leq c(a',v').$$

Contradiction.

Assume that there is a self-loop $(p_i(u), p_i(u'))$ with $p_i(u) = p_i(u')$ for some $u, u' \in V(G)$. By Lemma 2.3.10, we have $c(u,u') < l_i = c(u,u')$, contradiction.

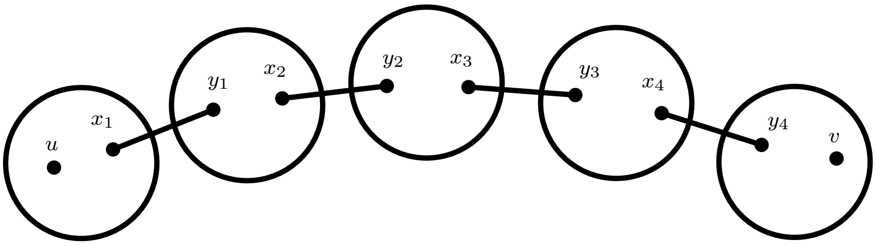


Figure 2.12: Sketch for the proof ofnumber Lemma 2.3.10.

Note that every edge $e \in T$ produces at least $\lfloor \frac{c(e)}{l_i} \rfloor$ edges in G_i. Hence, G_i has in total at least $\sum_{e \in T} \lfloor \frac{c(e)}{l_i} \rfloor \geq \sum_{e \in T} (\frac{c(e)}{l_i} - 1) = n(\frac{f}{l_i} - 1)$ edges. For $i \leq \log_9(f^{1-\epsilon})$ we have $l_i \leq 9^{\log_9(f^{1-\epsilon})} = f^{1-\epsilon}$. Therefore, we have at least $n(\frac{f}{l_i} - 1) \geq n(f^\epsilon - 1) = s$ edges. $\square$

Let n_i be the number of active vertices and m_i be the number of edges where both endpoints are active vertices in G_i. Our next aim is to get a lower bound on m_i and an upper bound on n_i.

Lemma 2.3.13. *We have $m_i \geq \frac{s}{2^i}$ for $i \leq \log_9(f^{1-\epsilon})$.*

Proof. Let $\delta_j(v)$ for $v \in V(G_j)$ be the sum of the indegree and outdegree of v in G_j. Similarly, let $\delta'_j(v)$ for $v \in V(G'_j)$ be the degree of v in G'_j. Since by Lemma 2.3.7 $|E_0^j| \leq \epsilon_j n_j^2 = \frac{s}{8 \cdot 2^j}$, we know that $\sum_{x \in X_j} \delta'_j(x) \leq \frac{s}{4 \cdot 2^j}$. By Lemma 2.3.12, we know that G_j is a simple directed graph, hence we delete at most one parallel edge between every pair of vertices while constructing the graph G'_j. This gives us $\sum_{x \in X_j} \delta_j(x) \leq \frac{s}{2 \cdot 2^j}$ for all $j < i$. Note that vertices $x \in X_j$ will not be contracted in future iterations and hence $x \in V(G_i)$ for all $j < i$. Moreover, l_i is divisible by l_j for all $j < i$. Thus, by construction $\delta_i(x) \leq \delta_j(x)$ for all $x \in X_j$ with $j < i$ and hence $\sum_{x \in X_j} \delta_i(x) \leq \sum_{x \in X_j} \delta_j(x) \leq \frac{s}{2 \cdot 2^j}$. By Lemma 2.3.12, we have $s \leq |E(G_i)| \leq \sum_{j=0}^{i-1} \sum_{x \in X_j} \delta_i(x) + m_i$. Therefore,

$$m_i \geq s - \sum_{j=0}^{i-1} \sum_{x \in X_j} \delta_i(x) \geq s - \sum_{j=0}^{i-1} \frac{s}{2^{j+1}} = \frac{s}{2^i}.$$

$\square$

Lemma 2.3.14. *There is a constant $d > 0$ such that $n_i \leq \frac{n}{(d(f^\epsilon - 1))^{2^i - 1}}$.*

Proof. By Lemma 2.3.7, we can bound the number of active vertices by $n_{i+1} \leq 16 \frac{1}{\epsilon_i} = \frac{16 \cdot 8 \cdot 2^i \cdot n_i^2}{s} = \frac{2^{i+7} n_i^2}{s}$. Now, we show by induction that $n_i \leq \frac{n^{2^i} 2^{2^{i+3} - i - 8}}{s^{2^i - 1}}$. For $n = 0$ we have $n_0 = n = \frac{n^{2^0} 2^{2^3 - 8}}{s^{2^0 - 1}}$. Moreover,

$$n_{i+1} \leq \frac{2^{i+7} n_i^2}{s} \leq \frac{2^{i+7}}{s} \cdot \frac{n^{2^{i+1}} 2^{2^{i+4} - 2i - 16}}{s^{2^{i+1} - 2}} = \frac{n^{2^{i+1}} 2^{2^{i+4} - (i+1) - 8}}{s^{2^{i+1} - 1}}.$$

Hence,

$$n_i \leq \frac{n^{2^i} 2^{2^{i+3} - i - 8}}{s^{2^i - 1}} = \frac{n^{2^i} 2^{2^{i+3} - i - 8}}{(n(f^\epsilon - 1))^{2^i - 1}} \leq \frac{n 2^{2^{i+3} - 8}}{(f^\epsilon - 1)^{2^i - 1}} = \frac{n}{(\frac{1}{2^8}(f^\epsilon - 1))^{2^i - 1}}.$$

$\square$

Lemma 2.3.15. *If $(c_1 f)^{c_2 f^{c_3}} \leq n$ for constants $c_1, c_2, c_3 > 0$, then $f \in O\left(\left(\frac{\log(n)}{\log\log(n)}\right)^{\frac{1}{c_3}}\right)$.*

Proof. We can w.l.o.g. assume that $c_1 \leq 1$ since $c_1 > 1$ reduces to the case $c_1 = 1$ by $f^{c_2 f^{c_3}} < (c_1 f)^{c_2 f^{c_3}} \leq n$. Set $f_1 := (c_1 f)^{c_3}$ and $c_4 := \frac{c_3}{c_2}$. Then, $n \geq (c_1 f)^{c_2 f^{c_3}} = f_1^{\frac{c_2}{c_1^{c_3} c_3} f_1} \geq f_1^{\frac{c_2}{c_3} f_1}$. Hence, $f_1^{f_1} \leq n^{c_4}$. The inverse function of x^x is $\frac{\log(x)}{W(\log(x))}$, where W is the Lambert W function with $W(x) \in \Theta(\log(x))$. Thus

$$f_1 \in O\left(\frac{\log(n^{c_4})}{\log\log(n^{c_4})}\right) = O\left(\frac{\log(n)}{\log(c_4) + \log\log(n)}\right) = O\left(\frac{\log(n)}{\log\log(n)}\right).$$

Therefore, $f = \frac{1}{c_1}(f_1)^{\frac{1}{c_3}} \in O\left(\left(\frac{\log(n)}{\log\log(n)}\right)^{\frac{1}{c_3}}\right)$. □

Theorem 2.3.16. *The approximation ratio of the 2-Opt algorithm for* Graph TSP *is $O\left(\left(\frac{\log(n)}{\log\log(n)}\right)^{\log_2(9)+\epsilon'}\right)$ for all $\epsilon' > 0$ where n is the number of vertices.*

Proof. By the definition of f, we have $\sum_{e \in T} c(e) = nf$. The cost of the optimal tour is at least n since it consists of n edges. Hence, the approximation ratio is at most f and it is enough to get an upper bound on f.

Consider the graph $G_{\lfloor \log_9(f^{1-\epsilon}) \rfloor}$. On the one hand, by Lemma 2.3.13 $m_{\lfloor \log_9(f^{1-\epsilon}) \rfloor} \geq \frac{s}{2^{\lfloor \log_9(f^{1-\epsilon}) \rfloor}} = \frac{n(f^\epsilon - 1)}{2^{\lfloor \log_9(f^{1-\epsilon}) \rfloor}} > 0$ and hence $n_{\lfloor \log_9(f^{1-\epsilon}) \rfloor} \geq 1$. On the other hand, we have by Lemma 2.3.14 $n_{\lfloor \log_9(f^{1-\epsilon}) \rfloor} \leq \frac{n}{(d(f^\epsilon - 1))^{2^{\lfloor \log_9(f^{1-\epsilon}) \rfloor} - 1}}$ for some constant d. Thus, for all $f \geq 2^{\frac{1}{\epsilon}}$ there exists a constant d_1 such that

$$\begin{aligned} n &\geq (d(f^\epsilon - 1))^{2^{\lfloor \log_9(f^{1-\epsilon}) \rfloor} - 1} \geq (d_1 f^\epsilon)^{2^{(1-\epsilon)\log_9(f)} - 2} = (d_1 f^\epsilon)^{2^{\frac{(1-\epsilon)\log_2(f)}{\log_2(9)}} - 2} \\ &= (d_1 f^\epsilon)^{f^{\frac{1-\epsilon}{\log_2(9)} - \frac{2}{\log_2(f)}}}. \end{aligned}$$

For a given $\epsilon' > 0$ we can choose constants ϵ, d_2 such that for all $f \geq d_2$ we have $\frac{1-\epsilon}{\log_2(9)} - \frac{2}{\log_2(f)} \geq \frac{1}{\log_2(9)+\epsilon'}$. By Lemma 2.3.15, we conclude $f \in O\left(\left(\frac{\log(n)}{\log\log(n)}\right)^{\log_2(9)+\epsilon'}\right)$. □

2.4 Approximation Ratio of k-Opt for Euclidean TSP

In this section we modify the instances given in [17] which show an asymptotic lower bound of $\frac{\log(n)}{\log\log(n)}$ for the 2-Opt algorithm. The new instances give the same asymptotic lower bound on the approximation ratio for the k-Opt algorithm as for the 2-Opt algorithm. As a generalization the lower bound not only works for the Euclidean TSP but for all TSP instances where the distances arise from the p-norm for some $p \geq 1$. Together with the known matching upper from [15] this implies that the k-Opt algorithm has an asymptotic approximation ratio of $\Theta\left(\frac{\log(n)}{\log\log(n)}\right)$ for Euclidean TSP. From now on, let us consider the k-Opt algorithm with the p-norm for some fixed k, p with $k \geq 2$ and $p \geq 1$.

For every odd $q \in \mathbb{N}$ we construct an instance I_q with

$$n := 2\sum_{i=0}^{q}(q^{(p+1)i} + 1) + q^{(p+1)q} - 1 + 2\sum_{i=0}^{q-1}(q^{(p+1)(q-i)-1} - 1)$$

vertices (Figure 2.13). Note that $n \in \Theta\left(q^{(p+1)q}\right)$ and hence $q \in \Theta\left(\frac{\log n}{W(n)}\right) = \Theta\left(\frac{\log n}{\log\log n}\right)$ where W is the Lambert W function.

For the construction of I_q first define $q+1$ lines $(l_i)_{i\in\{0,\dots,q\}}$ parallel to the x-axis, where l_i satisfies the function $y = \sum_{s=0}^{i-1} q^{(p+1)(q-s)-1}$. We will call l_i the ith *layer*.

The instance consists of four sets of vertices: V_1, V_2, V_3 and V_4. For V_1 we place $\frac{q^{(p+1)q}}{q^{(p+1)(q-i)}}+1 = q^{(p+1)i}+1$ equidistant vertices on the ith layer l_i between the x-coordinate 0 and $q^{(p+1)q}$ such that the distance between consecutive vertices is $q^{(p+1)(q-i)}$. Note that the coordinates of these vertices are independent of the p-norm since the vertices lie on a line parallel to the x-axis.

The vertices in V_2 are copies of V_1 shifted to the right by $2q^{(p+1)q}$, i.e. every vertex in V_1 with coordinates (e,f) corresponds to a vertex in V_2 with coordinates $(e+2q^{(p+1)q}, f)$.

Now, we fill the gaps in the topmost layer. V_3 consists of $q^{(p+1)q}-1$ vertices dividing the line segment between $(q^{(p+1)q}, \sum_{s=0}^{q-1} q^{(p+1)(q-s)-1})$ and $(2q^{(p+1)q}, \sum_{s=0}^{q-1} q^{(p+1)(q-s)-1})$ into $q^{(p+1)q}$ equidistant parts such that the distance between consecutive vertices is 1.

Define the vertical line segments $(h_i)_{0\le i<q}$ parallel to the y-axis with y-coordinate between $\sum_{s=0}^{i-1} q^{(p+1)(q-s)-1}$ and $\sum_{s=0}^{i} q^{(p+1)(q-s)-1}$ as follows: If i is even, it is the line segment with the x-coordinate 0, otherwise it is the line segment with the x-coordinate $q^{(p+1)q}$.

Similarly, define the shifted vertical line segments $(h'_s)_{0\le s<q}$ parallel to the y-axis with y-coordinate between $\sum_{s=0}^{i-1} q^{(p+1)(q-s)-1}$ and $\sum_{s=0}^{i} q^{(p+1)(q-s)-1}$ as follows: If i is even, it is the line segment with the x-coordinate $3q^{(p+1)q}$, otherwise it is the line segment with the x-coordinate $2q^{(p+1)q}$.

Finally, the set V_4 consists of the following vertices: For each i we place $q^{(p+1)(q-i)-1}-1$ equidistant vertices on h_i and h'_i such that the distance between two consecutive vertices is 1.

The coordinates of the vertices of the instance I_q are given explicitly by:

$$V_1 := \bigcup_{0\le i\le q, 0\le j\le q^{(p+1)i}} (jq^{(p+1)(q-i)}, \sum_{s=0}^{i-1} q^{(p+1)(q-s)-1})$$

$$V_2 := \bigcup_{0\le i\le q, 0\le j\le q^{(p+1)i}} (jq^{(p+1)(q-i)} + 2q^{(p+1)q}, \sum_{s=0}^{i-1} q^{(p+1)(q-s)-1})$$

$$V_3 := \bigcup_{1\le j\le q^{(p+1)q}-1} (q^{(p+1)q}+j, \sum_{s=0}^{q-1} q^{(p+1)(q-s)-1})$$

$$V_4 := \bigcup_{\substack{0\le i\le q-1, i \text{ even},\\ 1\le j\le q^{(p+1)(q-i)-1}-1}} (0, j+\sum_{s=0}^{i-1} q^{(p+1)(q-s)-1}) \cup (3q^{(p+1)q}, j+\sum_{s=0}^{i-1} q^{(p+1)(q-s)-1})$$
$$\bigcup_{\substack{0\le i\le q-1, i \text{ odd},\\ 1\le j\le q^{(p+1)(q-i)-1}-1}} (q^{(p+1)q}, j+\sum_{s=0}^{i-1} q^{(p+1)(q-s)-1}) \cup (2q^{(p+1)q}, j+\sum_{s=0}^{i-1} q^{(p+1)(q-s)-1})$$

Let $V(I_q) := V_1 \cup V_2 \cup V_3 \cup V_4$. Note that $|V_1| = |V_2| = \sum_{i=0}^{q}(q^{(p+1)i}+1), |V_3| =$

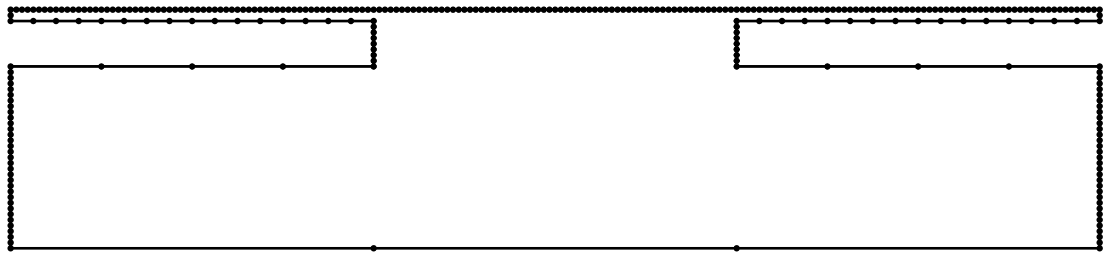


Figure 2.13: The constructed instance I_q for $q = 2, p = 1$ and the tour T. The vertices in V_1, V_2, V_3 lie on the horizontal lines l_0, l_1, l_2 and l_3 where l_0 is the bottommost line. The vertices in V_4 lie on the vertical line segments h_i and h'_i.

$q^{(p+1)q} - 1$ and $|V_4| = 2\sum_{i=0}^{q-1}(q^{(p+1)(q-i)-1} - 1)$. Hence,

$$|V_1| + |V_2| + |V_3| + |V_4| = 2\sum_{i=0}^{q}(q^{(p+1)i} + 1) + q^{(p+1)q} - 1 + 2\sum_{i=0}^{q-1}(q^{(p+1)(q-i)-1} - 1) = n.$$

Define the tour T as shown in Figure 2.13 by connecting consecutive equidistant vertices in V_1, V_2, V_3 and V_4. More formally, define

$$E_1 := \bigcup_{0\le i\le q, 0\le j\le q^{(p+1)i}-1} \{(jq^{(p+1)(q-i)}, \sum_{s=0}^{i-1} q^{(p+1)(q-s)-1}), ((j+1)q^{(p+1)(q-i)}, \sum_{s=0}^{i-1} q^{(p+1)(q-s)-1})\}$$

$$E_2 := \bigcup_{0\le i\le q, 0\le j\le q^{(p+1)i}-1} \{(jq^{(p+1)(q-i)} + 2q^{(p+1)q}, \sum_{s=0}^{i-1} q^{(p+1)(q-s)-1}),$$
$$((j+1)q^{(p+1)(q-i)} + 2q^{(p+1)q}, \sum_{s=0}^{i-1} q^{(p+1)(q-s)-1})\}$$

$$E_3 := \bigcup_{0\le j\le q^{(p+1)q}-1} \{(q^{(p+1)q} + j, \sum_{s=0}^{q-1} q^{(p+1)(q-s)-1}), (q^{(p+1)q} + j + 1, \sum_{s=0}^{q-1} q^{(p+1)(q-s)-1})\}$$

$$E_4 := \bigcup_{\substack{0\le i\le q-1, i \text{ even},\\ 0\le j\le q^{(p+1)(q-i)-1}-1}} \{(0, j + \sum_{s=0}^{i-1} q^{(p+1)(q-s)-1}), (0, j + 1 + \sum_{s=0}^{i-1} q^{(p+1)(q-s)-1})\}$$
$$\cup \{(3q^{(p+1)q}, j + \sum_{s=0}^{i-1} q^{(p+1)(q-s)-1}), (3q^{(p+1)q}, j + 1 + \sum_{s=0}^{i-1} q^{(p+1)(q-s)-1})\}$$
$$\bigcup_{\substack{0\le i\le q-1, i \text{ odd},\\ 0\le j\le q^{(p+1)(q-i)-1}-1}} \{(q^{(p+1)q}, j + \sum_{s=0}^{i-1} q^{(p+1)(q-s)-1}), (q^{(p+1)q}, j + 1 + \sum_{s=0}^{i-1} q^{(p+1)(q-s)-1})\}$$
$$\cup \{(2q^{(p+1)q}, j + \sum_{s=0}^{i-1} q^{(p+1)(q-s)-1}), (2q^{(p+1)q}, j + 1 + \sum_{s=0}^{i-1} q^{(p+1)(q-s)-1})\}$$

$$E_5 := \{(q^{(p+1)q}, 0), (2q^{(p+1)q}, 0)\}$$

Now, let $T := E_1 \cup E_2 \cup E_3 \cup E_4 \cup E_5$. Note that since $q + 1$ is even, T is indeed a tour. Let T^* be an optimal tour of the instance I. Next, we bound the length of T and T^*.

Lemma 2.4.1. *The length of the tour T as defined above for the instance I_q is at least $q \cdot q^{(p+1)q}$.*

Proof. The proof is similar to the proof of Claim 4.5 in [17].
Consider only the horizontal edges connecting consecutive vertices of V_1, i.e. the edge set E_1. On each of the $q+1$ layers these edges form line segments each with length $q^{(p+1)q}$. Hence, we can bound the length of the tour by $(q+1) \cdot q^{(p+1)q} > q \cdot q^{(p+1)q}$. □

Lemma 2.4.2. *The length of the optimal tour T^* for the instance I_q is at most $14q^{(p+1)q}$.*

Proof. The proof is similar to the proof of Claim 4.4 in [17].
We bound the length of the optimal tour by twice the length of a spanning tree. For any vertex in $V_1 \cup V_2$ not on the topmost layer l_q consider the vertical line segment to the next higher layer. Since the distance between consecutive vertices on l_{i+1} is divisible by the distance of vertices on l_i, every vertex not on l_p is connected this way with a vertex on the next higher layer. For all i it is easy to see that h_i and h'_i and hence the vertices of V_4 lie on these vertical line segments. There are $2\left(q^{(p+1)i}+1\right)$ vertices on l_i and each of the connection edges to l_{i+1} has length $q^{(p+1)(q-i)-1}$. Thus, these edges have a total length of $2\sum_{i=0}^{q-1} q^{(p+1)(q-i)-1}\left(q^{(p+1)i}+1\right)$. We get a spanning tree by adding edges connecting consecutive vertices on l_p on which the vertices of V_3 lies. These edges form a line segment with length $3q^{(p+1)q}$. Altogether, the total length of the spanning tree is:

$$3q^{(p+1)q} + 2\sum_{i=0}^{q-1} q^{(p+1)(q-i)-1}\left(q^{(p+1)i}+1\right) = 3q^{(p+1)q} + 2\sum_{i=0}^{q-1}\left(q^{(p+1)q-1} + q^{(p+1)(q-i)-1}\right)$$
$$\leq 3q^{(p+1)q} + 4q^{(p+1)q} = 7q^{(p+1)q}.$$

The length of the optimal tour can now be bounded by twice the cost of this spanning tree. □

Combining both lemmas we can already see that the ratio between the length of T and the optimal tour is at least $\frac{q \cdot q^{(p+1)q}}{14q^{(p+1)q}} = \frac{q}{14}$ and recall that $q \in \Theta\left(\frac{\log(n)}{\log\log(n)}\right)$. It remains to show that the tour T for the instance I_q is k-optimal for q large enough. For that we first show some auxiliary lemmas.

Definition 2.4.3. The *bounding box* of a set of points P is the smallest axis-parallel rectangle containing all points in P.

Lemma 2.4.4. *Let P be a polygon such that the bounding box of P has the side length d_x and d_y in x and y direction, respectively. Then, the perimeter of P is at least $2\sqrt[p]{d_x^p + d_y^p}$ where the distances are induced by the p-norm.*

Proof. For each side of the bounding box mark a vertex of P that lies on that side. Note that a vertex may be marked multiple times for different sides. For every other unmarked vertex we can shortcut the two adjacent edges to get a new polygon without increasing the perimeter and changing the bounding box. In the end we end up with a polygon that has at most 4 sides. We can further assume that the polygon is simple since otherwise we can perform a 2-move to remove the crossing edges without increasing the length of the perimeter and changing the bounding box. Therefore, we may assume that P consists of the vertices a, b, c, d lying on the top, left, right, bottom side of the bounding square, respectively. Note that these vertices may coincide in case that P contains less than four edges. We reflect the vertex d by the left and right side of the bounding box to obtain e and f, respectively. Then, we reflect f by the top side of the bounding box to obtain g (Figure 2.14). By symmetry and the triangle inequality, the perimeter of P is at least.

$$\begin{aligned}
&\|a-b\|_p + \|a-c\|_p + \|b-d\|_p + \|c-d\|_p \\
=&\|a-b\|_p + \|a-c\|_p + \|b-e\|_p + \|c-f\|_p \\
\geq&\|a-e\|_p + \|a-f\|_p = \|a-e\|_p + \|a-g\|_p \geq \|e-g\|_p
\end{aligned}$$

Let a', b', c', d', e', f' and g' be the projections of a, b, c, d, e, f and g to the x-axis, respectively. Again by symmetry we have

$$\begin{aligned}
\|e'-g'\|_p = \|e'-a'\|_p + \|a'-g'\|_p &= \|e'-b'\|_p + \|b'-a'\|_p + \|a'-f'\|_p \\
&= \|d'-b'\|_p + \|b'-a'\|_p + \|a'-c'\|_p + \|c'-f'\|_p \\
&= \|d'-b'\|_p + \|b'-a'\|_p + \|a'-c'\|_p + \|c'-d'\|_p = 2d_x
\end{aligned}$$

Together with a similar calculation with the projections of the vertices to the y-axis we can conclude that the bounding box of $\{e, g\}$ has side length $2d_x$ and $2d_y$. Therefore, we have $\|e-g\|_p = \sqrt[p]{(2d_x)^p + (2d_y)^p} = 2\sqrt[p]{d_x^p + d_y^p}$ which completes the proof. $\square$

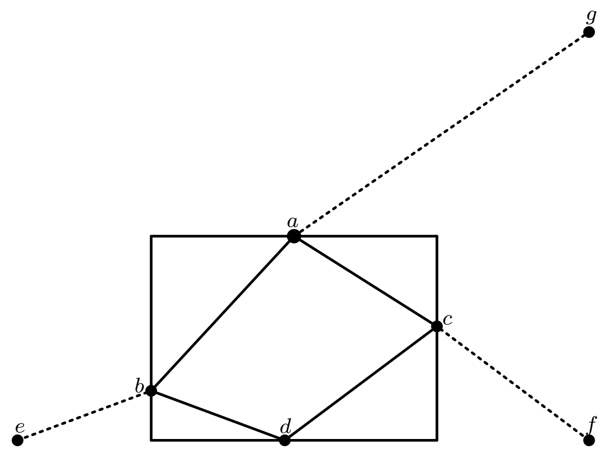


Figure 2.14: Sketch for Lemma 2.4.4

Lemma 2.4.5. *There is a $\bar{q}$ such that for all $q \geq \bar{q}$ and $0 \leq a, b \leq k$ we have*

$$\sqrt[p]{(aq^{(p+1)(q-s)})^p + q^{p((p+1)(q-s)-1)}} > aq^{(p+1)(q-s)} + bq^{(p+1)(q-s-1)}.$$

Proof. We have for q large enough

$$\begin{aligned}
&\left(\sqrt[p]{(aq^{(p+1)(q-s)})^p + q^{p((p+1)(q-s)-1)}}\right)^p - (aq^{(p+1)(q-s)} + bq^{(p+1)(q-s-1)})^p \\
=&(aq^{(p+1)(q-s)})^p + q^{p((p+1)(q-s)-1)} - (aq^{(p+1)(q-s)} + bq^{(p+1)(q-s-1)})^p \\
=&q^{p((p+1)(q-s)-1)} - O(q^{(p+1)(q-s)(p-1)}q^{(p+1)(q-s-1)}) \\
=&q^{p((p+1)(q-s)-1)} - O(q^{(p+1)(q-s)p-(p+1)}) > 0.
\end{aligned}$$

The statement follows from the fact that the power function is monotonically increasing. $\square$

Next, we show that the tour T is indeed k-optimal for q large enough.

Lemma 2.4.6. *There exists a $\bar{q}$ such that for all $q \geq \bar{q}$ the tour T for the constructed instance I_q is k-optimal.*

Proof. Assume that T is not k-optimal. Then there is a closed alternating walk C with at most k tour edges and positive gain. We distinguish two cases:

Case 1: C visits vertices from at least two layers.
Assume that l_s is the layer C visits with the smallest index s. Moreover, let C contain exactly a tour edges with both endpoints lying on l_s. By construction, we can bound the length of the tour edges in C from above by $aq^{(p+1)(q-s)} + (k-a)q^{(p+1)(q-s-1)}$. By assumption, the bounding box of C has side length at least $aq^{(p+1)(q-s)}$ in x-direction. Since C contains at least vertices from two layers, its bounding box has side length at least the distance between l_s and l_{s+1} which is $q^{(p+1)(q-s)-1}$ in y-direction. By Lemma 2.4.4 viewing C as a polygon, the edges of C have total length at least $2\sqrt[p]{(aq^{(p+1)(q-s)})^p + q^{p((p+1)(q-s)-1)}}$. Hence, the length of the non-tour edges is at least $2\sqrt[p]{(aq^{(p+1)(q-s)})^p + q^{p((p+1)(q-s)-1)}} - (aq^{(p+1)(q-s)} + (k-a)q^{(p+1)(q-s-1)})$ and by Lemma 2.4.5 the total gain of C has to be negative for q large enough, contradiction.

Case 2: C visits vertices from at most one layer.
Let $S = \{e_1, \dots, e_l\}$ be the set of edges in C with both endpoints lying on the same layer. Moreover, let the endpoints of e_i have the coordinates (x_i, y_i) and (x'_i, y'_i) and w.l.o.g. assume that $x_i < x'_i$. Since C is a closed walk, it intersects the vertical line $x = c$ for all $c \in \mathbb{R}$ an even number of times. This means that for $x_i < c < x'_i$ the vertical line $x = c$ intersects C in a set of edges J_c with $|J_c| \geq 2$. Furthermore, J_c contains besides of e_i only non-tour edges since otherwise C would visit at last two layers. By construction, J_c contains the same edges for all $x_i < c < x'_i$ since otherwise C would visit more than one layer. Therefore, we can assign every edge e_i in S to an arbitrary non-tour edge f_i in J_c for some $x_i < c < x'_i$. Note that a non-tour edge may be assigned multiple tour edges in S. We mark all edges in S and the corresponding edges they are assigned to.

We claim that the length of the marked non-tour edges is at least as long as the marked tour edges. To see this we assign the length of edges $e_i \in S$ to that part of f_i that the vertical line $x = c$ intersects for values of c satisfying $x_i < c < x'_i$ (Figure 2.15). We call that part of the edge f'_i. Note that f'_i and f'_j are disjoint for $i \neq j$ since the interior of e_i and e_j have disjoint x-coordinates. Let d_i be the length of the projection of f'_i to the y-axis. As $\sqrt[p]{d_i^p + (x'_i - x_i)^p} \geq \sqrt[p]{(x'_i - x_i)^p} = x'_i - x_i$ the length of f'_i is shorter than that of e_i. This proves the claim.

Since we marked one non-tour edge for every tour edge in S, there are at least as many unmarked non-tour edges as unmarked tour edges. Note that the length of each of the unmarked non-tour edges is at least 1 and that of each of the unmarked tour edges is exactly 1. Therefore, the total length of the unmarked non-tour edges is larger than or equal the length of the unmarked tour edges. Combining with the results for the marked edges we see that C cannot have positive gain, contradiction.

□

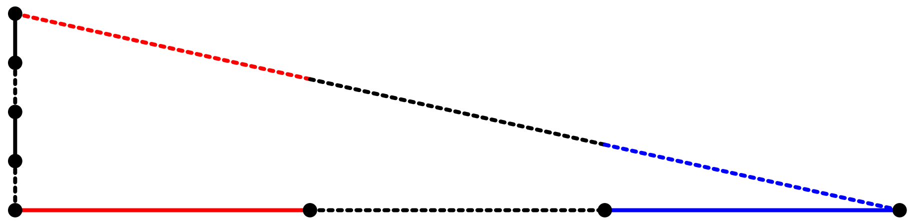


Figure 2.15: Sketch for Case 2 in the proof of Lemma 2.4.6: The solid edges are the tour edges and the dotted edges are the non-tour edges of C. The set S consists of the red and blue solid edges. The red and blue tour edges are assigned to the red and blue dotted parts, respectively.

Theorem 2.4.7. *The approximation ratio of the k-Opt algorithm for instances whose distances arise from the p-norm is $\Omega\left(\frac{\log(n)}{\log\log(n)}\right)$ where n is the number of vertices.*

Proof. By Lemma 2.4.6 there is a q such that the tour T for the instance I_q is k-optimal. By Lemma 2.4.1 and 2.4.2 the tour T has length at least $q \cdot q^{(p+1)q}$ while the optimal tour T^* for I_q has length at most $14q^{(p+1)q}$. Therefore, the approximation ratio is at least $\frac{q \cdot q^{(p+1)q}}{14q^{(p+1)q}} = \frac{q}{14}$. Recall that $n \in \Theta\left(q^{(p+1)q}\right)$ and hence $q \in \Theta\left(\frac{\log n}{\log\log n}\right)$. □

In particular, for $p = 2$ we get the result for the Euclidean TSP.

Corollary 2.4.8. *The approximation ratio of the k-Opt algorithm for* Euclidean TSP *instances is $\Omega\left(\frac{\log(n)}{\log\log(n)}\right)$ where n is the number of vertices.*

2.5 Approximation Ratio of k-Opt for (1,2)-TSP

In this section we investigate the approximation ratio of the k-Opt algorithm for the (1,2)-TSP. First, we give a lower bound for the approximation ratio of the 2-Opt algorithm of $\frac{3}{2}$. Then, we show that the approximation ratio of the 3-Opt algorithm is $\frac{11}{8}$. We introduce the k-Opt++ algorithm, a slightly modified version of the k-Opt algorithm, for the (1,2)-TSP. We prove that the approximation ratio of the 3-Opt++ algorithm is $\frac{4}{3}$. Moreover, we show that the k-Opt and k-improv algorithm have at least an approximation ratio of $\frac{11}{10}$ for all fixed k.

2.5.1 Lower Bound on the Approximation Ratio of the 2-Opt Algorithm

In this subsection we give a lower bound of $\frac{3}{2}$ on the approximation ratio of the 2-Opt algorithm for (1,2)-TSP. Note that in [43] a matching upper bound of $\frac{3}{2}$ was given and it was noted that this bound can be proven to be tight. Nevertheless, an explicit construction for the lower bound was not given. The construction is based on the construction for a lower bound of $2\left(1 - \frac{1}{n}\right)$ on the approximation ratio of the k-Opt algorithm for Metric TSP in [56].

We construct an instance with n vertices $\{v_1, \dots, v_n\}$ and a 2-optimal tour T. For the instance set the cost of the edges of $\{\{v_i, v_{i+2}\} | i \in \{1, \dots, n-2\}, i \text{ odd}\} \cup \{\{v_i, v_{i+1}\} | i \in \{1, \dots, n-1\}\}$ and $\{v_n, v_1\}$ to 1 and the cost of all other edges to 2. The tour T consists of the edges $\{v_1, v_2\}, \{v_{n-1}, v_n\}$ and $\{\{v_i, v_{i+2}\} | i \in \{1, \dots, n-2\}\}$ (Figure 2.16).

The number of edges with cost 2 in T is $\lfloor \frac{n-2}{2} \rfloor$. Thus, T has total length $n + \lfloor \frac{n-2}{2} \rfloor$. The optimal tour for this instance has length n since the tour visiting $v_1, v_2, \dots, v_n$ in

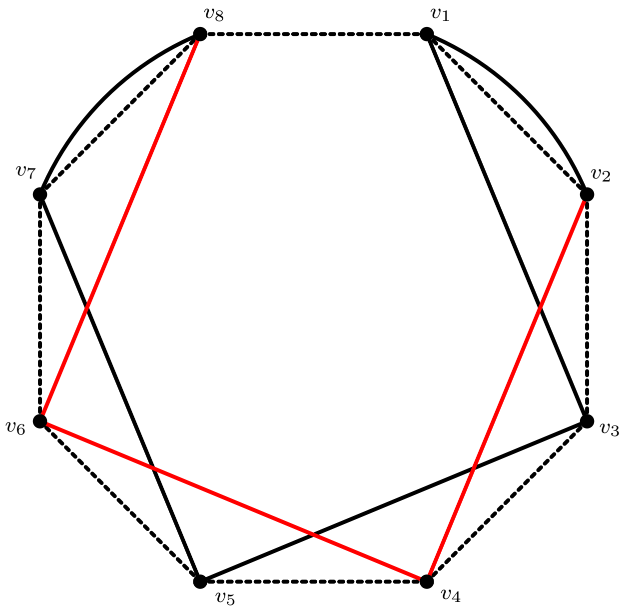


Figure 2.16: The constructed tour for $n = 8$. The black and red edges have cost 1 and 2, respectively. T is the straight tour, the optimal tour is dotted.

this order has length n and is hence optimal. Therefore, the approximation ratio is at least $\lim_{n\to\infty} \frac{n+\lfloor\frac{n-2}{2}\rfloor}{n} = \frac{3}{2}$. It remains to show that T is indeed 2-optimal for large n.

Lemma 2.5.1. *The tour T constructed above is 2-optimal for $n \geq 7$.*

Proof. Assume that there exists an improving 2-move. Then, this 2-move replaces at least an edge of length two. Fix an orientation of T such that the tour edge (v_2, v_4) is oriented this way. It is easy to see that all edges with cost two of the form (v_i, v_{i+2}) with i even are oriented this way while for i odd the edges are oriented as (v_{i+2}, v_i).

Assume that the improving 2-move replaces two edges of cost two. Then, these edges have by the definition of T the form (v_i, v_{i+2}) and (v_j, v_{j+2}) for even $i, j \in \{2, \ldots, n-2\}, i \neq j$. According to the fixed orientation the 2-move replaces (v_i, v_{i+2}) and (v_j, v_{j+2}) by $\{v_i, v_j\}$ and $\{v_{i+2}, v_{j+2}\}$. Since $i \neq j$ both even, the edges $\{v_i, v_j\}$ and $\{v_{i+2}, v_{j+2}\}$ both have cost two. Thus, this 2-move is not improving contradicting the assumption.

It remains the case that the improving 2-move replaces an edge (v_i, v_l) of cost 1 and an edge (v_j, v_{j+2}) of cost 2. Then the new edges $\{v_i, v_j\}$ and $\{v_l, v_{j+2}\}$ both must have length 1. We distinguish two cases: either $l = i - 2$ with i odd or $\{v_i, v_l\} = \{v_h, v_{h+1}\}$ for some $h \in \{1, \ldots, n\}$ where $v_{n+1} := v_1$. In the first case the difference of the indices of at least one of the new edges $\{v_i, v_j\}$ or $\{v_{i-2}, v_{j+2}\}$ has to be at least three. Hence, at least one new edge has cost 2, contradicting the assumption that we have an improving 2-move. In the second case note that the vertices of the edges $\{v_h, v_{h+1}\}$ and $\{v_j, v_{j+2}\}$ are disjoint, otherwise we do not get a tour after the 2-move. Hence, the difference of the indices of at least one new edge has to be at least three and the total cost of the new edges cannot be 2. $\square$

Theorem 2.5.2. *The approximation ratio of the 2-Opt algorithm for* (1,2)-TSP *is at least $\frac{3}{2}$.*

Proof. We have constructed an instance with a tour T which is by Lemma 2.5.1 2-optimal. Recall that the length of T is $n + \lfloor \frac{n-2}{2} \rfloor$ while the length of the optimal tour is n. Hence, the approximation ratio is at least $\lim_{n\to\infty} \frac{n+\lfloor \frac{n-2}{2} \rfloor}{n} = \frac{3}{2}$. $\square$

2.5.2 Approximation Ratio of the 3-Opt Algorithm

In this subsection we show that the exact approximation ratio of the 3-Opt algorithm is $\frac{11}{8}$.

Lower Bound on the Approximation Ratio of the 3-Opt Algorithm

We construct for all integer $s \geq 3$ an instance I_s with the vertices $\{v_0, \ldots, v_{8s-1}\}$ together with a tour T. We show that for all even $s \geq 12$ the construced tour T for the instance I_s is 3-optimal. Moreover, the ratio of its length and that of the optimal tour is at least $\frac{11s}{8s+4}$.

For simplicity we consider from now on all indicies modulo $8s$ for some fixed s. Set the cost of the edges $\{\{v_{8h}, v_{8h+1}\}, \{v_{8h+1}, v_{8h+2}\}, \{v_{8h+2}, v_{8h+3}\}, \{v_{8h+3}, v_{8h+4}\}, \{v_{8h+4}, v_{8h+5}\}, \{v_{8h+2}, v_{8h+5}\}, \{v_{8h+2}, v_{8(h+1)+5}\}, \{v_{8h+3}, v_{8h}\}, \{v_{8h+3}, v_{8(h-1)}\}, \{v_{8h+4}, v_{8h+6}\}, \{v_{8h+4}, v_{8(h+1)+6}\}, \{v_{8h+7}, v_{8(h+1)+1}\}, \{v_{8h+7}, v_{8(h+2)+1}\} | h \in \mathbb{Z}\}$ to 1 and the cost of all other edges to 2 (Figure 2.17).

The tour T consists of the edges $\{\{v_i, v_{i+1}\} | i \in \mathbb{Z}\}$. From the construction it is easy to see that the cost of the tour T is $11s$. Next, we bound the length of the optimal tour.

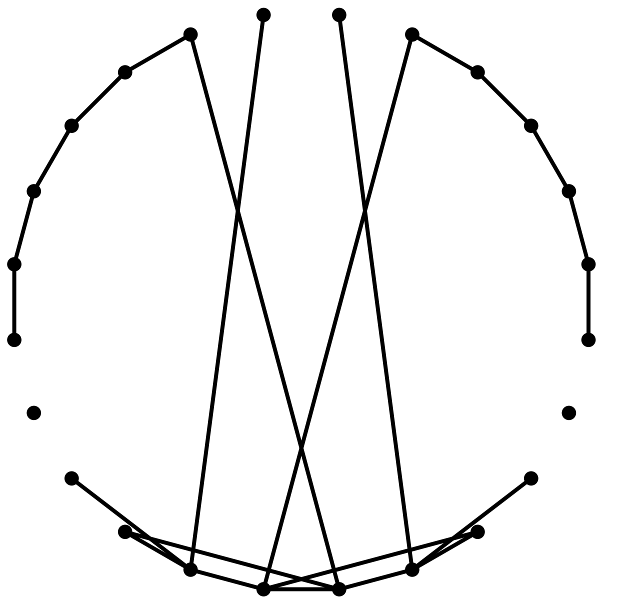


Figure 2.17: The instance I_3 where the black edges have cost 1. Due to clarity, not all edges with cost 1 are drawn. The drawn pattern of edges with cost 1 repeats periodically. The tour T connects adjacent vertices on the circle.

Lemma 2.5.3. *The optimal tour T^* for I_s has length at most $8s + 4$.*

Proof. Note that the four sets

$\{\{v_{8h+2}, v_{8h+5}\}, \{v_{8h+2}, v_{8(h+1)+5}\} | h \in \mathbb{Z}\}, \{\{v_{8h+3}, v_{8h}\}, \{v_{8h+3}, v_{8(h-1)}\} | h \in \mathbb{Z}\},$

$\{\{v_{8h+4}, v_{8h+6}\}, \{v_{8h+4}, v_{8(h+1)+6}\} | h \in \mathbb{Z}\}, \{\{v_{8h+7}, v_{8(h+1)+1}\}, \{v_{8h+7}, v_{8(h+2)+1}\} | h \in \mathbb{Z}\}$

form four vertex disjoint cycles with edges of cost 1 whose union visits every vertex exactly once. We can construct a tour of cost at most $8s+4$ for the instance by removing an arbitrary edge from each cycle and connect the four paths arbitrarily to a tour. $\square$

To show the 3-optimality of T we could make a big case distinction. But instead, we use the next Lemma that allows us to perform a computer-assisted proof.

Definition 2.5.4. A family of instances $(I'_s)_{s\in\mathbb{N}}$ for (1,2)-TSP is called *regular*, if the following conditions are satisfied:

- There is an $l \in \mathbb{N}$ such that the vertices of I'_s can be labeled by $v_0, \dots, v_{ls-1}$. In the following we consider the indicies modulo ls. We partition the vertices in *segments* such that each segment consists of the vertices $\{v_{hl}, \dots, v_{(h+1)l-1}\}$ for some $h \in \mathbb{Z}$.
- The edge $\{v_i, v_j\}$ has cost 1 if and only if $\{v_{i+l}, v_{j+l}\}$ has cost 1.
- If v_i does not lie in the same segment as any of v_j, v_{j-l} and v_{j+l}, then the edge $\{v_i, v_j\}$ has cost 2.

Lemma 2.5.5. *For a regular family of* (1,2)*-TSP instances* $(I'_s)_{s\in\mathbb{N}}$ *we have if* I'_{2k} *is* k*-optimal then* I'_s *is also* k*-optimal for all* $s \geq 2k$*.*

Proof. Assume that there is an improving k-move for some I'_s with $s \geq 2k$. We show that there is an improving k-move for I'_{2k}. Assume that the k-move removes the edges $e_1, \dots, e_k$ which lie on T in this cyclic order. If there is an $i \in \{1, \dots, k\}$ such that between e_i and e_{i+1} lie more than two complete segments where $e_{k+1} := e_1$, then we can map this k-move to I'_{s-1} by removing one of these segments. More precisely, we map $e_1, \dots, e_k$ to I'_{s-1} without changing their positions in the segments and the distances between the edges e_j and e_{j+1} for all $j \in \{1, \dots, k\}\backslash\{i\}$. The distance between e_i and e_{i+1} is by the length of a segment, i.e. by l, shorter as in I'_s. After removing $e_1, \dots, e_k$ the new k-move for I'_{s-1} connects the same endpoints of $e_1, \dots, e_k$ as the original k-move for I'_s. By the regular property, the cost of the edges we add and remove by the two k-moves are the same. Repeat this procedure and stop if we have a k-move for I'_{2k}. If no such modifications are possible, there is at most one complete segment between e_j and e_{j+1} for all $j \in \{1, \dots, k\}$, therefore the instance has at most $2k$ segments. Hence, in the end we get an improving k-move for I'_{2k}. $\square$

Lemma 2.5.6. *The constructed tour* T *is 3-optimal for* I_s *with* s *even and* $s \geq 12$*.*

Proof. For $l = 8$ the instances I_s are not regular but satisfy the first two conditions of regularity by construction. The third condition is violated since there are for example edges $\{v_{8h+7}, v_{8(h+2)+1}\}$ of cost 1 whose endpoints are up to 2 segments apart. We can construct the instance $I'_s := I_{2s}$ and choose $l = 16$ to get a regular family of instances since the third condition is also satisfied. Therefore, it is enough to check that $I'_6 = I_{12}$ is 3-optimal. We checked this using a self-written computer program that generates all possible 3-moves for the instance and observed that none of them is improving. $\square$

Theorem 2.5.7. *The approximation ratio of the 3-Opt algorithm for* (1,2)-TSP *is at least* $\frac{11}{8}$*.*

Proof. By Lemma 2.5.6, the constructed tour T is 3-optimal for I_s with s even and $s \geq 12$. By construction, it has length $11s$ and by Lemma 2.5.3 the length of the optimal tour is at most $8s + 4$. Thus, the approximation ratio is at least $\frac{11s}{8s+4} \to \frac{11}{8}$ for $s \to \infty$. $\square$

Upper Bound on the Approximation Ratio of the 3-Opt Algorithm

For the upper bound assume that an instance with a 3-optimal tour T is given. Let T^* be a fixed optimal tour of the instance. Our general strategy is like [10] to distribute counters to the vertices such that on the one hand if there are many edges of cost 2 in T, many counters are distributed. On the other hand, for many counters we need many edges of length 1 to avoid creating an improving 3-move. This way, we get a lower bound on the fraction of the edges with length 1 in T and this implies an upper bound on the approximation ratio.

Let the *1-paths* be the connected components we obtain after deleting all edges with cost 2 in T. We call the vertices with degree 1 in a 1-path the *endpoints* of the 1-path. We distribute counters as follows: For 1-paths of length 0 consisting of the vertex v we distribute two counters on the vertex w if $\{v, w\} \in T^*, c(v, w) = 1$. We call these counters *good*. For every 1-path of length greater than 0 we distribute a counter on w if v is an endpoint of the 1-path and $\{v, w\} \in T^*, c(v, w) = 1$. These counters are called *bad*.

Next, we show some properties of T we need for the analysis.

Lemma 2.5.8. *Let p, q be the endpoints of different 1-paths of a 3-optimal tour T, then $c(p, q) = 2$.*

Proof. Assume there is an edge $\{p, q\}$ connecting two endpoints p, q of different 1-paths with $c(p, q) = 1$ we show that there is an improving 3-move. Let $\{p, q\}$ be incident to the two edges $\{p, u\}$ and $\{q, v\}$ with cost 2 in T. We perform first a 2-move replacing the edges $\{p, u\}$ and $\{q, v\}$ by $\{p, q\}$ and $\{u, v\}$. Then, the cost decreases since $c(p, u) + c(q, v) = 4 > 1 + 2 \geq c(p, q) + c(u, v)$. Hence, if afterward the tour stays connected we have found an improving 2-move. It remains the case that the tour splits into two connected components. Since $\{p, q\}$ does not connect two endpoints of a single 1-path, there has to be an edge $\{a, b\}$ of cost 2 in the connected component containing p and q (Figure 2.18). We perform a 2-move replacing $\{u, v\}$ and $\{a, b\}$ by $\{a, u\}$ and $\{b, v\}$ to get a connected tour again. Note that in total we performed a single 3-move since we added $\{u, v\}$ and removed it again. In the end the total cost decreased compared to the initial tour T since

$$c(p, u) + c(q, v) + c(a, b) = 2 + 2 + 2 > 1 + 2 + 2 \geq c(p, q) + c(a, u) + c(b, v).$$

This is a contradiction to the 3-optimality of T. □

Corollary 2.5.9. *The endpoints p and q of a 1-path of T can only have counters distributed by the edge $\{p, q\}$. In particular, each of them can have at most one bad counter.*

Proof. For any other endpoint r by Lemma 2.5.8 we have $c(p, r) = c(q, r) = 2$. Therefore, these edges cannot assign a counter to p or q. □

Lemma 2.5.10. *There are no vertices p, q, u, v, a, b such that $\{p, u\}, \{q, v\}, \{a, b\} \in T$, $c(p, u) = c(q, v) = 2$, $c(a, u) = c(b, v) = 1$ and p, q, a, b lie on the same side of $\{u, v\}$ (Figure 2.18).*

Proof. Like in the proof of Lemma 2.5.8 we can replace the edges $\{p, u\}$, $\{q, v\}$ and $\{a, b\}$ by $\{p, q\}$, $\{a, u\}$ and $\{b, v\}$. The cost of the tour decreases since in this case we have

$$c(p, u) + c(q, v) + c(a, b) \geq 2 + 2 + 1 > 2 + 1 + 1 \geq c(p, q) + c(a, u) + c(b, v).$$

□

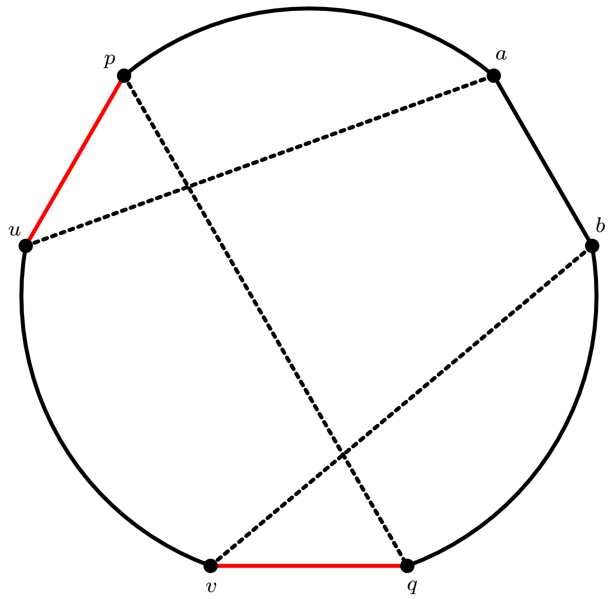


Figure 2.18: Sketch for Lemma 2.5.8 and Lemma 2.5.10. The tour T consists of the solid edges and the cost of each red edge is 2. In Lemma 2.5.8 we have in addition $c(a, b) = 2$ and $c(p, q) = 1$, while in Lemma 2.5.10 we have instead $c(a, u) = c(b, v) = 1$. In both cases we replace the edges $\{p, u\}$, $\{q, v\}$ and $\{a, b\}$ by $\{p, q\}$, $\{a, u\}$ and $\{b, v\}$.

Lemma 2.5.11. *If the vertices r and t have good counters, then $\{r, t\} \notin T$. Moreover, if there is a vertex s with $\{r, s\}, \{s, t\} \in T$, then s does not have a counter.*

Proof. For the first statement assume the contrary, then there are two 1-paths of length 0 consisting of the vertices u and v such that $c(r, u) = c(t, v) = 1$, respectively. Since u and v are 1-paths of length 0 we can choose $a = r$, $b = t$ and appropriate neighbors p and q to contradict Lemma 2.5.10.

Similarly, for the second statement assume there are such vertices r, s, t. Then, there is an endpoint w of a 1-path with $c(w, s) = 1$ and a vertex z with $\{z, w\} \in T$ and $c(z, w) = 2$. Now, z lies either on the same side of $\{w, s\}$ as r or t. Depending on this we get a contradiction to Lemma 2.5.10 for $a = s, b = t$ or $a = r, b = s$. □

Lemma 2.5.12. *Let p, q be the endpoints of a 1-path containing w with $c(w, q) = 1$. If w has a good counter, then p, q do not have counters.*

Proof. Assume the contrary, by Corollary 2.5.9 the counters of p and q are distributed by the edge $\{p, q\}$ and we must have $c(p, q) = 1$. Let the good counter of w be originated from the 1-path of length 0 consisting of the vertex u. Moreover, let w.l.o.g. v_1 be the vertex adjacent to u in T with $c(v_1, u) = 2$ and lying on the same side of $\{u, w\}$ as p and let r be the vertex with $\{q, r\} \in T$ and $c(q, r) = 2$ (Figure 2.19). Then, we can replace the edges $\{v_1, u\}$, $\{p, w\}$ and $\{q, r\}$ by $\{v_1, r\}$, $\{u, w\}$ and $\{p, q\}$. The cost of T decreases since

$$c(v_1, u) + c(p, w) + c(q, r) = 2 + 1 + 2 > 2 + 1 + 1 \geq c(v_1, r) + c(u, w) + c(p, q).$$

This is a contradiction to the 3-optimality. □

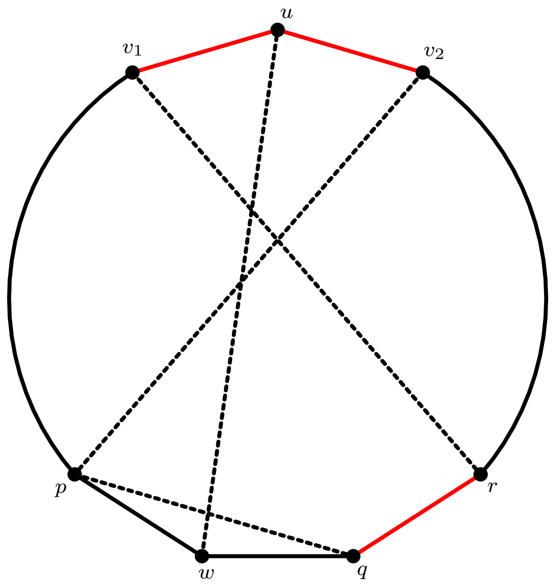


Figure 2.19: Sketch for Lemma 2.5.12 and Lemma 2.5.19. The solid edges are the edges of the tour T and the red edges have cost 2. In Lemma 2.5.12 we replace the edges $\{v_1, u\}$, $\{p, w\}$ and $\{q, r\}$ by $\{v_1, r\}$, $\{u, w\}$ and $\{p, q\}$. In Lemma 2.5.19 the edges $\{u, v_2\}$ and $\{p, w\}$ are replaced by $\{v_2, p\}$ and $\{u, w\}$.

Before we bound the number of counters, we summarize the properties we showed about counters.

Corollary 2.5.13. *The following properties hold:*

1. *Every vertex has two* slots *where a slot can be empty or contain two good counters or one bad counter.*

2. *If the vertices a and c both have good counters, then $\{a, c\} \notin T$. Moreover, if there is a vertex b such that $\{a, b\}, \{b, c\} \in T$, then b does not have counters.*

3. *The endpoint of a 1-path of length 0 does not have counters. Each endpoint of all other 1-paths can only have at most one bad counter and no good counters.*

4. *If the endpoint p of a 1-path has a counter, then w does not have a good counter if $\{w, p\} \in T$ and $c(w, p) = 1$.*

5. *The total number of bad counters is less or equal to four times the number of 1-paths of length greater than 0.*

Proof. The first property is due to the fact that every vertex is incident to at most two edges of cost 1 in the optimal tour and every such edge can distribute two good counters or one bad counter. The second, third and fourth property follow from Lemma 2.5.11, Corollary 2.5.9 and Lemma 2.5.12, respectively. The fifth property follows from the fact that the bad counters are distributed by the endpoints of 1-paths with length greater than 0 and each such endpoint distributes at most two bad counters. □

Lemma 2.5.14. *Every 3-optimal tour T has at most $\frac{12}{5}h$ counters, where h is the number of edges with cost 1 in T.*

Proof. Assume the contrary that there is an instance together with a tour T with more than $\frac{12}{5}h$ counters, where h is the number of edges with cost 1 in T. Take such an instance and tour T with h minimal. We want to get a contradiction only using the properties in Corollary 2.5.13. More precisely, we discard any other properties of counters that are implied by the 3-optimality of T and assume that the counters are distributed arbitrarily on the vertices of T such that the properties in Corollary 2.5.13 are satisfied.

First, note that if a vertex v has two good counters, then we may assume that it has four good counters instead of exactly two good counters or two good counters and one bad counter. It is easy to check that this assumption does not decrease the number of counters or contradict Corollary 2.5.13 since v already had a good counter.

Next, we exclude the case that there are two vertices a and c on the same 1-path with good counters such there is a vertex b with $\{a,b\},\{b,c\} \in T$. In this case by property 2 in Corollary 2.5.13 the vertex b does not have a counter. Let u be the neighbor of a in T other than b. We remove the vertices a and b from the instance, remove the edges $\{u,a\}$, $\{a,b\}$ and $\{b,c\}$ and add the edge $\{u,c\}$ to T and set its cost to 1. By property 3 in Corollary 2.5.13, u belongs to the same 1-path as a in the original tour, hence the new tour does not contradict property 3. This also implies that the number of edges with cost 1 decreased by 2 while the number of counters decreased by 4. Moreover, we did not introduce new bad counters or new vertices neighboring to a vertex with a good counter. Hence, the new tour does not contradict Corollary 2.5.13. Let the number of counters before the modification be s. Since the edges $\{u,a\}$, $\{a,b\}$ and $\{b,c\}$ all had cost 1 in the old tour we have $h-2 \geq 1$. We conclude that the ratio $\frac{s}{h}$ does not decrease as

$$\frac{s}{h} > \frac{12}{5} > 2 \Rightarrow 2s > 4h \Rightarrow \frac{s-4}{h-2} > \frac{s}{h} > \frac{12}{5}.$$

This contradicts that we chose a tour T with more than $\frac{12}{5}h$ counters and h minimal. Thus, the distance of two vertices a and c on the same 1-path each having good counters is at least 3.

We define for $i \in \mathbb{N}_{>0}$

$$\begin{aligned}
b_i &:= 4 \cdot \frac{i}{3} - 2, & g_i &:= 4 \cdot \frac{i}{3} & &\text{for } i \equiv 0 \mod 3\\
b_i &:= 4 \cdot \frac{i-1}{3} + 2, & g_i &:= 4 \cdot \frac{i-1}{3} & &\text{for } i \equiv 1 \mod 3\\
b_i &:= 4 \cdot \frac{i-2}{3}, & g_i &:= 4 \cdot \frac{i+1}{3} & &\text{for } i \equiv 2 \mod 3.
\end{aligned}$$

Since the distance of two vertices on a 1-path each having good counters is at least 3 and the endpoints of the 1-path do not contain good counters, the number of good counters on a 1-path with i edges is at most g_i. To maximize the number of counters, we have to maximize the number of vertices with good counters, since every vertex can contain at most four good counters or two bad counters. For $i \equiv 0 \mod 3$ and $i \equiv 2 \mod 3$ we can either have no counters on the endpoints of the 1-path and g_i good counters and at most b_i bad counters or one bad counter on each of the endpoints but by property 4 in Corollary 2.5.13 only $g_i - 4$ good counters and at most $b_i + 2$ additional bad counters. The total number of counters is the same but the first case has the advantage that we have fewer bad counters and the total number of bad counters is bounded by property 5 in Corollary 2.5.13. For $i \equiv 1 \mod 3$ each endpoint of the 1-path can contain a bad counter such that in total we have g_i good counters and at most b_i bad counters.

Therefore, we may assume for all i that a 1-path with i edges has g_i good counters and at most b_i bad counters for all $i \in \mathbb{N}_{i>0}$. Every 1-path of length 0 does not contain counters by property 3 in Corollary 2.5.13.

Consider the following LP:

$$\begin{aligned} \max \sum_{i\in\mathbb{N}} g_i x_i + & z \\ s.t. \qquad \sum_{i\in\mathbb{N}} i x_i &= h \\ z - \sum_{i\in\mathbb{N}} b_i x_i &\le 0 \\ z - 4\sum_{i\in\mathbb{N}} x_i &\le 0 \\ x_i &\ge 0 \qquad \forall i \in \mathbb{N}_{>0} \end{aligned}$$

This LP gives an upper bound on the number of counters in a tour with h edges of cost 1. The variable x_i counts the number of 1-paths with i edges. Each of these 1-paths have g_i good counters and at most b_i bad counters. The variable z is equal to the total number of bad counters. The first equation ensures that the total number of edges with cost 1 is h while the second inequality bounds the number of bad counters by the sum of the upper bounds of bad counters on each 1-path. The third inequality ensures that the number of bad counters is at most four times the number of 1-paths with length greater than 0 since every such 1-path distributes at most 4 bad counters.

Note that by the first equation for any fixed h only a finite number of x_i are nonzero. Now, consider the dual LP:

$$\begin{aligned} & \min\ y_1 h \\ s.t. \qquad i y_1 - b_i y_2 - 4 y_3 &\ge g_i \qquad \forall i \in \mathbb{N}_{>0} \\ y_2 + y_3 &\ge 1 \\ y_2, y_3 &\ge 0. \end{aligned}$$

We show that $y_1 = \frac{12}{5}, y_2 = \frac{4}{5}$ and $y_3 = \frac{1}{5}$ is a feasible dual solution with value $\frac{12}{5}h$. Therefore, by weak duality we get $\frac{12}{5}h$ as upper bound on the primal LP. Obviously, we have $y_2 + y_3 = 1$ and $y_2, y_3 \ge 0$. Moreover, we can check the first inequality:

$$\begin{aligned} i \cdot \frac{12}{5} - \left(4 \cdot \frac{i}{3} - 2\right) \cdot \frac{4}{5} - 4 \cdot \frac{1}{5} = 4 \cdot \frac{i}{3} + \frac{4}{5} > 4 \cdot \frac{i}{3} & \qquad \text{for } i \equiv 0 \mod 3 \\ i \cdot \frac{12}{5} - \left(4 \cdot \frac{i-1}{3} + 2\right) \cdot \frac{4}{5} - 4 \cdot \frac{1}{5} = 4 \cdot \frac{i-1}{3} & \qquad \text{for } i \equiv 1 \mod 3 \\ i \cdot \frac{12}{5} - 4 \cdot \frac{i-2}{3} \cdot \frac{4}{5} - 4 \cdot \frac{1}{5} = 4 \cdot \frac{i+1}{3} & \qquad \text{for } i \equiv 2 \mod 3. \end{aligned}$$

Thus, the first inequality is also satisfied and we get the upper bound of $\frac{12}{5}h$ on the number of counters. □

Lemma 2.5.15. *Let the number of counters in the tour T be at most $d \cdot h$ where d is a constant and h is the number of edges with cost 1 in T. Then, the ratio between the length of T and that of the optimal tour is at most $1 + \frac{d}{4+d}$.*

Proof. Let l and f be the number of edges with cost 2 in T and the optimal tour, respectively. If the optimal tour consists only of edges with cost 1, every 1-path distributes four counters: For every 1-path of length 0 the unique endpoint is incident to two edges of length 1 in the optimal tour and distributes four good counters. Every endpoint of the other 1-paths is incident to two edges of length 1 in the optimal tour and distributes two bad counters. Every edge of cost 2 in the optimal tour decreases the number of counters distributed by at most 4. Note that the number of 1-paths is equal to the number of edges with cost 2 which is l. Therefore, the l 1-paths distribute at least $4l - 4f$ counters and we conclude $dh \geq 4l - 4f$ or $h \geq \frac{1}{d}4(l - f)$. Note that the length of the tour T is $h + 2l$, while that of the optimal tour T^* is $n + f = h + l + f$. Therefore, we get the upper bound on the ratio of

$$\frac{c(T)}{c(T^*)} = \frac{h + 2l}{h + l + f} = 1 + \frac{l - f}{h + l + f} \leq 1 + \frac{l - f}{\frac{1}{d}4(l - f) + l + f} = 1 + \frac{l - f}{\frac{4+d}{d}l - \frac{4-d}{d}f}$$
$$\leq 1 + \frac{l - f}{\frac{4+d}{d}l - \frac{4+d}{d}f} = 1 + \frac{d}{4 + d}.$$

□

Theorem 2.5.16. *The approximation ratio of the 3-Opt algorithm for* (1,2)-TSP *is at most* $\frac{11}{8}$.

Proof. By Lemma 2.5.14, there are at most $\frac{12}{5}h$ counters in T where h is the number of edges with cost 1 in T. By Lemma 2.5.15, the approximation ratio is at most $1 + \frac{\frac{12}{5}}{4+\frac{12}{5}} = \frac{11}{8}$. □

2.5.3 Approximation Ratio of the 3-Opt++ Algorithm

In the last subsection we showed that the approximation ratio of the 3-Opt algorithm is $\frac{11}{8}$. The 1-paths with length 0 play a central role in the analysis. Every endpoint of such a 1-path distribute at most four good counters instead of two bad counters and increase the approximation ratio. We introduce the k-Opt++ algorithm by adapting a concept from the k-improv algorithm. The k-Opt++ algorithm does not only search for improving k-moves but also k-moves that decrease the number of 1-paths of length 0 without increasing the cost of the tour. We show that the 3-Opt++ algorithm has a better approximation ratio of $\frac{4}{3}$ than 3-Opt.

The k-Opt++ algorithm

We first describe the k-Opt++ algorithm. An *improving k-Opt++-move* is a k-move that is either improving or does not change the cost of the tour but decreases the number of 1-paths of length 0 of the tour. A tour is called *k-Opt++-optimal* if there does not exist an improving k-Opt++-move. Like the k-Opt algorithm the k-Opt++ algorithm starts with an arbitrary tour T and performs improving k-Opt++ moves until T is k-Opt++-optimal (Algorithm 4).

Obviously, every k-Opt++-optimal tour is also k-optimal. Moreover, every iteration of the k-Opt++ algorithm can also be performed in polynomial time since the number of k-moves is polynomially bounded. Thus, the algorithm also terminates in polynomial time since the length of the initial tour is bounded by $2n$ and every step improves the length of the tour by at least 1 or decreases the number of 1-paths with length 0 by 1.

Algorithm 4 k-Opt++ Algorithm

Input: Instance of TSP (K_n, c)
Output: Tour T

Start with an arbitrary tour T
while $\exists$ improving k-Opt++-move for T **do**
    Perform an improving k-Opt++-move on T
return T

Lower Bound on the Approximation Ratio of the 3-Opt++ Algorithm

For every natural number $s \geq 2$ we construct an instance I_s with the vertices $\{v_0, \dots, v_{6s-1}\}$ and approximation ratio $\frac{4}{3}$. For simplicity we consider the indicies modulo $6s$. Set the cost of the edges $\{\{v_{6h}, v_{6h+1}\}, \{v_{6h+2}, v_{6h+3}\}, \{v_{6h+3}, v_{6h+4}\}, \{v_{6h+4}, v_{6h+5}\}, \{v_{6h}, v_{6h+3}\}, \{v_{6h+2}, v_{6h+5}\}, \{v_{6h+4}, v_{6(h+1)+1}\} | h \in \mathbb{Z}\}$ to 1 and the cost of all other edges to 2.

The tour T consists of the edges $\{\{v_i, v_{i+1}\} | i \in \mathbb{Z}\}$. The optimal tour T^* consists of the edges $\{\{v_{2h+1}, v_{2h}\}, \{v_{2h}, v_{2h+3}\} | h \in \mathbb{Z}\}$ (Figure 2.20). It is easy to check that T^* consists only of edges with cost 1 and is hence optimal.

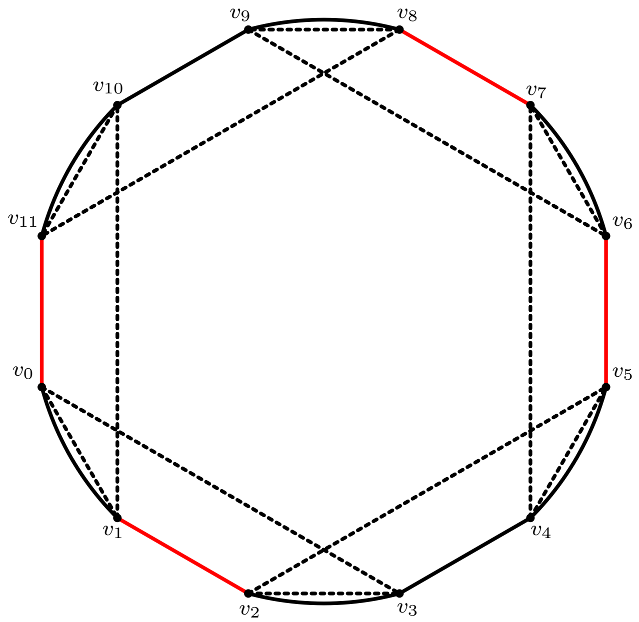


Figure 2.20: The constructed tour for $s = 2$. The black and red edges have cost 1 and 2, respectively. T is the straight tour, the optimal tour T^* is dotted.

From the construction it is easy to see that the cost of the tour T is $8s$ while that of the optimal tour is $6s$.

Lemma 2.5.17. *For I_s with $s \geq 6$ the tour T constructed above is 3-Opt++-optimal .*

Proof. Since the constructed instances $(I_s)_{n \in \mathbb{N}}$ do not contain 1-paths of length 0, the tour T is 3-Opt++-optimal if and only if T is 3-optimal. By construction, the constructed family of instance $(I_s)_{s \in \mathbb{N}}$ is regular. Hence, by Lemma 2.5.5 it is enough to check that I_6 is 3-optimal. We checked this using a self-written computer program that generates all possible 3-moves for the instance and observed that none of them is improving. $\square$

Theorem 2.5.18. *The approximation ratio of the 3-Opt++ algorithm for* (1,2)-TSP *is at least* $\frac{4}{3}$.

Proof. By Lemma 2.5.17, the tour T for I_s with $s \geq 6$ is 3-Opt++-optimal. Moreover, we know that the length of T is $8s$ and that of the optimal tour T^* is $6s$. Therefore, we get an approximation ratio of $\frac{8s}{6s} = \frac{4}{3}$. □

Upper Bound on the Approximation Ratio of the 3-Opt++ Algorithm

For the upper bound assume that an instance with a 3-Opt++-optimal tour T is given. Let the counters be distributed as in the analysis of the upper bound of the 3-Opt algorithm. Since every 3-Opt++-optimal tour is also 3-optimal, we can apply the results from the upper bound of the 3-Opt algorithm.

Lemma 2.5.19. *Every 1-path in a 3-Opt++-optimal tour T having a good counter has exactly two edges.*

Proof. Let the good counter be assigned to the vertex w from the 1-path of length 0 consisting of the vertex u which is incident to the edges $\{v_1, u\}$ and $\{u, v_2\}$ of cost 2 in T. Then $\{u, w\}$ has cost 1 and assume that w is not the internal vertex of a 1-path of length 2. By Lemma 2.5.8 w cannot be an endpoint of a 1-path. Hence, there is an edge $\{p, w\} \in T$, $c(p, w) = 1$ such that p is not endpoint of a 1-path. W.l.o.g. assume that p lies on the same side of $\{u, w\}$ as v_1 (Figure 2.19). Now, we can replace the edges $\{u, v_2\}$ and $\{p, w\}$ by $\{v_2, p\}$ and $\{u, w\}$. As

$$c(u, v_2) + c(p, w) = 2 + 1 \geq c(v_2, p) + c(u, w),$$

after the 2-move the cost of T does not increase. Moreover, we have one 1-path of length 0 less than before, namely u. Since p is not endpoint of a 1-path we did not create new 1-paths of length 0, contradicting the 3-Opt++-optimality of T. □

Lemma 2.5.20. *Every 1-path with x edges in a 3-Opt++-optimal tour T has at most $2x$ counters.*

Proof. For $x = 0$ the 1-path consists of one vertex and by Corollary 2.5.9 it cannot have any counters. For $x \geq 1$ every 1-path with x edges has two endpoints and $x - 1$ internal vertices. If the 1-path does not have a good counter, then each of the inner vertices has two incident edges in the optimal tour and can have at most two bad counters. By Corollary 2.5.9 the two endpoints can each have at most one counter. Therefore, the 1-path can have in total at most $2x$ counters. For the case that the 1-path does have a good counter by Lemma 2.5.19 the 1-path must have exactly two edges and there are at most $2 \cdot 2 = 4$ good counters on the internal vertex. Moreover, by Lemma 2.5.12 the endpoints of this 1-path cannot have any counters. Thus, this 1-path can have at most 4 counters. □

Theorem 2.5.21. *The approximation ratio of the 3-Opt++ algorithm for* (1,2)-TSP *is at most* $\frac{4}{3}$.

Proof. By Lemma 2.5.20, we distributed at most $2h$ counters where h is the number of edges with cost 1 in T. Hence, by Lemma 2.5.15 the approximation ratio is at most $1 + \frac{2}{4+2} = \frac{4}{3}$. □

2.5.4 Lower Bound on the Approximation Ratio of the k-Improv and k-Opt Algorithm

In this subsection we show that the approximation ratio of the k-improv algorithm is at least $\frac{11}{10}$ for arbitrary fixed k. For given fixed $k \geq 2$ and $\epsilon > 0$, we construct a k-improv-optimal instance $I_{k,\epsilon}$ with approximation ratio at least $\frac{11}{10} - \epsilon$. Moreover, we show that every corresponding tour of a $2k$-improv-optimal 2-matching is also k-optimal. Thus, this lower bound on the approximation ratio also carries over to the k-Opt algorithm.

We first construct some auxiliary graphs before the construction of the instance. Let S be a graph with 10 vertices $w_0, \dots, w_9$ and the edges $\{w_0, w_1\}$, $\{w_0, w_4\}$, $\{w_2, w_3\}$, $\{w_3, w_4\}$, $\{w_5, w_9\}$, $\{w_5, w_6\}$, $\{w_6, w_7\}$ and $\{w_8, w_9\}$ (Figure 2.21).

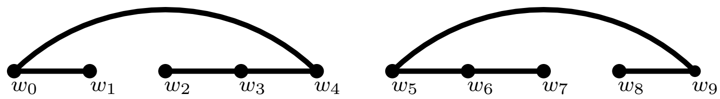


Figure 2.21: The graph S consists of 10 vertices and the drawn edges.

By Lemma 1.4.11, for $g := \max\{2k+1, \frac{1}{\frac{11}{11-10\epsilon}-1}\}$ and $2s' \geq 3^{g-1} - 1$ there exists a 4-regular graph G_0 with $2s'$ vertices and girth at least g. Construct a 4-regular bipartite graph G_1 from G_0 with $s := 4s'$ vertices and girth at least g as follows: Add two copies u and u' of every vertex $u \in V(G_0)$ to the vertex set $V(G_1)$. Then, add edges $\{u, v'\}$ and $\{u', v\}$ to $E(G_1)$ for every edge $\{u, v\} \in E(G_0)$. By construction, G_1 is bipartite and still 4-regular. Moreover, it still has girth at least g since every cycle in G_1 can be mapped to a closed walk of the same length in G_0 by mapping the copies of the vertices to the original vertex.

Since G_1 is bipartite and 4-regular, we can color by Theorem 1.4.2 the edges of G_1 with four colors such that no two incident edges have the same color. Now, we color every vertex with degree one of the graph S, namely w_1, w_2, w_7 and w_8, with one of the four colors such that each color is used once. Next, we want to construct a graph G_S with the vertex set $\{v_0, \dots, v_{10s-1}\}$. For simplicity we consider in the following all indices modulo $10s$. For the construction we replace every vertex of G_1 by a copy of S such that the vertices $w_0, \dots, w_9$ of each copy of S is mapped to $v_{10h}, \dots, v_{10h+9}$ in this order for some $h \in \mathbb{Z}$. For every edge $\{u, v\} \in E(G_1)$ we connect the two vertices with the color of $\{u, v\}$ in the corresponding copies of S (Figure 2.22 and Figure 2.23). The 4-edge coloring and 4-regularity ensures that this procedure is well defined and every vertex of G_S has degree 2.

Figure 2.22: A part of a graph G_1 from which we will construct a part of the graph G_S. The edges of the 4-regular graph are colored with four colors.

The vertex set of the instance $I_{k,\epsilon}$ is $V(G_S)$. The set of edges of $I_{k,\epsilon}$ with cost 1 is the union of $E(G_S)$ with the edges $\{\{v_{10h+4}, v_{10h+5}\} | h \in \mathbb{Z}\}$. All other edges have cost 2.

The tour T consists of the edges $\{\{v_i, v_{i+1}\} | i \in \mathbb{Z}\}$. It has cost $11s$ since all of its edges except $\{\{v_{10h+9}, v_{10h+10}\} | h \in \mathbb{Z}\}$ have cost 1. Let $\widetilde{T}$ be the corresponding 2-matching we

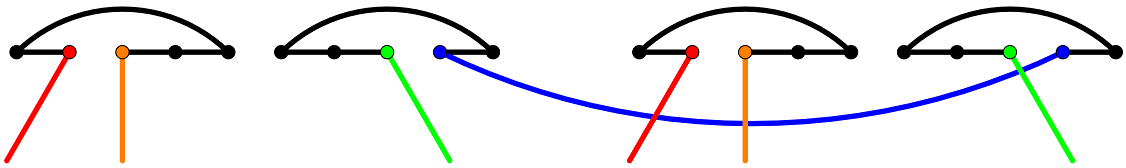


Figure 2.23: A part of the graph G_S we constructed from the part of G_1 shown in Figure 2.22. For its construction replace every vertex of G_1 with a copy of S. Color the vertices with degree 1 in S by the four colors. For every edge in G_1 connect the vertices with the color of the edge in the corresponding copies of S in G_S.

get by removing all edges with degree 2 in T. Next, we show that $\widetilde{T}$ is k-improv-optimal.

Lemma 2.5.22. *The corresponding 2-matching $\widetilde{T}$ to the tour T constructed above for the instance $I_{k,\epsilon}$ is k-improv-optimal.*

Proof. Assume that there is an improving k-improv-move for $\widetilde{T}$ and $\widetilde{T}'$ is the result after performing it. Consider $\widetilde{T}' \triangle \widetilde{T}$, it can be decomposed into edge-disjoint alternating cycles and paths. We choose such a decomposition that contains a minimal number of alternating paths, i.e. we cannot merge two alternating paths to a longer alternating path. We distinguish four types of alternating cycles and paths:

1. Alternating cycles.
2. Alternating paths starting and ending with non-tour edges.
3. Alternating paths starting and ending with exactly one tour edge and one non-tour edge.
4. Alternating paths starting and ending with tour edges.

First, we show that there are no alternating cycles. Assume that there is an alternating cycle. Note that the tour edges of $\widetilde{T}$ do not connect two copies of S. Therefore, if there is a cycle visiting at least two copies of S, its non-tour edges have to form a cycle in G_1. Since G_1 has girth at least g it has to contain at least $g > k$ non-tour edges, contradiction. Now, assume that the cycle only visits one copy of S. Since we chose a decomposition of alternating paths and cycles that is edge-disjoint and every edge is contained in $\widetilde{T}$ at most once, we cannot add an edge that is already in $\widetilde{T}$ in an alternating path or cycle. As the edges $\{w_0, w_4\}$ and $\{w_5, w_9\}$ are the only edges with cost 1 in S that are not contained in $\widetilde{T}$, they have to be the non-tour edges of the alternating cycle. But then the edge $\{w_0, w_9\}$ is not in $\widetilde{T}$ and cannot be a tour edge, contradiction.

With the same argument as above we can show that any augmenting path does not visit any copy of S twice. We will need this property later.

Alternating paths starting and ending with non-tour edges increase the number of edges in the 2-matching by one. Moreover, to maintain the property that every vertex in $\widetilde{T}'$ has degree 1 or 2, they have to start and end at a vertex with degree 1 in T, i.e. in w_0 or w_9 in a copy of S. Since the decomposition we chose is edge-disjoint, there is only one possibility for such a path, namely the path (w_0, w_4, w_5, w_9) in one copy of S. Note that each of such paths produces two cycles in that copy of S by adding the edges $\{w_0, w_4\}$ and $\{w_5, w_9\}$. We call alternating paths of this type *cycle-creating*.

Starting with $\widetilde{T}$ we augment all cycle-creating alternating paths. They create $2q$ cycles in q copies of S where q is the number of these paths. We will call these copies of S *cycle-containing*.

Alternating paths that start and end with exactly one tour edge and one non-tour edge do not change the number of edges in the 2-matching. Note that there are only four possibilities for such alternating paths, namely such visiting the following vertices of a single copy of S: (w_0, w_4, w_5); (w_0, w_4, w_3); (w_9, w_5, w_4) and (w_9, w_5, w_6). Note that each of the possibilities adds either $\{w_0, w_4\}$ or $\{w_5, w_9\}$. Thus, the copy of S visited by such a path is not cycle-containing since the decomposition to alternating paths is edge-disjoint. Hence, these alternating paths do not remove any cycle created by cycle-creating paths.

Last, alternating paths starting and ending with tour edges decrease the number of edges in the 2-matching by one. As the alternating paths of the previous type do not remove cycles, either these paths remove a cycle created by the cycle-creating alternating paths or the cycle remains in $\widetilde{T}'$. The alternating paths of this type are called *cycle-removing*.

We construct an auxiliary multigraph G_P with q vertices. Each of its vertices corresponds to one cycle-containing copy of S in G_S. For every cycle-removing alternating path p we construct a path p' in G_P we call the *S-path* of p as follows: The vertices of p' are the vertices in G_P corresponding to the copies of S visited by p. The edges of p' connect the vertices in the order in which the corresponding copies of S are visited by p (Figure 2.24). Note that p' may consist of only one vertex or the empty set. Since we showed that every alternating path visits any copy of S at most once, p' is indeed a path without self-loops. The edge multiset $E(G_P)$ consists of the disjoint union of all S-paths.

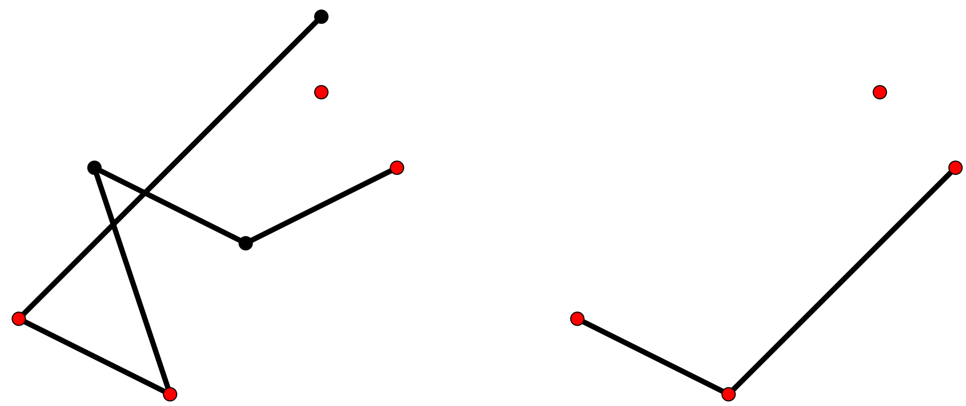


Figure 2.24: The construction of the S-paths. Let p be a cycle-removing alternating path in G_S. Left: The edges of p in G_1 after contracting the copies of S and removing self-loops. The red vertices are cycle-containing. Right: the corresponding S-path p' in G_P of p. It visits the red vertices in the same order as p in G_1.

If G_P contains a cycle, the cycle corresponds to a closed walk in G_1 by considering for every edge in the cycle the corresponding S-path p' it belongs to and the corresponding subpath in p. The length of this closed walk is bounded by the number of edges in the cycle-removing alternating paths, which is at most $2k$. As $g > 2k$ and G_1 has girth at least g we get a contradiction. Hence, G_P is acyclic and in particular a simple graph.

Note that if a cycle-removing alternating path visits vertices from $\{w_0, w_1, w_2, w_3, w_4\}$ and $\{w_5, w_6, w_7, w_8, w_9\}$ in a copy of S without leaving the copy, it has to add the edges $\{w_0, w_4\}$ or $\{w_5, w_9\}$ since the decomposition is edge-disjoint. In this case this copy of S is not cycle-containing again by the edge-disjoint property of the decomposition. Thus, every vertex of an S-path can remove at most one cycle. Therefore, every cycle-removing alternating path with h edges can remove at most $h + 1$ cycles.

Since $\widetilde{T}$ does not contain singletons, either $\widetilde{T}'$ contains less connected components than $\widetilde{T}$ or the same number but more cycles. In both cases $\widetilde{T}'$ contains more edges than $\widetilde{T}$. Recall that a cycle-creating and cycle-removing alternating path increases and decreases the number of tour edges by one, respectively. Thus, there are at most $r \leq q - 1$ cycle-removing alternating paths. Note that G_P can have in total at most $q - 1$ edges since it is a simple acyclic graph with q vertices. Thus, by definition also the union of the S-paths contain at most $q - 1$ edges. We conclude that at most $r + (q - 1)$ cycles can be removed and at least $2q - r - (q - 1) = q + 1 - r$ cycles are remaining. Therefore, $\widetilde{T}'$ must contain at least $q + 1 - r$ more edges than $\widetilde{T}$ to maintain at least the same number of connected components as $\widetilde{T}'$. Since $\widetilde{T}'$ contains $q - r$ more edges than $\widetilde{T}$, we have $q - r \geq q + 1 - r$, which is a contradiction. □

Next, we show that if we start the k-improv algorithm with T, then it computes a tour with the same cost as T.

Lemma 2.5.23. *If the k-improv algorithm starts with the tour T constructed above, it outputs a tour with the same cost as T.*

Proof. The k-improv algorithm first computes the corresponding 2-matching $\widetilde{T}$ from T. By Lemma 2.5.22 $\widetilde{T}$ is k-improv-optimal, hence the algorithm cannot make any improvements. Therefore, it computes a tour corresponding to $\widetilde{T}$ and outputs it. Thus, it is enough to show that every tour corresponding to the k-improv-optimal 2-matching $\widetilde{T}$ has the same cost as T. Since $\widetilde{T}$ arises from removing all edges of cost 2 from T any tour corresponding to $\widetilde{T}$ has at most the cost of T. Assume that there is a tour corresponding to $\widetilde{T}$ with less cost than T. Then, there is at least one edge connecting two vertices in $\widetilde{T}$ with degree 1. Adding this edge to $\widetilde{T}$ would reduce the number of connected components, contradiction to the k-improv-optimality of $\widetilde{T}$. □

Now, we need an upper bound on the length of the optimal tour before we can conclude the lower bound on the approximation ratio.

Lemma 2.5.24. *The optimal tour T^* of the instance $I_{k,\epsilon}$ has at most cost $10s + \frac{10s}{g}$.*

Proof. Note that the edges in G_S form disjoint cycles since the degree of every vertex is exactly 2. Moreover, every cycle in G_S corresponds to a closed walk in G_1 if we contract the copies of S. Since the girth of the graph G_1 is at least g and S is acyclic, the girth of G_S is also at least g. Thus, each of the disjoint cycles in G_S has at least g edges. We can get a tour by removing an arbitrary edge from each cycle and arbitrarily add edges to complete the paths to a tour. Recall that all edges in G_S have cost 1. Hence, we introduced at most one edge with cost 2 for every $g - 1$ edges with cost 1 and the constructed tour has at most length $|V(G_S)| + \frac{|V(G_S)|}{g} = 10s + \frac{10s}{g}$. □

Theorem 2.5.25. *The approximation ratio of the k-improv algorithm with arbitrarily fixed k for* (1,2)-TSP *is at least $\frac{11}{10}$.*

Proof. The constructed tour T has length $11s$ and by Lemma 2.5.23 the k-improv algorithm outputs a tour of this length if starting with T. The optimal tour T^* has cost at most $10s + \frac{10s}{g}$ by Lemma 2.5.24. Recall that for any fixed $\epsilon > 0$ we chose $g \geq \frac{1}{\frac{11}{11-10\epsilon}-1}$ which implies $\frac{11}{10} \cdot \frac{1}{1+\frac{1}{g}} \geq \frac{11}{10} - \epsilon$. Hence, for every $\epsilon > 0$ there exists an instance with approximation ratio at least

$$\frac{c(T)}{c(T^*)} = \frac{11s}{10s + \frac{10s}{g}} = \frac{11s}{10s(1+\frac{1}{g})} = \frac{11}{10} \cdot \frac{1}{1+\frac{1}{g}} \geq \frac{11}{10} - \epsilon.$$

□

In the end we show that we can carry over the result to the k-Opt algorithm.

Lemma 2.5.26. *The constructed tour T above for the instance $I_{2k,\epsilon}$ is k-optimal.*

Proof. Assume that there is an improving k-move after which augmentation we get a shorter tour T'. Let $\widetilde{T}'$ be the corresponding 2-matching we obtain by removing all edges with cost 2 from T'. Then, $\widetilde{T}'$ must contain less connected component than $\widetilde{T}$ and we can perform a $2k$-improv-move to obtain $\widetilde{T}'$ from $\widetilde{T}$. This is a contradiction to the $2k$-improv-optimality of $\widetilde{T}$ by Lemma 2.5.22. □

Remark 2.5.27. The fact that we need a $2k$-improv-optimal (instead of a k-improv-optimal) 2-matching to ensure that every corresponding tour is k-optimal is caused by the different definitions of the two algorithms. In contrast to a k-move where at most k edges can be removed and added a k-improv-move is defined such that at most k edges can be removed and added in total.

Theorem 2.5.28. *The approximation ratio of the k-Opt algorithm with arbitrarily fixed k for* (1,2)-TSP *is at least $\frac{11}{10}$.*

Proof. For all given k and $\epsilon > 0$ we can construct a tour T for the instance $I_{2k,\epsilon}$ with length $11s$. By Lemma 2.5.26, the tour T is k-optimal and by Lemma 2.5.24 the optimal tour has cost at most $10s + \frac{10s}{g}$. By a similar calculation as in Theorem 2.5.25, we get the result that the approximation ratio is at least $\frac{11}{10} - \epsilon$. □

2.6 A Polynomial Time Local Search Algorithm for Graph TSP

In Section 2.3 we showed that the k-Opt algorithm does not have a constant approximation ratio in the graphic case for any fixed k. Nevertheless, it can be modified to achieve a constant approximation ratio. In this section we present a polynomial time local search algorithm for Graph TSP we call the incomplete n-Opt algorithm. In contrast to the k-Opt algorithm, this algorithm considers alternating cycles of arbitrary length but only these having a large gain. As we will show later it has a constant approximation ratio of 5.

Definition 2.6.1. Given a tour T in G we construct a directed weighted auxiliary graph G_T. Start with an empty graph G_T, for every vertex $v \in V(G)$ add two vertices v and v' to $V(G_T)$. For every edge $\{u, v\} \in T$ add the edges (u, v') and (v, u') to $E(G_T)$ with cost $-c(u, v)$. For any pair of distinct vertices $u, v \in V(G)$ we add the edges (u', v) and (v', u) with cost $c(u, v) + 2$ to $E(G_T)$.

The next observation is crucial for the algorithm.

Lemma 2.6.2. *Every cycle in G_T can be transformed into a closed alternating walk in G and every closed alternating walk in G can be transformed into a closed walk in G_T.*

Proof. By the construction of G_T, we can transform edges of a cycle C in G_T to a closed alternating walk in G as follows: Map every edge of the form $(u, v') \in C$ with $u, v \in V(G)$ to the tour edge $\{u, v\} \in T$ and every edge of the form (u', v) to the non-tour edge $\{u, v\}$. By construction, the graph G_T is bipartite, hence C contains alternately edges of both forms. Therefore, the transformed cycle is indeed a closed alternating walk. The transformation can also be done in the other direction. $\square$

Our algorithm starts, as the k-Opt and Lin-Kernighan algorithm, with an arbitrary tour T and decreases the cost of it in every iteration. More precisely, in every iteration we perform the following steps:

- Search for a negative cycle C with the Moore-Bellman-Ford algorithm in G_T.
- If no such cycle is found, output T and terminate.
- Else transform C into a closed alternating walk C' in G.
- Modify T by augmenting the closed alternating walk C'.
- If T is not connected after the augmentation step, we compute a minimum spanning tree in a graph where the connected components of T are contracted and add two copies of the spanning tree to T.
- Shortcut T to a tour.

Algorithm 5 Incomplete n-Opt Algorithm

Input: Instance of Graph TSP
Output: Tour T

```
Start with an arbitrary tour T
while there is a negative cycle C in G_T do
    Let C' be the corresponding closed alternating walk of C in G
    T := T △ C'
    if T is not connected then
        Contract the connected components of T in a copy G' of G
        Compute a minimum spanning tree S of G'
        Add two copies of S to T
        Shortcut T to a tour
return T
```

The last steps of the algorithm ensure that after every iteration the algorithm maintains a tour and hence it returns a tour when it terminates.

Lemma 2.6.3. *The cost of the tour decreases after every iteration of the incomplete n-Opt algorithm. Moreover, when the algorithm terminates, every closed alternating walk with h tour edges has at most gain $2h$.*

Proof. Assume that the cycle C we found in G_T has $2h$ edges and total cost $-g < 0$. Then, by the definition of G_T the transformed closed alternating walk C' in G has gain $g+2h$ and h tour edges. After the augmentation of C', the tour has at most h connected components. Hence, we need to add at most $2(h-1)$ edges of cost 1 in the next step to make the graph connected again. Thus, the difference between the old tour and the modified tour is at least $g + 2h - 2(h-1) = g + 2 > 0$.

To show the second statement assume that there is a closed alternating walk with h tour edges and gain greater than $2h$. By construction, this corresponds to a union of cycles in G_T with total cost less than 0. In this case there has to be a cycle with negative cost in G_T that the Moore-Bellman-Ford algorithm would have found. This means that the algorithm would not terminate. □

Lemma 2.6.4. *The incomplete n-Opt algorithm terminates in polynomial time.*

Proof. The Moore-Bellman-Ford algorithm runs in $O(|V(G_T)||E(G_T)|) = O(n^3)$ time. Moreover, the minimum spanning tree can also be computed in polynomial time. Hence, every iteration of the algorithm can be performed in polynomial time. Initially, the cost of the tour is at most $n(n-1)$ since the longest distance in G is at most $n-1$. In every step, the cost decreases by at least 1. Hence, after at most $n(n-1)$ iterations the algorithm has to terminate. □

Lemma 2.6.5. *The approximation ratio of the incomplete n-Opt algorithm is at most 5.*

Proof. Consider the output of the algorithm. We know that we can augment a union of closed alternating walks to obtain the optimal tour. Since the degree of every vertex in a tour is 2, every vertex is incident to at most two tour edges and two non-tour edges. Thus, these closed walks have at most $2n$ tour edges in total. By Lemma 2.6.3, the total gain of these closed walks is at most $4n$. Since the optimal tour consists of at least n edges, it has length at least n. This implies that the approximation ratio is at most 5. □

3 Integrality Ratio of the Subtour LP

3.1 Construction of Instances with Integrality Ratio Converging to $\frac{4}{3}$

In this section we describe a procedure to construct families of instances whose integrality ratios converge to $\frac{4}{3}$. The instances created this way can have a completely different structure than the known instances from the literature.

3.1.1 Construction

We choose a planar embedding of a 2-edge-connected graph G in $\mathbb{R}^2$. For all $k \in \mathbb{N}$ choose a Euclidean TSP instances G_k consisting of the embedded vertices of G and a set of vertices $vw_1, vw_2, \ldots, vw_{l_k}$ in this order subdividing the line segment vw for every edge $\{v, w\} \in E(G)$. Note that the number of subdividing vertices l_k may differ for different edges of G. We call the pairs of vertices of the form vw_i, vw_{i+1} and the pairs v, vw_1 and vw_{l_k}, w *consecutive vertices*. Let δ_k be the greatest distance between two consecutive vertices in G_k. We further require that the instances G_k satisfy the condition $\lim_{k\to\infty} \delta_k = 0$.

Let T_k^* and x_k^* be an optimal tour and optimal fractional tour for G_k, respectively. Moreover, let J be an optimal T-join of the vertices with odd degree in G.

Lemma 3.1.1. *For the optimal fractional tour x_k^* of the instance G_k we have $c(x_k^*) \leq c(E(G))$ for all k.*

Proof. Set $x(e) = 1$ for all edges e connecting two consecutive vertices and $x(e) = 0$ for all other edges e. This is a solution to the LP relaxation of the 2-Edge Connected Spanning Subgraph LP: After the subdivision, the graph stays 2-connected, hence each cut goes through at least two edges and has x-value at least 2. The cost of this solution is exactly the cost of $E(G)$. By Theorem 1.4.13, this is also an upper bound for the optimal solution of the subtour LP. □

Lemma 3.1.2. *For the optimal tour T_k^* of the instance G_k we have $\lim_{k\to\infty} c(T_k^*) \geq c(E(G)) + c(J)$.*

Proof. For $\epsilon_1 > 0$ we construct a new instance G_{k,ϵ_1} from G_k by deleting all subdividing vertices with distance less than ϵ_1 to any vertex $v \in G$. Let U_{k,ϵ_1} be the set of edges connecting two consecutive vertices of G_k where at least one vertex is deleted in G_{k,ϵ_1}. For a vertex $p \in V(G_{k,\epsilon_1})$ and an edge $e \in E(G)$ we say that p *lies* on e if $p \in e$ or p is a vertex subdividing e.

Let ϵ_2 be the shortest distance of two vertices $p, q \in E(G_{k,\epsilon_1})$ that do not lie on a common edge $e \in E(G)$ and $\alpha := \min_{\substack{\{u,w\},\{w,v\}\in E(G)\\ \{u,w\}\neq\{w,v\}}} \angle uwv$ be the smallest angle between two different edges with a common vertex in $E(G)$. We claim that for ϵ_1 fixed we have $\liminf_{k\to\infty} \epsilon_2 > 0$. Since the embedding is planar, we have $\alpha > 0$

and $\liminf_{k\to\infty} \mathrm{dist}_2(p,q) > 0$ if p and q lie on different edges of G not incident to each other. Now, let p and q lie on two different edges $\{u,w\},\{w,v\} \in E(G)$ with a common vertex, respectively. If $\angle uvw > \frac{\pi}{2}$, then $\mathrm{dist}_2(p,q) \geq \epsilon_1$. Else, for p fixed the distance $\mathrm{dist}_2(p,q)$ is minimized if pq is perpendicular to wv. Thus, we have $\mathrm{dist}_2(p,q) \geq \mathrm{dist}_2(p,w)\sin(\angle uwv) \geq \epsilon_1 \sin(\alpha) > 0$ which proves the claim.

Now, consider the subset of edges S_{k,ϵ_1} of an optimal tour T^*_{k,ϵ_1} of G_{k,ϵ_1} consisting of edges not connecting two vertices lying on the same edge. Recall that by definition every edge in S_{k,ϵ_1} has length at least $\epsilon_2 > 0$. Now, if $|S_{k,\epsilon_1}| > \frac{c(E(G))+c(J)}{\epsilon_2}$, we have $c(T^*_k) \geq c(T^*_{k,\epsilon_1}) > c(E(G)) + c(J)$ since the vertices of G_{k,ϵ_1} is a subset of that of G_k. It remains the case that $|S_{k,\epsilon_1}| \leq \frac{c(E(G))+c(J)}{\epsilon_2}$. Note that by Lemma 1.4.14 the optimal tour is a simple polygon. Thus, the edges in $T^*_{k,\epsilon_1}\backslash S_{k,\epsilon_1}$ are connecting consecutive vertices and we have

$$\begin{aligned} c(T^*_{k,\epsilon_1}\backslash S_{k,\epsilon_1}) &= c(E(G_{k,\epsilon_1}) \cap T_{k,\epsilon_1}) = c(E(G_{k,\epsilon_1})) - c(E(G_{k,\epsilon_1})\backslash T_{k,\epsilon_1}) \\ &\geq c(E(G_{k,\epsilon_1})) - |S_{k,\epsilon_1}|\delta_k = c(E(G)) - |S_{k,\epsilon_1}|\delta_k - c(U_{k,\epsilon_1}) \\ &\geq c(E(G)) - |S_{k,\epsilon_1}|\delta_k - 2|E(G)|(\epsilon_1+\delta_k) \end{aligned}$$

since $c(U_{k,\epsilon_1}) \leq 2|E(G)|(\epsilon_1+\delta_k)$. Furthermore, the edges in S_{k,ϵ_1} are a T-join for the vertices with odd degree in $(V(G_{k,\epsilon_1}), T^*_{k,\epsilon_1}\backslash S_{k,\epsilon_1})$. Therefore, $S_{k,\epsilon_1} \cup (E(G_{k,\epsilon_1})\backslash T_{k,\epsilon_1}) \cup U_{k,\epsilon_1}$ is a T-join for the vertices with odd degree in G_k as $E(G_k) = (T^*_{k,\epsilon_1}\backslash S_{k,\epsilon_1}) \cup (E(G_{k,\epsilon_1})\backslash T_{k,\epsilon_1}) \cup U_{k,\epsilon_1}$. Since G has the same set of vertices with odd degree as G_k, this is also a T-join for the vertices with odd degrees in G. Thus,

$$c(S_{k,\epsilon_1}) \geq c(J) - c(E(G_{k,\epsilon_1})\backslash T_{k,\epsilon_1}) - c(U_{k,\epsilon_1}) \geq c(J) - |S_{k,\epsilon_1}|\delta_k - 2|E(G)|(\epsilon_1+\delta_k).$$

Altogether, for the total length of the tour we have:

$$\begin{aligned} \lim_{k\to\infty} c(T^*_k) &\geq \lim_{\epsilon_1\to 0}\lim_{k\to\infty} c(T^*_{k,\epsilon_1}) = \lim_{\epsilon_1\to 0}\lim_{k\to\infty} c(T^*_{k,\epsilon_1}\backslash S_{k,\epsilon_1}) + c(S_{k,\epsilon_1}) \\ &\geq \lim_{\epsilon_1\to 0}\lim_{k\to\infty} c(E(G)) - |S_{k,\epsilon_1}|\delta_k - 2|E(G)|(\epsilon_1+\delta_k) \\ &\quad + c(J) - |S_{k,\epsilon_1}|\delta_k - 2|E(G)|(\epsilon_1+\delta_k) \\ &\geq \lim_{\epsilon_1\to 0}\lim_{k\to\infty} c(E(G)) - 2\frac{c(E(G))+c(J)}{\epsilon_2}\delta_k - 4|E(G)|(\epsilon_1+\delta_k) + c(J) \\ &= \lim_{\epsilon_1\to 0} c(E(G)) - 4|E(G)|\epsilon_1 + c(J) = c(E(G)) + c(J). \end{aligned}$$ □

Hence, we conclude:

Theorem 3.1.3. *The integrality ratios of the family of instances G_k converge as $k \to \infty$ to at least $\frac{c(E(G))+c(J)}{c(E(G))}$, where J is the cost of the optimal T-join of the vertices with odd degree in G.*

Remark 3.1.4. It is not possible to construct instances with higher integrality ratio than $\frac{4}{3}$ using this procedure. Let a planar embedding of a 2-connected graph G be given. Consider the vector $y \in \mathbb{R}^{E(G)}$ with $y(e) = \frac{1}{3}$ for every $e \in E(G)$. Since the graph is 2-edge-connected, every cut S that intersets an odd number of edges intersects $E(G)$ at least three times. Hence, for every odd cut S the total y-value of $\delta(S)$ is at least $3 \cdot \frac{1}{3} = 1$. Therefore, y lies in the T-join polytope and has cost $\frac{c(E(G))}{3}$. Thus, we have $c(J) \leq \frac{c(E(G))}{3}$. A similar construction of a vector in the T-join polytope was already used in [50].

In the next section we will see concrete examples of G_k whose integrality ratios converge to $\frac{4}{3}$.

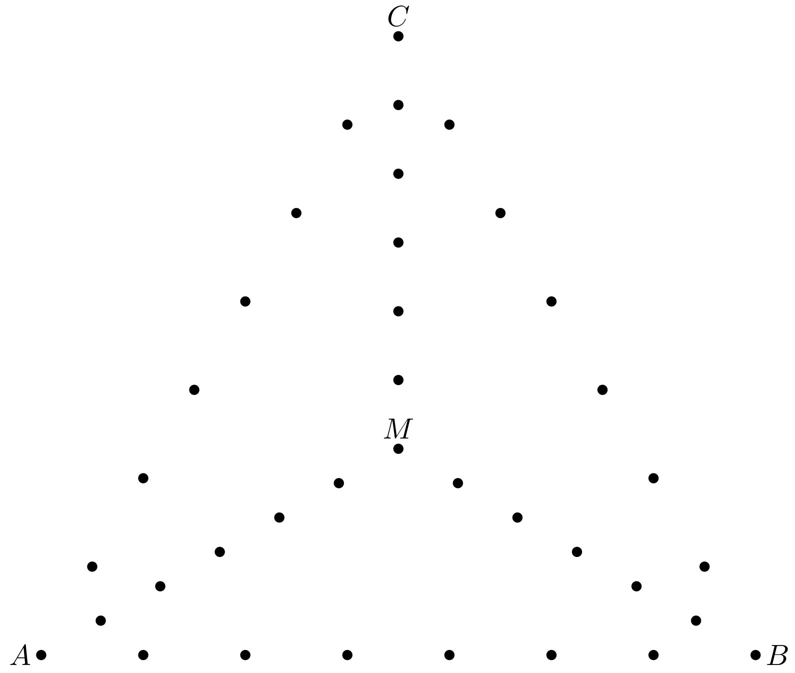


Figure 3.1: The tetrahedron instance: The graph G is a complete graph consisting of the vertices A, B, C, M. In the figure the edges are subdivided equidistantly by $a_k = 6$ and $b_k = 5$ vertices, respectively.

3.1.2 Applications

Now, we apply the results of the last section to construct families of instances whose integrality ratios converge to $\frac{4}{3}$.

The *tetrahedron instances* were already introduced in [38]. It consists of the vertices A, B, C forming an equilateral triangle with the center M. The sides of the triangle AB, BC, CA and the segments MA, MB, MC are subdivided equidistantly by a_k and b_k equidistant vertices, respectively (Figure 3.1). Moreover, we have $a_k, b_k \to \infty$ as $k \to \infty$. We can apply Theorem 3.1.3 to get another proof of Theorem 3.19 in [38], that the integrality ratio of this family converges to $\frac{4}{3}$ as $k \to \infty$: We take a complete graph K_4 and embed it to the Euclidean plane such that the vertices coincide with the vertices A, B, C, M of the instance. Now, every vertex has an odd degree in G. Hence, a T-join has to correct the parity of every vertex and the cost of the T-join is at least $\text{dist}(A, B) + \text{dist}(C, M)$ which is by symmetry $\frac{c(E(G))}{3}$.

Another example are the *hexagon instances*. In contrast to the tetrahedron instances we do not uniquely define the graph G we subdivide. We only require that the vertices of G form small regular hexagons that tesselate a subset of $\mathbb{R}^2$ (Figure 3.2). Every vertex of G not lying on the border of the tesselation has odd degree. Thus, every T-join is incident to each of these vertices. The cost of each path in the T-join can be bounded by the shortest distance between two distinct vertices, which is a side length of the hexagons. Hence, if we tesselate a subset of $\mathbb{R}^2$ where the number of vertices on the border is small compared to the total number of vertices, the cost of the T-join is at least roughly $\frac{c(E(G))}{3}$. We can take growing tesselations and subdivide the edges to get instances with integrality ratio converging to $\frac{4}{3}$.

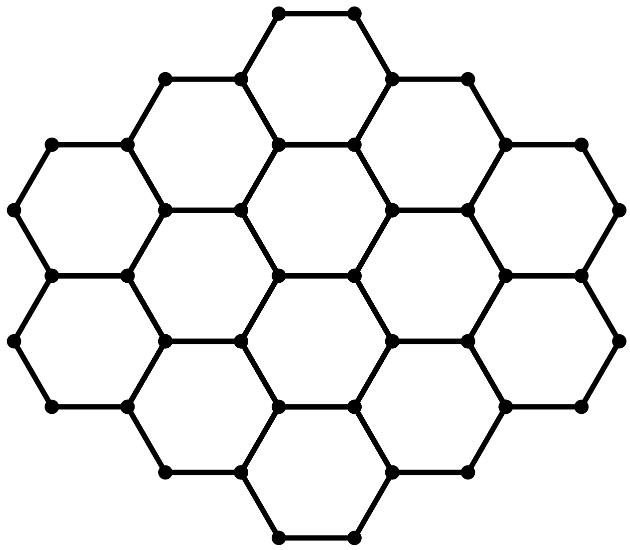


Figure 3.2: A possible graph G for the hexagon instances. One possible construction of the instances G_k is to subdivide every edge by k equidistant vertices.

3.2 Computing the Exact Integrality Ratio for Rectilinear TSP

In this section we compute the exact integrality ratio for Rectilinear TSP instances with a small fixed number of vertices. For that we used Sylvia Boyd's list of the extremal points of the subtour polytope published on her homepage [12]. This approach was already used to compute the exact integrality ratio of Metric TSP instances with $6 \le n \le 12$ [6, 14].

Let $\mathscr{T}$ be the set of all tours for the vertex set $V(K_n)$. Given an extremal point x of the subtour polytope we solve the corresponding program given by

$$\max f \tag{3.1}$$

$$\sum_{\{u,v\}\in T} |u_x - v_x| + |u_y - v_y| \ge f \qquad \text{for all } T \in \mathscr{T} \tag{3.2}$$

$$\sum_{u,v\in K_n} (|u_x - v_x| + |u_y - v_y|) \cdot x(\{u,v\}) \le 1 \tag{3.3}$$

$$u_x, u_y, f \in \mathbb{R} \qquad \text{for all } u \in V(K_n) \tag{3.4}$$

The variables u_x and u_y for all $u \in V(K_n)$ represent the x- and y-coordinate of the vertex u and f represents the integrality ratio of the instance.

On the one hand, each of the programs gives a lower bound for the integrality ratio since condition (3.2) ensures that the length of the optimal tour is at least f and condition (3.3) ensures that the cost of the optimal fractional tour is at most 1. Therefore, their ratio is at least f. On the other hand, the maximum value of the programs for all extremal points x of the subtour LP is the maximum achievable integrality ratio, since the instance achieving the maximal integrality ratio can be scaled such that the optimal fractional tour has length 1. Thus, the largest f of all programs corresponding to all extremal points with a fixed number of vertices is the exact integrality ratio.

For performance reasons we added the constraints $0 \le f \le 2$ and $0 \le u_x, u_y \le 1$. This can be done without loss of generality since by Wolsey's analysis f lies between 1 and $\frac{3}{2}$ [68]. Moreover, every instance with a fractional tour of cost at most 1 fits into the unit square since the fractional tour intersects every cut at least twice. Therefore, by translating this instance we can assume that it lies in the unit square.

Note that it is known that for $n \leq 5$ every optimal fractional tour is also integral and thus the integrality ratio is 1. For $6 \leq n \leq 10$ vertices and every extremal point x of the subtour polytope the corresponding program has been solved using `Gurobi 8.11`. It took about a month of computation time to solve all 461 programs for the case $n = 10$ with an `Intel i5-4670`. On the same machine a month of computation time was not sufficient to solve a single program for $n = 11$.

The resulting instances are shown in Figure 3.3. They have a similar structure as the instances maximizing the integrality ratio in the metric case. Their explicit coordinates will be given in Subsection 3.3.3 where we generalize these instances to higher n.

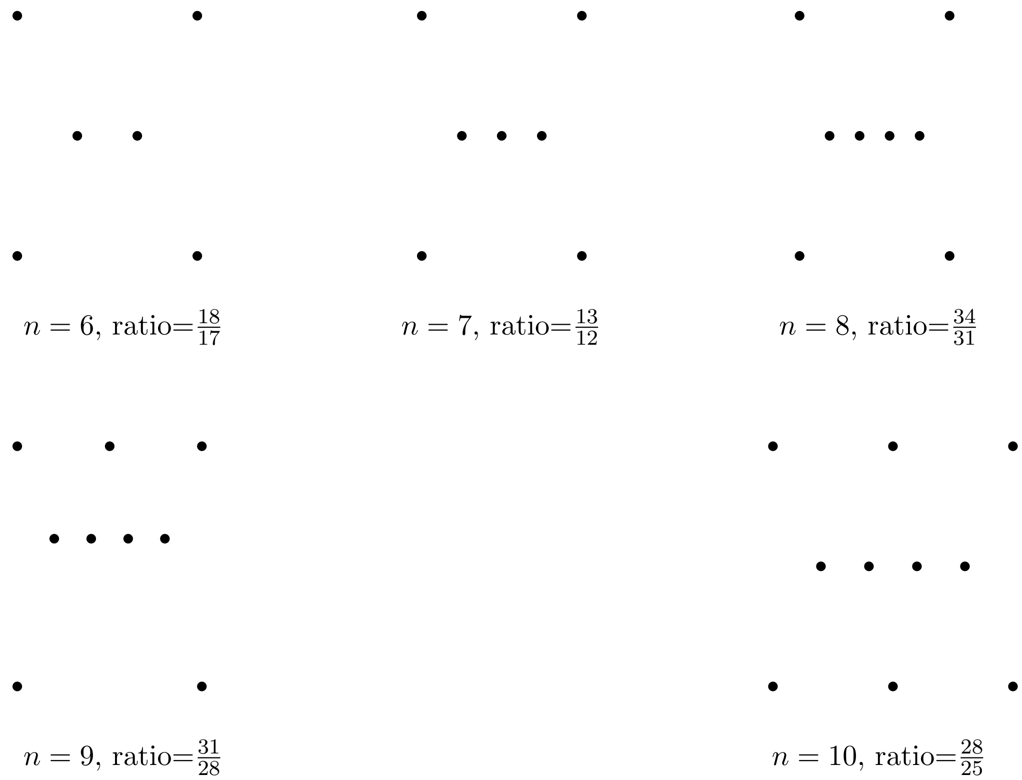


Figure 3.3: Instances with n vertices maximizing the integrality ratio for Rectilinear TSP.

We see that vertices are not distributed equally on the three lines as in the metric case. Here we get a higher integrality ratio if the number of vertices is higher on the center line. In the following sections we will investigate this phenomenon and other structural properties of this family of instances.

3.3 Integrality Ratio for Rectilinear TSP

In this section we construct a family of Rectilinear TSP instances with similar structure and properties as the results of the computations from Section 3.2 and analyze their integrality ratio.

3.3.1 Structure of the Fractional Tours

The optimal fractional tours of the instances maximizing the integrality ratio found in Section 3.2 have the same form. In this section we describe and extend it to higher number of vertices.

Note that every fractional tour x can be interpreted as a weighted complete graph where the vertex set consists of the vertices of the instance and the weight of the edge e is equal to $x(e)$. We define the weighted graphs $x_{i,j,k}$ for nonnegative integers i, j, k as follows: The vertex set $V(x_{i,j,k}) := \{X_0, \ldots, X_{i+1}, Y_0, \ldots, Y_{j+1}, Z_0, \ldots, Z_{k+1}\}$ consists of $i+j+k+6$ vertices. We set the weight of the edges $\{X_r, X_{r+1}\}, \{Y_s, Y_{s+1}\}, \{Z_t, Z_{t+1}\}$ for all $r \in \{0, \ldots, i\}, s \in \{0, \ldots, j\}, t \in \{0, \ldots, k\}$ to 1. Moreover, we set the weight of the edges $\{X_0, Y_0\}, \{X_0, Z_0\}$, $\{Y_0, Z_0\}$, $\{X_{i+1}, Y_{j+1}\}, \{X_{i+1}, Z_{k+1}\}$ and $\{Y_{j+1}, Z_{k+1}\}$ to $\frac{1}{2}$. All other edges have weight 0 (Figure 3.4, the instance $I_{2,2,1}$ will be defined later).

The optimal fractional tours of the instances maximizing the integrality ratio found in Section 3.2 are isomorphic to $x_{i,j,k}$ for some i, j, k.

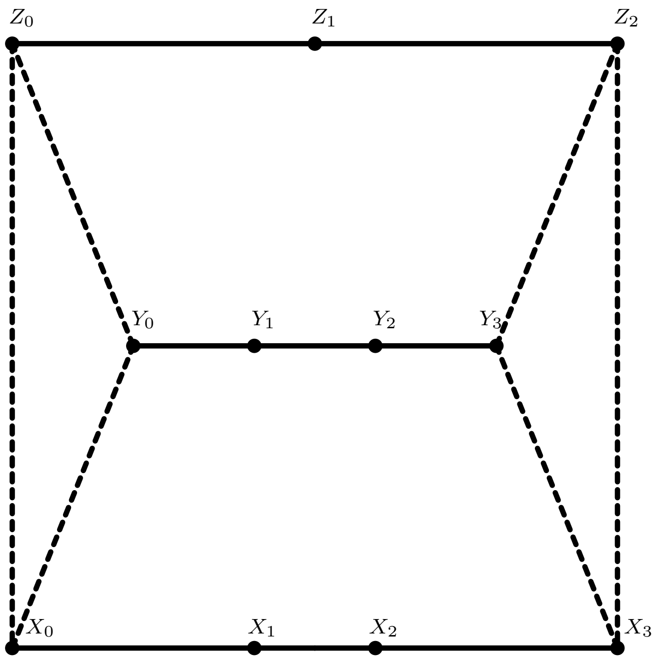


Figure 3.4: The instance $I_{2,2,1}$ with optimal fractional tour $x_{2,2,1}$. The straight and dashed edges have weights 1 and $\frac{1}{2}$ in $x_{2,2,1}$, respectively.

3.3.2 Structure of the Optimal Tours

In this subsection we describe the structure of the optimal tours of the instances maximizing the integrality ratio computed in Section 3.2 which is the motivation for the construction of the generalized instances $I^2_{i,j,k}$ in the next subsection.

A *pseudo-tour* of a TSP instance is a closed walk that visits every vertex at least once. We first define a set of pseudo-tours $\mathfrak{T}$ which is the union of three sets of pseudo-tours $T^{\uparrow}, T^{\circ}, T^{\downarrow}$ and the pseudo-tours $T^{\nwarrow}, T^{\nearrow}, T^{\leftarrow}, T^{\rightarrow}, T^{\swarrow}, T^{\searrow}$.

Let $T^{\uparrow} := \{T^{\uparrow}_0, \ldots, T^{\uparrow}_k\}$ be a set of pseudo-tours where the pseudo-tour $T^{\uparrow}_l$ for some $0 \le l \le k$ consists of (Figure 3.5):

- two copies of the edges $\{Z_s, Z_{s+1}\}$ for all $0 \leq s \leq k, s \neq l$
- a copy of the edges $\{Y_s, Y_{s+1}\}$ for all $0 \leq s \leq j$
- a copy of the edges $\{X_s, X_{s+1}\}$ for all $0 \leq s \leq i$
- a copy of the edges $\{Z_0, Y_0\}$, $\{Z_0, X_0\}$, $\{Z_{k+1}, Y_{j+1}\}$ and $\{Z_{k+1}, X_{i+1}\}$

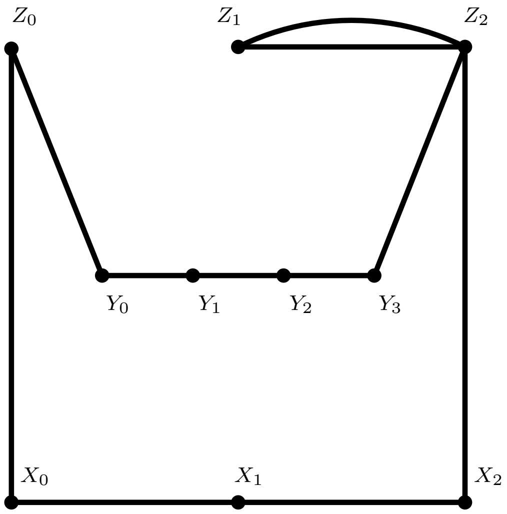


Figure 3.5: The pseudo-tour $T_0^{\uparrow}$.

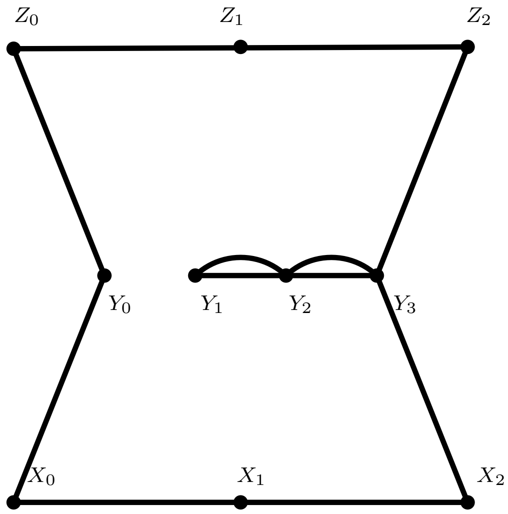


Figure 3.6: The pseudo-tour T_0°.

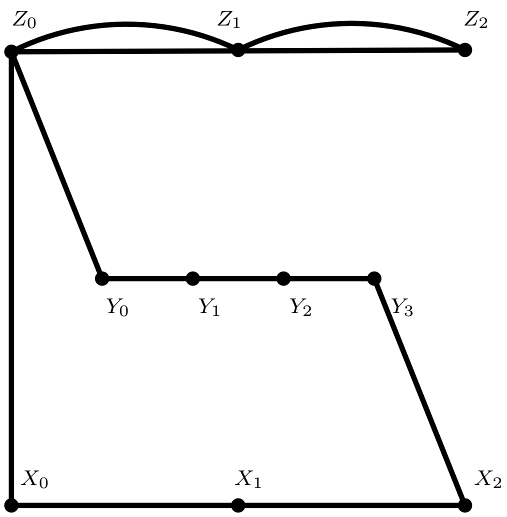


Figure 3.7: The pseudo-tour $T^{\nwarrow}$.

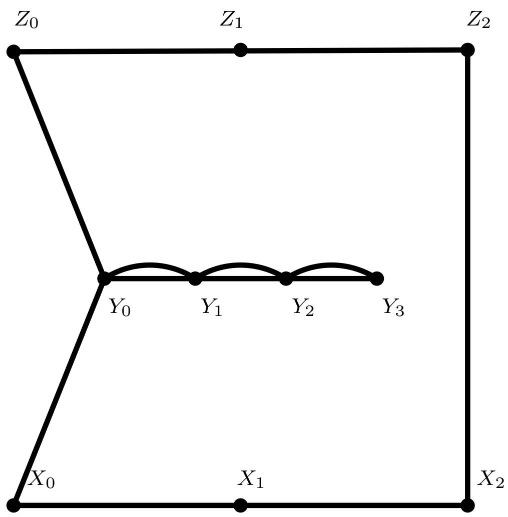


Figure 3.8: The pseudo-tour $T^{\leftarrow}$.

We also define the sets of pseudo-tours $T^{\circ} := \{T_0^{\circ}, \ldots, T_j^{\circ}\}$ and $T^{\downarrow} := \{T_0^{\downarrow}, \ldots, T_i^{\downarrow}\}$. Each of the tours T_l° (Figure 3.6) and $T_l^{\downarrow}$ are defined similarly: Instead of $\{Z_s, Z_{s+1}\}$ we take two copies of the edges $\{Y_s, Y_{s+1}\}$ and $\{X_s, X_{s+1}\}$ except $\{Y_l, Y_{l+1}\}$ and $\{X_l, X_{l+1}\}$, respectively.

The tour $T^{\nwarrow}$ consists of (Figure 3.7):

- two copies of the edges $\{Z_s, Z_{s+1}\}$ for all $0 \leq s \leq k$
- a copy of the edges $\{Y_s, Y_{s+1}\}$ for all $0 \leq s \leq j$
- a copy of the edges $\{X_s, X_{s+1}\}$ for all $0 \leq s \leq i$
- a copy of the edges $\{Z_0, Y_0\}$, $\{Z_0, X_0\}$ and $\{Y_{j+1}, X_{i+1}\}$

The pseudo-tour $T^{\nearrow}$ is defined similarly. Instead of the edges $\{Z_0, Y_0\}$, $\{Z_0, X_0\}$ and $\{Y_{j+1}, X_{i+1}\}$ it consists of the edges $\{Z_{k+1}, Y_{j+1}\}$, $\{Z_{k+1}, X_{i+1}\}$ and $\{Y_0, X_0\}$. We also define the pseudo-tours $T^{\leftarrow}$ (Figure 3.8), $T^{\rightarrow}, T^{\swarrow}, T^{\searrow}$ similarly where we double the edges $\{Y_s, Y_{s+1}\}$ or $\{X_s, X_{s+1}\}$ instead of $\{Z_s, Z_{s+1}\}$.

We observe that the optimal tours of the instances maximizing the integrality ratio for $6 \leq n \leq 10$ computed in Section 3.2 are the non-intersecting shortcuts of the pseudo-tours in $\mathfrak{T}$. In the next subsection the instances $I^2_{i,j,k}$ are constructed such that the vertices lie on three lines, are symmetric and the optimal tours are the non-intersection shortcuts of pseudo-tours in $\mathfrak{T}$. Similar properties also hold for other variants of the TSP. For example, the shortcuts of the pseudo-tours in $\mathfrak{T}$ are the optimal tours of the instances maximizing the integrality ratio in the metric case for $6 \leq n \leq 12$ given in [6].

3.3.3 The Instance $I^2_{i,j,k}$

We define an embedding of $x_{i,j,k}$ in $\mathbb{R}^2$ (Figure 3.4) as follows: The vertices $\{X_0, \ldots, X_{i+1}\}$, $\{Y_0, \ldots, Y_{j+1}\}$ and $\{Z_0, \ldots, Z_{k+1}\}$ lie on the three parallel lines l_1, l_2 and l_3, respectively. The line l_2 lies between l_1 and l_3 in the plane. Moreover, l_1, l_2 and l_2, l_3 have distances $b_1 := \frac{1}{2} + \frac{j+1}{j+3}\left(\frac{1}{k+1} - \frac{1}{2}\right)$ and $b_2 := \frac{1}{2} + \frac{j+1}{j+3}\left(\frac{1}{i+1} - \frac{1}{2}\right)$ to each other, respectively. The vertices X_0, X_{i+1}, Z_0 and Z_{k+1} form an axis-parallel rectangle with side lengths 1 and $1 + \frac{j+1}{j+3}\left(\frac{1}{i+1} + \frac{1}{k+1} - 1\right)$. Let Y'_l and Z'_l be the orthogonal projection of Y_l and Z_l to the line X_0X_{i+1}, respectively. We call a sequence of points $v_1, \ldots, v_s$ an *equidistant progression* if $\text{dist}_1(v_l, v_{l+1}) = \text{dist}_1(v_1, v_2)$ for all $1 \leq l \leq s-1$. Then $X_0, Y'_0, Y'_1, \ldots, Y'_{j+1}, X_{i+1}$ and $Y'_0, X_1, X_2, \ldots, X_i, Y'_{j+1}$ and $Y'_0, Z'_1, Z'_2, \ldots, Z'_k, Y'_{j+1}$ are three equidistant progressions.

Note that this embedding also defines a Rectilinear TSP instance we call $I^2_{i,j,k}$. The explicit coordinates of the vertices are given by:

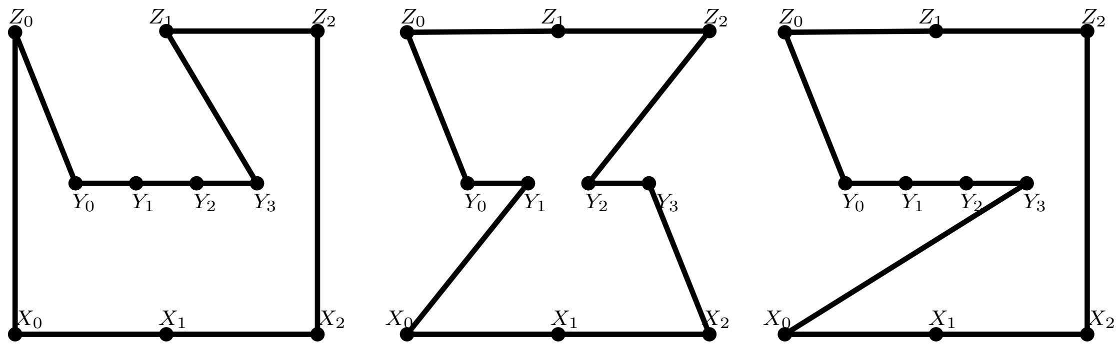
$$\begin{aligned}
X_0 &= (0,0) \\
X_{i+1} &= (1,0) \\
Z_0 &= \left(0, 1 + \frac{j+1}{j+3}\left(\frac{1}{i+1} + \frac{1}{k+1} - 1\right)\right) \\
Z_{k+1} &= \left(1, 1 + \frac{j+1}{j+3}\left(\frac{1}{i+1} + \frac{1}{k+1} - 1\right)\right) \\
X_s &= \left(s \cdot \frac{j+1}{j+3} \cdot \frac{1}{i+1} + \frac{1}{j+3}, 0\right) && \forall 1 \leq s \leq i, \\
Y_s &= \left(\frac{(s+1)}{j+3}, \frac{1}{2} + \frac{j+1}{j+3}\left(\frac{1}{k+1} - \frac{1}{2}\right)\right) && \forall 0 \leq s \leq j+1, \\
Z_s &= \left(s \cdot \frac{j+1}{j+3} \cdot \frac{1}{k+1} + \frac{1}{j+3}, 1 + \frac{j+1}{j+3}\left(\frac{1}{i+1} + \frac{1}{k+1} - 1\right)\right) && \forall 1 \leq s \leq k.
\end{aligned}$$

n	Instance	Opt. frac. tour	Integrality ratio
6	$I_{0,0,0}$	$x_{0,0,0}$	$\frac{18}{17} \approx 1.059$
7	$I_{0,1,0}$	$x_{0,1,0}$	$\frac{13}{12} \approx 1.083$
8	$I_{0,1,1}$	$x_{0,1,1}$	$\frac{34}{31} \approx 1.097$
9	$I_{0,2,1}$	$x_{0,2,1}$	$\frac{31}{28} \approx 1.107$
10	$I_{1,2,1}$	$x_{1,2,1}$	$\frac{28}{25} = 1.120$

Table 3.1: The instances maximizing the integrality ratio computed in Section 3.2 with their corresponding fractional optimal tours and integrality ratios.

We call the vertices $X_0, \ldots, X_{i+1}, Z_0, \ldots, Z_{k+1}$ the *outer vertices* and $Y_0, \ldots, Y_{j+1}$ the *inner vertices*.

Revising the instances that maximize the integrality ratio from Section 3.2 we see that they can be transformed to $I^2_{i,j,k}$ for some i, j and k by scaling, rotating and translating. Moreover, their optimal fractional tours are isomorphic to $x_{i,j,k}$ and their optimal tours are the non-intersecting shortcuts of the pseudo-tours in $\mathfrak{T}$ (Figure 3.9). The values of i, j, k and the corresponding integrality ratios are listed in Table 3.1.



The non-intersecting shortcut of $T^{\uparrow}_1$ for $I_{1,2,1}$.

A non-intersecting shortcut of T°_2 for $I_{1,2,1}$. All shortcuts of T°_2 are non-intersecting.

A non-intersecting shortcut of $T^{\leftarrow}$ for $I_{1,2,1}$. All shortcuts of $T^{\leftarrow}$ are non-intersecting.

Figure 3.9: Some of the non-intersecting shortcuts of pseudo-tours in $\mathfrak{T}$.

We observe that for $I^2_{i,j,k}$ there is a unique non-intersecting shortcut of pseudo-tours in $T^{\uparrow}$ and $T^{\downarrow}$ while all shortcuts of pseudo-tours in $T^{\circ}, T^{\leftarrow}, T^{\rightarrow}$ are non-intersecting. Moreover, the non-intersecting shortcuts of $T^{\nwarrow}, T^{\nearrow}, T^{\swarrow}$ and $T^{\searrow}$ are also non-intersecting shortcuts of $T^{\leftarrow}$ and $T^{\rightarrow}$. Hence, the set of optimal tours are the non-intersecting shortcuts of $T^{\uparrow}, T^{\downarrow}$ and all shortcuts of $T^{\circ}, T^{\leftarrow}, T^{\rightarrow}$.

For n fixed let the instance $I^2_n := I^2_{i^*,j^*,k^*}$ maximize the integrality ratio among all instances $I_{i,j,k}$ with $i + j + k + 6 = n$.

3.3.4 Length of the Optimal Tours for $I^2_{i,j,k}$

In this subsection we determine the length of the optimal tours for $I^2_{i,j,k}$. This will be used to compute lower bounds on the integrality ratio of the instances in the next subsection.

A *subpath* of an oriented tour consists of vertices $v_1, \dots, v_l$, such that v_{i+1} is visited by the tour immediately after v_i for all $i = 1, \dots, l-1$. A subpath of a tour starting and ending at outer vertices and containing no other outer vertex is called a *trip* if it contains at least one inner vertex.

By Lemma 1.4.15, we know that each optimal tour of $I^2_{i,j,k}$ can be decomposed into a set of trips and a set of edges connecting consecutive outer vertices such that all inner vertices are contained in some trip and two different trips intersect in at most one outer vertex.

Lemma 3.3.1. *The length of the optimal tour for $I^2_{i,j,k}$ is $4 + 2b_1 + 2b_2 - \frac{2}{j+3}$.*

Proof. Assume that we have given an optimal tour. Since all inner vertices lie on a line, every trip visits a set of consecutive inner vertices. We start with a cycle visiting the outer vertices in cyclic order. This cycle has length

$$\mathrm{dist}_1(X_0, X_{i+1}) + \mathrm{dist}_1(X_{i+1}, Z_{k+1}) + \mathrm{dist}_1(Z_0, Z_{k+1}) + \mathrm{dist}_1(X_0, Z_0) = 2 + 2b_1 + 2b_2$$

Now, we successively replace an edge between two consecutive outer vertices by a trip until every inner vertex is visited and we get the given optimal tour.

If we replace $\{Z_r, Z_{r+1}\}$ for some $1 \leq r \leq k-1$ by a trip visiting the inner vertices $Y_s, Y_{s+1}, \dots, Y_t$, the cost of the cycle increases by at least

$$\begin{aligned}
&\mathrm{dist}_1(Z_r, Y_s) + \mathrm{dist}_1(Y_s, Y_t) + \mathrm{dist}_1(Y_t, Z_{r+1}) - \mathrm{dist}_1(Z_r, Z_{r+1})\\
\geq& 2\,\mathrm{dist}_1(Y_s, Y_t) - 2\,\mathrm{dist}_1(Z_r, Z_{r+1}) + 2b_2\\
=& 2\,\mathrm{dist}_1(Y_s, Y_t) - 2 \cdot \frac{j+1}{j+3} \cdot \frac{1}{k+1} + 1 + \frac{j+1}{j+3}\left(\frac{2}{k+1} - 1\right) = 2\,\mathrm{dist}_1(Y_s, Y_t) + \frac{2}{j+3}
\end{aligned}$$

Similarly, the cost increases by at least the same amount when we replace $\{X_r, X_{r+1}\}$ for some $1 \leq r \leq k-1$ by a trip.

In the case where we replace $\{Z_0, Z_1\}$ by a trip visiting the inner vertices $Y_s, Y_{s+1}, \dots, Y_t$ let Y_0' be the orthogonal projection of Y_0 to the line $Z_0 Z_{k+1}$. The cost of the cycle increases by at least

$$\begin{aligned}
&\mathrm{dist}_1(Z_0, Y_s) + \mathrm{dist}_1(Y_s, Y_t) + \mathrm{dist}_1(Y_t, Z_1) - \mathrm{dist}_1(Z_0, Z_1)\\
\geq& 2\,\mathrm{dist}_1(Y_s, Y_t) - 2\,\mathrm{dist}_1(Y_0', Z_1) + 2b_2\\
=& 2\,\mathrm{dist}_1(Y_s, Y_t) - 2 \cdot \frac{j+1}{j+3} \cdot \frac{1}{k+1} + 1 + \frac{j+1}{j+3}\left(\frac{2}{k+1} - 1\right) = 2\,\mathrm{dist}_1(Y_s, Y_t) + \frac{2}{j+3}
\end{aligned}$$

Similarly, the cost increases by at least the same amount when we replace one of the edges $\{Z_k, Z_{k+1}\}$, $\{X_0, X_1\}$ or $\{X_i, X_{i+1}\}$ by a trip.

If we replace $\{X_0, Z_0\}$ by a trip visiting the vertices $Y_s, Y_{s+1}, \dots, Y_t$, the cost of the cycle increases by at least:

$$\begin{aligned}
&\operatorname{dist}_1(X_0, Y_s) + \operatorname{dist}_1(Y_s, Y_t) + \operatorname{dist}_1(Y_t, Z_0) - \operatorname{dist}_1(X_0, Z_0) \\
=&\operatorname{dist}_1(X_0, Y_t) + \operatorname{dist}_1(Z_0, Y_t) - \operatorname{dist}_1(X_0, Z_0) \\
=&(t+1)\frac{1}{j+3} + b_1 + (t+1)\frac{1}{j+3} + b_2 - (b_1 + b_2) = 2(t+1)\frac{1}{j+3} \\
\geq& 2\operatorname{dist}(Y_s, Y_t) + \frac{2}{j+3}
\end{aligned}$$

Similarly, we get the same value when we replace $\{X_{i+1}, Z_{k+1}\}$ by a trip. Assume that the optimal tour has exactly w trips $t_1, \ldots, t_w$. Since all the inner vertices are visited by the w trips, all except $w-1$ edges of the form $\{Y_s, Y_{s+1}\}$ for $0 \leq s \leq j$ are contained in the trips. Hence the total length of the optimal tour is at least:

$$\begin{aligned}
&2 + 2b_1 + 2b_2 + 2\operatorname{dist}(Y_0, Y_{j+1}) - (w-1) \cdot 2 \cdot \frac{1}{j+3} + w\frac{2}{j+3} \\
=&2 + 2b_1 + 2b_2 + 2 - \frac{4}{j+3} + \frac{2}{j+3} = 4 + 2b_1 + 2b_2 - \frac{2}{j+3}
\end{aligned}$$

In fact a straightforward calculation shows that all non-intersecting shortcuts of $\mathfrak{T}$ are optimal tours. For example the non-intersecting shortcut of the tour $T^{\leftarrow}$ has length:

$$\begin{aligned}
&\operatorname{dist}_1(X_0, Y_{j+1}) + \operatorname{dist}_1(Y_0, Y_{j+1}) + \operatorname{dist}_1(Y_0, Z_0) + \operatorname{dist}_1(Z_0, Z_{k+1}) + \operatorname{dist}_1(Z_{k+1}, X_{i+1}) \\
+&\operatorname{dist}_1(X_{i+1}, X_0) = (b_1 + 1 - \frac{1}{j+3}) + (1 - 2 \cdot \frac{1}{j+3}) + (b_2 + \frac{1}{j+3}) + 1 + (b_1 + b_2) + 1 \\
=&4 + 2b_1 + 2b_2 - \frac{2}{j+3}
\end{aligned}$$

Hence, the lower bound on the length of the optimal tour is tight. □

3.3.5 The Integrality Ratio of $I^2_{i,j,k}$

In this section we investigate the integrality ratio of $I^2_{i,j,k}$. Recall that I^2_n is defined as the instance of the form $I_{i,j,k}$ with n vertices and maximal integrality ratio.

Theorem 3.3.2. *The integrality ratio of $I^2_{i,j,k}$ is at least $1 + \frac{1}{3+2\left(\frac{5}{j+1}+\frac{1}{k+1}+\frac{1}{i+1}\right)}$. In particular, the integrality ratios of instances I^2_n converge to $\frac{4}{3}$ as $n \to \infty$.*

Proof. The cost of the optimal fractional tour of $I_{i,j,k}$ is at most the cost of $x_{i,j,k}$ which is

$$\begin{aligned}
&\operatorname{dist}_1(X_0, X_{i+1}) + \operatorname{dist}_1(Y_0, Y_{j+1}) + \operatorname{dist}_1(Z_0, Z_{k+1}) + \frac{1}{2}\operatorname{dist}_1(X_0, Y_0) + \frac{1}{2}\operatorname{dist}_1(X_0, Z_0) \\
+&\frac{1}{2}\operatorname{dist}_1(Y_0, Z_0) + \frac{1}{2}\operatorname{dist}_1(X_{i+1}, Y_{j+1}) + \frac{1}{2}\operatorname{dist}_1(X_{i+1}, Z_{k+1}) + \frac{1}{2}\operatorname{dist}_1(Y_{j+1}, Z_{k+1}) \\
=&3 + 2b_1 + 2b_2.
\end{aligned}$$

By Lemma 3.3.1, the cost of the optimal tour of $I_{i,j,k}$ is $4 + 2b_1 + 2b_2 - \frac{2}{j+3}$. Hence, the integrality ratio is at least

$$\begin{aligned}
&\frac{4+2b_1+2b_2-\frac{2}{j+3}}{3+2b_1+2b_2} = 1+\frac{1-\frac{2}{j+3}}{3+2b_1+2b_2}\\
=&1+\frac{1-\frac{2}{j+3}}{3+1+\frac{j+1}{j+3}\left(\frac{2}{k+1}-1\right)+1+\frac{j+1}{j+3}\left(\frac{2}{i+1}-1\right)}\\
=&1+\frac{j+3-2}{5(j+3)+(j+1)\left(\frac{2}{k+1}+\frac{2}{i+1}-2\right)} = 1+\frac{j+1}{3j+13+(j+1)\left(\frac{2}{k+1}+\frac{2}{i+1}\right)}\\
=&1+\frac{1}{3+2\left(\frac{5}{j+1}+\frac{1}{k+1}+\frac{1}{i+1}\right)}
\end{aligned}$$

To get the highest integrality ratio of the instances $I^2_{i,j,k}$ we have to find $\min_{i,j,k} \frac{5}{j+1} + \frac{1}{i+1} + \frac{1}{k+1}$ where $i+j+k = n-6$ is fixed.

We get the following estimate by the Cauchy-Schwarz inequality:

$$\begin{aligned}
\left(\frac{5}{j+1}+\frac{1}{i+1}+\frac{1}{k+1}\right)(n-3) &= \left(\frac{5}{j+1}+\frac{1}{i+1}+\frac{1}{k+1}\right)(j+1+i+1+k+1)\\
&\geq (\sqrt{5}+\sqrt{1}+\sqrt{1})^2 = (\sqrt{5}+2)^2
\end{aligned}$$

With equality if and only if $\frac{5}{(j+1)^2} = \frac{1}{(i+1)^2} = \frac{1}{(k+1)^2}$. In this case the actual integrality ratio is

$$1+\frac{1}{3+2\frac{(\sqrt{5}+2)^2}{n-3}} \in \frac{4}{3} - \Theta(n^{-1})$$

We see that this expression converges to $\frac{4}{3}$ as $n \to \infty$. However, the optimal values of i, j and k we chose above do not have to be integral. So in order to conclude that the actual integrality ratios of the instance $I^2_n = I^2_{i,j,k}$ for the best choice of i, j, k also converge to $\frac{4}{3}$ we need to show that the optimal integral values of i, j, k achieve a similar integrality ratio. Let i', j' and k' be real numbers satisfy $\frac{5}{(j')^2} = \frac{1}{(i')^2} = \frac{1}{(k'+1)^2}$ and $i'+j'+k' = n-6$. With the restriction that i, j and k are integers we can choose $i = \lfloor i' \rfloor, j = \lfloor j' \rfloor$ and $k = \lceil k' \rceil$ or $k = \lceil k' \rceil + 1$ such that $i+j+k = i'+j'+k' = n-6$. In this case we have by a similar application of the Cauchy-Schwarz inequality as above:

$$\begin{aligned}
1+\frac{1}{3+2\left(\frac{5}{j+1}+\frac{1}{k+1}+\frac{1}{i+1}\right)} &\geq 1+\frac{1}{3+2\left(\frac{5}{j'}+\frac{1}{k'+1}+\frac{1}{i'}\right)} = 1+\frac{1}{3+2\frac{(\sqrt{5}+2)^2}{n-5}}\\
&\in \frac{4}{3} - \Theta(n^{-1}).
\end{aligned}$$

□

3.4 Integrality Ratio for Metric TSP

In this section we give an upper bound on the integrality ratio for Metric TSP instances whose optimal fractional tour is isomorphic to $x_{i,j,k}$ for some i, j, k with $i+j+k+6 = n$. This implies that, assuming Conjecture 1.2.16, the Metric TSP instances described in [6] maximize the integrality ratio. Note that in [13] it was already shown that these instances have integrality ratios that are upper bounded by $\frac{4}{3}$. We start by defining coefficients for the pseudo-tours in $\mathfrak{T}$.

Definition 3.4.1. We define the real coefficients

$$\lambda^{\uparrow} := \frac{1}{|T^{\uparrow}|} \cdot \frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}$$
$$\lambda^{\circ} := \frac{1}{|T^{\circ}|} \cdot \frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}$$
$$\lambda^{\downarrow} := \frac{1}{|T^{\downarrow}|} \cdot \frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}$$
$$\lambda^{\nwarrow} := \lambda^{\nearrow} := \frac{\frac{1}{k+1}}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}$$
$$\lambda^{\leftarrow} := \lambda^{\rightarrow} := \frac{\frac{1}{j+1}}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}$$
$$\lambda^{\swarrow} := \lambda^{\searrow} := \frac{\frac{1}{i+1}}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}.$$

Lemma 3.4.2. *The integrality ratio of* Metric TSP *instances whose optimal fractional tour isomorphic to* $x_{i,j,k}$ *is at most* $1+\frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}$.

Proof. For a pseudo-tour T let χ^T be the vector such that $\chi^T(e)$ is the number of occurrence of e in T for all $e \in E(K_n)$. In order to show the statement, we show that

$$\sum_{T\in T^{\uparrow}} \lambda^{\uparrow}\chi^T + \sum_{T\in T^{\circ}} \lambda^{\circ}\chi^T + \sum_{T\in T^{\downarrow}} \lambda^{\downarrow}\chi^T + \lambda^{\nwarrow}\chi^{T^{\nwarrow}} + \lambda^{\nearrow}\chi^{T^{\nearrow}} + \lambda^{\leftarrow}\chi^{T^{\leftarrow}} + \lambda^{\rightarrow}\chi^{T^{\rightarrow}} + \lambda^{\swarrow}\chi^{T^{\swarrow}}$$
$$+\lambda^{\searrow}\chi^{T^{\searrow}} = \left(1+\frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}\right) x_{i,j,k}.$$

This implies the Lemma since

$$\sum_{T\in T^{\uparrow}} \lambda^{\uparrow} + \sum_{T\in T^{\circ}} \lambda^{\circ} + \sum_{T\in T^{\downarrow}} \lambda^{\downarrow} + \lambda^{\nwarrow} + \lambda^{\nearrow} + \lambda^{\leftarrow} + \lambda^{\rightarrow} + \lambda^{\swarrow} + \lambda^{\searrow}$$
$$=3\cdot \frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})} + 2\cdot \frac{\frac{1}{k+1}}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})} + 2\cdot \frac{\frac{1}{j+1}}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}$$
$$+2\cdot \frac{\frac{1}{i+1}}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})} = 1.$$

Consider the edge $\{Z_l, Z_{l+1}\}$ for some $0 \le l \le k$. It is contained in every pseudo-tour of $T^{\circ}, T^{\downarrow}, T^{\leftarrow}, T^{\rightarrow}, T^{\swarrow}, T^{\searrow}$ once and in each of $T^{\nwarrow}, T^{\nearrow}$ twice. In the pseudo-tours of $T^{\uparrow}$

it is contained twice except of the tour $T_l^{\uparrow}$ where it is not contained. Hence,

$$\begin{aligned}
&\sum_{T\in T^{\uparrow}} \lambda^{\uparrow}\chi^{T}(\{Z_l,Z_{l+1}\}) + \sum_{T\in T^{\circ}} \lambda^{\circ}\chi^{T}(\{Z_l,Z_{l+1}\}) + \sum_{T\in T^{\downarrow}} \lambda^{\downarrow}\chi^{T}(\{Z_l,Z_{l+1}\})\\
&+\lambda^{\nwarrow}\chi^{T^{\nwarrow}}(\{Z_l,Z_{l+1}\}) + \lambda^{\nearrow}\chi^{T^{\nearrow}}(\{Z_l,Z_{l+1}\}) + \lambda^{\leftarrow}\chi^{T^{\leftarrow}}(\{Z_l,Z_{l+1}\})\\
&+\lambda^{\rightarrow}\chi^{T^{\rightarrow}}(\{Z_l,Z_{l+1}\}) + \lambda^{\swarrow}\chi^{T^{\swarrow}}(\{Z_l,Z_{l+1}\}) + \lambda^{\searrow}\chi^{T^{\searrow}}(\{Z_l,Z_{l+1}\})\\
&=2\lambda^{\uparrow} - \frac{2}{|T^{\uparrow}|}\lambda^{\uparrow} + \lambda^{\circ} + \lambda^{\downarrow} + 2\lambda^{\nwarrow} + 2\lambda^{\nearrow} + \lambda^{\leftarrow} + \lambda^{\rightarrow} + \lambda^{\swarrow} + \lambda^{\searrow}\\
&=4\cdot\frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})} - \frac{2}{k+1}\cdot\frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})} + 4\cdot\frac{\frac{1}{k+1}}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}\\
&\quad+2\cdot\frac{\frac{1}{j+1}}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})} + 2\cdot\frac{\frac{1}{i+1}}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}\\
&=4\cdot\frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})} + 2\cdot\frac{\frac{1}{k+1}}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})} + 2\cdot\frac{\frac{1}{j+1}}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}\\
&\quad+2\cdot\frac{\frac{1}{i+1}}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}\\
&=1+\frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})} = \left(1+\frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}\right)x_{i,j,k}(\{Z_l,Z_{l+1}\}).
\end{aligned}$$

Next, consider the edge $\{Z_0,Y_0\}$. It is contained in the pseudo-tours of $T^{\uparrow},T^{\circ},T^{\nwarrow},T^{\leftarrow},T^{\searrow}$ once and not contained in the pseudo-tours of $T^{\downarrow},T^{\nearrow},T^{\rightarrow},T^{\swarrow}$. Hence,

$$\begin{aligned}
&\sum_{T\in T^{\uparrow}} \lambda^{\uparrow}\chi^{T}(\{Z_0,Y_0\}) + \sum_{T\in T^{\circ}} \lambda^{\circ}\chi^{T}(\{Z_0,Y_0\}) + \sum_{T\in T^{\downarrow}} \lambda^{\downarrow}\chi^{T}(\{Z_0,Y_0\})\\
&+\lambda^{\nwarrow}\chi^{T^{\nwarrow}}(\{Z_0,Y_0\}) + \lambda^{\nearrow}\chi^{T^{\nearrow}}(\{Z_0,Y_0\}) + \lambda^{\leftarrow}\chi^{T^{\leftarrow}}(\{Z_0,Y_0\})\\
&+\lambda^{\rightarrow}\chi^{T^{\rightarrow}}(\{Z_0,Y_0\}) + \lambda^{\swarrow}\chi^{T^{\swarrow}}(\{Z_0,Y_0\}) + \lambda^{\searrow}\chi^{T^{\searrow}}(\{Z_0,Y_0\})\\
&=\lambda^{\uparrow} + \lambda^{\circ} + \lambda^{\nwarrow} + \lambda^{\leftarrow} + \lambda^{\searrow}\\
&=2\cdot\frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})} + \frac{\frac{1}{k+1}}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})} + \frac{\frac{1}{j+1}}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}\\
&\quad+\frac{\frac{1}{i+1}}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}\\
&=\frac{1}{2}\left(1+\frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}\right) = \left(1+\frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}\right)x_{i,j,k}(\{Z_0,Y_0\}).
\end{aligned}$$

The statement can be shown for all other edges of $x_{i,j,k}$ analogously to one of the two cases above. □

Remark 3.4.3. Theorem 4.1 in [6] shows that the upper bound in Lemma 3.4.2 is tight, i.e. there is actually an instance where the integrality ratio is equal to $1+\frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}$.

Theorem 3.4.4. *The integrality ratio of* Metric TSP *instances whose optimal fractional tour is isomorphic to* $x_{i,j,k}$ *with* $i+j+k+6=n$ *is at most*

$$\begin{cases} 1+\frac{1}{3+\frac{18}{n-3}} & \text{if } n \equiv 0 \mod 3 \\ 1+\frac{1}{3+2(\frac{6}{n-4}+\frac{3}{n-1})} & \text{if } n \equiv 1 \mod 3 \\ 1+\frac{1}{3+2(\frac{3}{n-5}+\frac{6}{n-2})} & \text{if } n \equiv 2 \mod 3. \end{cases}$$

Proof. In order to maximize the integrality ratio, we need to find i,j,k with $i+j+k+6=n$ that minimize $\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1}$. Since the function $f(x)=\frac{1}{x}$ is convex, we can use Jensen's inequality to see that the integrality ratio is maximized for $i+1=j+1=k+1$ if n is divisible by 3. For $n \equiv 1 \mod 3$ and $n \equiv 2 \mod 3$ we use the Karamata's inequality (Theorem 1.4.17) to find the best values of i,j and k. For $n \equiv 1 \mod 3$ it is maximized for $i+1=j+1=k$ since this triple is been majorized by all other integer triples. Similarly, for $n \equiv 2 \mod 3$ the integrality ratio is maximized for $i+1=j=k$. □

Remark 3.4.5. Conjecture 4.1 in [6] states the given bounds in Theorem 3.4.4 hold for arbitrary Metric TSP instances with n vertices and is tight. Thus, Conjecture 1.2.16 would imply Conjecture 4.1 in [6].

3.5 Integrality Ratio for Multidimensional Rectilinear TSP

In this section we show that there are Rectilinear TSP instances in $\mathbb{R}^3$ that have the same integrality ratio as the upper bounds given in Theorem 3.4.4 for the Metric TSP. Hence, assuming Conjecture 1.2.16 the exact integrality ratio for Multidimensional Rectilinear TSP is the same as in the metric case. Since the instances in $\mathbb{R}^3$ can be embedded into $\mathbb{R}^d$ for $d \geq 3$, the statement also holds for these spaces.

We start by constructing an instance $I^3_{i,j,k}$ with the vertex set of $x_{i,j,k}$. The coordinates of the vertices are given by $X_s=(0,0,\frac{s}{i+1}), Y_s=(\frac{1}{i+1}+\frac{1}{j+1},0,\frac{s}{j+1})$ and $Z_s=(\frac{1}{i+1},\frac{1}{k+1},\frac{s}{k+1})$. The vertices form a prism in the three dimensional space where the triangle X_0, Y_0, Z_0 lies in the plane $z=0$ and the triangle $X_{i+1}, Y_{j+1}, Z_{k+1}$ lies in the plane $z=1$. The sequences of vertices $X_0,\dots,X_{i+1}$ and $Y_0,\dots,Y_{j+1}$ and $Z_0,\dots,Z_{k+1}$ are equidistant progressions such that each sequence lies on one of three parallel lines.

We can check that $\mathrm{dist}_1(X_l,X_{l+1})=\frac{1}{i+1}, \mathrm{dist}_1(Y_l,Y_{l+1})=\frac{1}{j+1}$ and $\mathrm{dist}_1(Z_l,Z_{l+1})=\frac{1}{k+1}$. Moreover, $\mathrm{dist}_1(X_0,Y_0)=\frac{1}{i+1}+\frac{1}{j+1}, \mathrm{dist}_1(Y_0,Z_0)=\frac{1}{j+1}+\frac{1}{k+1}$ and $\mathrm{dist}_1(X_0,Z_0)=\frac{1}{i+1}+\frac{1}{k+1}$. The same distances also hold for the triangle $X_{i+1}, Y_{j+1}, Z_{k+1}$. Note these distances are the same as the corresponding distances of the Metric TSP instances Benoit and Boyd described in [6]. Nevertheless, it is not clear that they have the same integraltiy ratio, since the remaining distances are given by the 1-norm instead of the metric closure of the weighted graph.

For all fixed n let the instance $I^3_n := I^3_{i^*,j^*,k^*}$ be the instance that maximizes the integrality ratio among the instances $I^3_{i,j,k}$ with $i+j+k+6=n$. Next, we determine the length of the optimal tours of $I^3_{i,j,k}$.

Lemma 3.5.1. *Every optimal tour of* $I^3_{i,j,k}$ *has at least length* $4+\frac{2}{i+1}+\frac{2}{j+1}+\frac{2}{k+1}$.

Proof. Assume that we have given an optimal tour T. We call an edge *vertical* if it is parallel to the z-axis, otherwise it is called *non-vertical*. A vertical edge is called *base edge* if it connects two consecutive vertices $\{X_s, X_{s+1}\}, \{Y_s, Y_{s+1}\}$ or $\{Z_s, Z_{s+1}\}$. We may assume that all vertical edges are base edges, otherwise we can replace them by a set of base edges to get a pseudo-tour with equal length. We call all base edges which are not in T *gaps*. Furthermore, we add auxiliary vertices $X_{-1} = (0, 0, -\frac{1}{i+1})$, $X_{i+2} = (0, 0, \frac{i+2}{i+1})$, $Y_{-1} = (\frac{1}{i+1} + \frac{1}{j+1}, 0, -\frac{1}{j+1})$, $Y_{j+2} = (\frac{1}{i+1} + \frac{1}{j+1}, 0, \frac{j+2}{j+1})$ and $Z_{-1} = (\frac{1}{i+1}, \frac{1}{k+1}, -\frac{1}{k+1})$, $Z_{k+2} = (\frac{1}{i+1}, \frac{1}{k+1}, \frac{k+2}{k+1})$ such that the sequences $X_{-1}, X_0, \ldots, X_{i+2}$ and $Y_{-1}, Y_0, \ldots, Y_{j+2}$ and $Z_{-1}, Z_0, \ldots, Z_{k+2}$ are equidistant progressions. We call the edges $\{X_{-1}, X_0\}$, $\{X_{i+1}, X_{i+2}\}$, $\{Y_{-1}, Y_0\}$, $\{Y_{j+1}, Y_{j+2}\}$, $\{Z_{-1}, Z_0\}$ and $\{Z_{k+1}, Z_{k+2}\}$ the *auxiliary gaps*.

Next, we assign the non-vertical edges in T to the gaps such that every non-vertical edge is assigned to two gaps incident to the edge on different lines, every non-auxiliary gap is assigned to two non-vertical edges and every auxiliary gap is assigned to one non-vertical edge. We do this as follows: For any endpoint of a non-vertical edge that is only incident to one gap we assign this edge to that gap. For any endpoint of a non-vertical edge incident to two gaps there has to be another non-vertical edge incident to that endpoint. We arbitrarily assign one of the edges to a gap and the other to the other gap. Note that since every gap has two endpoints we assigned two non-vertical edges to it this way. Moreover, every auxiliary gap has only one endpoint which is vertex of $x_{i,j,k}$ and hence it is assigned to one non-vertical edge.

Since we use the Manhattan norm, we can replace every edge $e \in T$ by three edges we call the *subedges* of e without changing the length of the tour such that they are parallel to the x-, y- and z-axis, respectively. After the replacement, we also call the subedges parallel to the z-axis *vertical*. Next, we replace all non-vertical subedges originated from a non-vertical edges of T by the two gaps it is assigned to and get a multiset of edges T'. We claim that T' has the same length as T. To see this assume that the non-vertical edge $\{X_l, Y_s\}$ is in T and note that the non-vertical subedges of it have total length $\text{dist}_1(X_0, Y_0) = \frac{1}{i+1} + \frac{1}{j+1}$. Assume that we assigned this edge to the gaps $\{X_l, X_{l+1}\}$ and $\{Y_s, Y_{s+1}\}$. Then, these two edges we add have also total length $\frac{1}{i+1} + \frac{1}{j+1}$. Similar statements hold for the non-vertical edges of the form $\{X_l, Z_s\}$ and $\{Y_l, Z_s\}$. Hence, T' has the same length as T.

Now, for $a \in \mathbb{R}$ consider the intersection of the plane $z = a$ with T'. We claim that it intersects T' at least four times for all $0 \leq a \leq 1$. The plane intersects each of the segments X_0X_{i+1}, Y_0Y_{j+1} and Z_0Z_{k+1} at a base edge or a gap. If it intersects at least one gap, the statement is true since every non-auxiliary gap was assigned to two non-vertical edges and we filled the gap by two edges. Otherwise, the plane intersects three base edges of T. Since the pseudo-tour T intersects a plane an even number of times, it has to intersect at least 4 times. Thus, it also intersects T' at least 4 times since we only replaced the non-vertical subedges. This shows the claim and implies that the part of the edges in T' with z-coordinate between 0 and 1 has length at least 4.

Moreover, we added an edge in T' to every auxiliary gap. Since the interior of the auxiliary gaps does not have z-coordinates between 0 and 1, this increases the lower bound of the length of T' by $\frac{2}{i+1} + \frac{2}{j+1} + \frac{2}{k+1}$. Therefore, the total length of T' and thus also that of T is at least $4 + \frac{2}{i+1} + \frac{2}{j+1} + \frac{2}{k+1}$. □

Corollary 3.5.2. *The integrality ratio of $I^3_{i,j,k}$ is at least* $1 + \frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}$.

Proof. By Lemma 3.5.1, the length of every tour is at least $4 + \frac{2}{i+1} + \frac{2}{j+1} + \frac{2}{k+1}$. The length of the fractional tour $x_{i,j,k}$ is

$$\begin{aligned}
&\mathrm{dist}_1(X_0, X_{i+1}) + \mathrm{dist}_1(Y_0, Y_{j+1}) + \mathrm{dist}_1(Z_0, Z_{k+1}) + \frac{1}{2}\mathrm{dist}_1(X_0, Y_0) + \frac{1}{2}\mathrm{dist}_1(X_0, Z_0)\\
&+\frac{1}{2}\mathrm{dist}_1(Y_0, Z_0) + \frac{1}{2}\mathrm{dist}_1(X_{i+1}, Y_{j+1}) + \frac{1}{2}\mathrm{dist}_1(X_{i+1}, Z_{k+1}) + \frac{1}{2}\mathrm{dist}_1(Y_{j+1}, Z_{k+1})\\
&=3 + \frac{2}{i+1} + \frac{2}{j+1} + \frac{2}{k+1}.
\end{aligned}$$

Hence, the integrality ratio is at least

$$\frac{4 + \frac{2}{i+1} + \frac{2}{j+1} + \frac{2}{k+1}}{3 + \frac{2}{i+1} + \frac{2}{j+1} + \frac{2}{k+1}} = 1 + \frac{1}{3 + 2(\frac{1}{i+1} + \frac{1}{j+1} + \frac{1}{k+1})}.$$

□

Corollary 3.5.3. *The instance $I^3_{i,j,k}$ has the same or higher integrality ratio as any* Metric TSP *instance whose optimal fractional solution is isomorphic to $x_{i,j,k}$.*

Proof. By Lemma 3.5.1, any Metric TSP instance with $x_{i,j,k}$ as the optimal fractional tour has at most integrality ratio $1 + \frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}$. The instance $I^3_{i,j,k}$ has at least the same integrality ratio by Corollary 3.5.2. □

Remark 3.5.4. Corollary 3.5.2 implies an alternative proof of the following statement: There exists a Metric TSP instance with $i + j + k + 6$ vertices having an integrality ratio of $1 + \frac{1}{3+2(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1})}$. This is a direct consequence of Lemma 4.1 in [6] and Lemma 4.2 in [6] and the key ingredient of Theorem 4.1 in [6].

3.6 Local Optimality

In this section we consider TSP instances with n vertices that can be embedded into $\mathbb{R}^d$ such that the distances arise from a norm which is totally differentiable in every non-zero point. An instance is called locally optimal if its integrality ratio cannot be increased by making small changes to its embedded vertices. We describe a criterion to check if an instance is locally optimal and develop an algorithm that finds locally optimal instances.

3.6.1 A Criterion for Local Optimality

Assume that we have given a norm $\|\cdot\|$ in $\mathbb{R}^d$ which is totally differentiable in every non-zero point. Let a TSP instance (K_n, c) be given where the vertices can be embedded into $\mathbb{R}^d$ as the vertex set $v = \{v_1, \ldots, v_n\}$ with $v_1, \ldots, v_n \in \mathbb{R}^d$ such that the cost function c arises from v and the given norm in $\mathbb{R}^d$. Since the instance is completely characterized by the embedded vertex set v, we will simply call this instance v. W.l.o.g. assume that no two vertices of the instance coincide, i.e. $v_i \neq v_j$ for $i \neq j$. We can interpret the vertex set v as a point in $\mathbb{R}^{nd}$ since each of the n vertices is a point in $\mathbb{R}^d$. For another point $w \in \mathbb{R}^{nd}$ and $\lambda \in \mathbb{R}$ we define a new instance $v + \lambda w$ where we add and multiply the coordinates of the vertices coordinate-wise. Moreover, let $\mathbb{T}$ be the set of optimal tours and $\mathbb{X}$ be the set of optimal fractional tours for v.

Definition 3.6.1. For a given tour T and fractional tour x and instance $y \in \mathbb{R}^{nd}$ let $l_T, l_x : \mathbb{R}^{nd} \to \mathbb{R}$ be defined as follows: $l_T(y)$ and $l_x(y)$ denote the length of T and cost of x for the instance y, respectively. Moreover, let $r_{T,x} : \mathbb{R}^{nd} \to \mathbb{R}$ be defined as $r_{T,x}(y) := \frac{l_T(y)}{l_x(y)}$, the ratio of the length of T and the cost of x for the instance v. Furthermore, let r_y be the integrality ratio of the instance y.

Now, we can define local optimality for a TSP instance.

Definition 3.6.2. The instance v is called *locally optimal* if there does not exist $w \in \mathbb{R}^{nd}$ such that $\liminf_{\epsilon \to 0} \frac{r_{v+\epsilon w} - r_v}{\epsilon} > 0$.

Lemma 3.6.3. *The instance v is locally optimal if and only if for some $x \in \mathbb{X}$ there does not exist $w \in \mathbb{R}^{nd}$ such that $\partial_w r_{T,x}(v) > 0$ for all $T \in \mathbb{T}$.*

Proof. Since the length of the tour and fractional tour is continuous in v and the number of tours and fractional tours is finite, given w for small enough ϵ the optimal tour of the instance $v + \epsilon w$ is still in $\mathbb{T}$ and the optimal fractional tour is still in $\mathbb{X}$. Hence, the new integrality ratio is

$$r_{v+\epsilon w} = \frac{\min_{T \in \mathbb{T}} l_T(v + \epsilon w)}{\min_{x \in \mathbb{X}} l_x(v + \epsilon w)} = \max_{x \in \mathbb{X}} \min_{T \in \mathbb{T}} \frac{l_T(v + \epsilon w)}{l_x(v + \epsilon w)} = \max_{x \in \mathbb{X}} \min_{T \in \mathbb{T}} r_{T,x}(v + \epsilon w).$$

Note that since the given norm is differentiable the functions l_T, l_x and $r_{T,x}$ are also differentiable. Moreover, we have by definition $r_v = r_{T,x}(v)$ for all $T \in \mathbb{T}$ and $x \in \mathbb{X}$. Therefore, $\liminf_{\epsilon \to 0} \frac{r_{v+\epsilon w} - r_v}{\epsilon} > 0$ if and only if we have $\lim_{\epsilon \to 0} \frac{r_{T,x}(v+\epsilon w) - r_{T,x}(v)}{\epsilon} = \partial_w r_{T,x}(v) > 0$ for some $x \in \mathbb{X}$ and all $T \in \mathbb{T}$. ☐

For a tour T and a fractional tour x let $\delta_T(v_i)$ and $\delta_x(v_i)$ denote the set of vertices incident to v_i in T and the support graph of x, respectively.

Lemma 3.6.4. *We have*

$$\partial_w l_T(v) = \frac{1}{2} \sum_{i \in \{1,\ldots,n\}} \sum_{v_j \in \delta_T(v_i)} \partial_w \|v_j - v_i\|$$
$$\partial_w l_x(v) = \frac{1}{2} \sum_{i \in \{1,\ldots,n\}} \sum_{v_j \in \delta_x(v_i)} x(\{v_i, v_j\}) \partial_w \|v_j - v_i\|.$$

Proof. Note that the length of the tour and fractional tour can be expressed as

$$l_T(v) = \frac{1}{2} \sum_{i \in \{1,\ldots,n\}} \sum_{v_j \in \delta_T(v_i)} \|v_j - v_i\|$$
$$l_x(v) = \frac{1}{2} \sum_{i \in \{1,\ldots,n\}} \sum_{v_j \in \delta_x(v_i)} x(\{v_i, v_j\}) \|v_j - v_i\|.$$

The statement follows from the fact that the derivative is linear. ☐

Definition 3.6.5. Define the function $g_{T,x,v} : \mathbb{R}^{nd} \to \mathbb{R}$ as $g_{T,x,v}(y) := l_T(y) - r_v l_x(y)$.

Lemma 3.6.6. *The instance v is locally optimal if and only if for some $x \in \mathbb{X}$ there does not exist $w \in \mathbb{R}^{nd}$ such that $\partial_w g_{T,x,v}(v) > 0$ for all $T \in \mathbb{T}$.*

Proof. By Lemma 3.6.3 v is locally optimal if and only if for some $x \in \mathbb{X}$ there does not exist $w \in \mathbb{R}^{nd}$ such that $\partial_w r_{T,x}(v) > 0$ for all $T \in \mathbb{T}$. We have:

$$\partial_w r_{T,x}(v) = \partial_w \frac{l_T(v)}{l_x(v)} = \frac{(\partial_w l_T(v))l_x(v) - l_T(v)(\partial_w l_x(v))}{l_x(v)^2}.$$

Hence

$$\begin{aligned}
\partial_w r_{T,x}(v) > 0 &\Leftrightarrow \frac{(\partial_w l_T(v))l_x(v) - l_T(v)(\partial_w l_x(v))}{l_x(v)^2} > 0 \\
&\Leftrightarrow (\partial_w l_T(v))l_x(v) - l_T(v)(\partial_w l_x(v)) > 0 \\
&\Leftrightarrow \partial_w l_T(v) - \frac{l_T(v)}{l_x(v)}\partial_w l_x(v) > 0 \\
&\Leftrightarrow \partial_w (l_T(v) - r_v l_x(v)) > 0 \\
&\Leftrightarrow \partial_w g_{T,x,v}(v) > 0.
\end{aligned}$$

□

Lemma 3.6.7. *The instance v is locally optimal if and only if for some $x \in \mathbb{X}$ there does not exist $w \in \mathbb{R}^{nd}$ such that $\langle w, \nabla g_{T,x,v}(v)\rangle > 0$ for all $T \in \mathbb{T}$.*

Proof. Recall that we assumed that the norm is totally differentiable for $p > 1$ in any non-zero point. Therefore, for $w \in \mathbb{R}^{nd}$ we have $\partial_w g_{T,x,v}(v) = \langle w, \nabla g_{T,x,v}(v)\rangle$. The statement follows from Lemma 3.6.6. □

Theorem 3.6.8. *The instance v is locally optimal if and only if for some $x \in \mathbb{X}$ there exist $\{\lambda_T \geq 0\}_{T\in\mathbb{T}}$ not all zero such that $\sum_{T\in\mathbb{T}} \lambda_T \nabla g_{T,x,v}(v) = \vec{0}$ where $\vec{0}$ is the vector consisting of zeros.*

Proof. Consider the following LP:

$$\begin{aligned}
&\min 0 \\
s.t.\ \langle w, \nabla g_{T,x,v}(v)\rangle &\geq 0 \qquad \forall T \in \mathbb{T}
\end{aligned} \tag{3.5}$$

and its dual LP

$$\begin{aligned}
&\max 0 \\
s.t.\ \sum_{T\in\mathbb{T}} \lambda_T \nabla g_{T,x,v}(v) &= \vec{0} \\
\lambda_T &\geq 0 \qquad \forall T \in \mathbb{T}
\end{aligned}$$

Note that all feasible solutions are optimal and both systems are feasible since we can set all variables equal to zero. By Lemma 3.6.7, the instance v is not locally optimal if and only if the primal has a solution where all inequalities are not tight. By complementary slackness, the dual has in this case only solutions where all λ_T are zero. Moreover, if the dual has a solution where $\lambda_T \neq 0$ for some $T \in \mathbb{T}$, by complementary slackness the corresponding inequality of the primal is tight for any primal solution. □

3.6.2 Local Optimality for the p-Norm

In this subsection we apply the criterion from the last subsection to the p-norm for $p > 1$ explicitly, i.e. we choose $\|\cdot\| = \|\cdot\|_p$. Note that the p-norm is differentiable in every non-zero point for $p > 1$ and hence satisfies the condition for the criterion.

Using a straightforward calculation with the chain rule we get for every unit vector $e \in \mathbb{R}^{nd}$:

$$\partial_e \|v_i - v_j\|_p = \operatorname{sgn}(\langle v_i - v_j, e\rangle) \frac{|\langle v_i - v_j, e\rangle|^{p-1}}{\|v_i - v_j\|_p^{p-1}}$$

where

$$\operatorname{sgn}(x) := \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{else} \end{cases}$$

is the sign function of x.

Thus, we have:

$$\begin{aligned} \partial_e l_T(v) &= \frac{1}{2} \sum_{i \in \{1,\dots,n\}} \sum_{q \in \delta_T(v_i)} \operatorname{sgn}(\langle v_i - q, e\rangle) \frac{|\langle v_i - q, e\rangle|^{p-1}}{\|v_i - q\|^{p-1}} \\ \partial_e l_x(v) &= \frac{1}{2} \sum_{i \in \{1,\dots,n\}} \sum_{q \in \delta_x(v_i)} x(\{v_i, q\}) \operatorname{sgn}(\langle v_i - q, e\rangle) \frac{|\langle v_i - q, e\rangle|^{p-1}}{\|v_i - q\|^{p-1}} \end{aligned}$$

With this we can compute $\nabla g_{T,x,v}$ as $\nabla g_{T,x,v} = \nabla l_T + g_v \nabla l_x$.

Remark 3.6.9. The above criterion can also be applied in the case of the 1-norm in restricted form. The 1-norm is totally differentiable for every instance where no two vertices have equal coordinates at the same position, i.e. in the two-dimensional case no two vertices have the same x- or y-coordinate. Hence, the criterion can be applied in these cases. For instances where there exist multiple vertices with equal coordinates at the same position we can treat these coordinates as one variable. In this case it is differentiable again and the criterion can be applied. Note that if the criterion is applied that way it does not necessarily detect all instances that are not locally optimal.

3.6.3 A Local Search Algorithm

In this subsection we develop a local search algorithm that finds a local optimal solution with respect to the integrality ratio.

Note that for instances that are not locally optimal the LP (3.5) has a solution where all inequalities are not tight. Therefore, we can solve a slightly modified LP that gives a direction vector that can be added to the instance to improve the integrality ratio.

We start by generating random instances until we get an instance v with integrality ratio by a given constant greater than 1. In every iteration we solve the following LP and try to improve the current integrality ratio:

$$\begin{aligned} &\max \delta \\ s.t.\ \langle w, \nabla g_{T,x,v}(v)\rangle &\geq \delta \qquad \forall T \in \mathbb{T} \\ -1 \leq w_i &\leq 1 \qquad \forall i \in \{1,\dots,nd\} \end{aligned} \tag{3.6}$$

If the objective value δ is greater than zero, w corresponds to a solution of LP (3.5) where all inequalities are not tight. Note that we added bounds for w_i to ensure that the LP is bounded. Given an optimal solution w of the LP we use binary search to determine the maximal η such that $v + \eta w$ has higher integrality ratio than v. We maintain a list $\mathbb{T}$ of optimal or near-optimal tours. In each iteration we include the current optimal tour T^* to $\mathbb{T}$ and delete the tours that are by more than a given constant longer than T^*. In contrast to the optimal tours we only store one current optimal fractional tour since in practice the local optima has usually many optimal tours but a unique optimal fractional tour (Algorithm 6).

Algorithm 6 Local Search Algorithm for Integrality Ratio

Input: Number of vertices n, accuracy parameters $\epsilon_0, \epsilon_1, \epsilon_2, \epsilon_3 > 0$
Output: Locally optimal instance v

do
  Generate a random instance v with n vertices
while integrality ratio of v is smaller than $1 + \epsilon_0$
Compute an optimal tour T^* and an optimal fractional tour x^* of v
Let $\mathbb{T} := \{T^*\}$
while LP (3.6) has a solution w with objective value $> \epsilon_1$ **do**
  Find by binary search η maximal such that $g(v + \eta w) > g(v)$
  if $\eta < \epsilon_2$ **then**
    break
  Let $v := v + \eta w$
  Compute an optimal tour T^* and an optimal fractional tour x^* of v
  Set $\mathbb{T} := \mathbb{T} \cup \{T^*\}$
  Delete all tours in $\mathbb{T}$ that is at least ϵ_3 longer than T^*
return v

3.7 Integrality Ratio for Euclidean TSP

In this section, we investigate the integrality ratio of Euclidean TSP. Using the local search algorithm described in Section 3.6.3 we can find local optima with respect to the integrality ratio for Euclidean TSP instances. Unfortunately, there are many such local optima. This means that we had to restart the algorithm several times with a small random modification of the last local optimum to get good results. The instances we found in the end with the highest integrality ratio seem to have the following structural properties that share similarities with the instances maximizing the integrality ratio in the rectilinear and metric case:

Observation 3.7.1. *We observe the following properties for the instances with the highest integrality ratio for* Euclidean TSP *found by the local search algorithm:*

1. *The optimal fractional solution is isomorphic to $x_{i,j,k}$ for some i, j, k.*
2. *The non-intersecting shortcuts of the pseudo-tours in $\mathfrak{T}$ are optimal tours (Figure 3.9).*

If the optimal fractional solution is isomorphic to $x_{i,j,i}$ for some i, j, we further obtain the following properties:

$n = 8$ $\quad\quad$ $n = 11$

Figure 3.12: The instances with the highest integrality ratio for $n = 8, 11$ found by the local search algorithm. Their optimal fractional tours are isomorphic to $x_{1,1,0}$ and $x_{2,2,1}$, respectively. The integrality ratios of these instances are approximately 1.0435 and 1.0789, respectively. The best integrality ratios found by the ellipse generation algorithm are approximately 1.0413 and 1.0784 achieved by instances which optimal fractional tour are isomorphic to $x_{1,0,1}$ and $x_{1,3,1}$, respectively.

3. *The instance can be rotated and shifted such that it is symmetric to the x- and y-axis and the inner vertices lie on the x-axis.*
4. *The outer vertices lie on an ellipse with foci on the x-axis.*

In the following we will refer to these properties by property 1, 2, 3 and 4. Using them we develop an efficient algorithm that constructs instances with high integrality ratio.

3.7.1 The Ellipse Construction Algorithm

In this subsection we describe an algorithm we call the *ellipse generation algorithm* that efficiently generates instances satisfying Observation 3.7.1.

In the following we assume that $i = k$ to use the additional properties 3 and 4 for an efficient algorithm to construct instances that match these patterns. This lets us construct instances with a larger number of vertices since due to the high number of local optima it is quite time consuming to use the local search method to generate good instances. The instances generated by the new efficient algorithm have for all tested n high integrality ratio. Like in the rectilinear case there are values of n where the optimal fractional solution is isomorphic to $x_{i,j,i+1}$ for some i, j, for example $n = 8, 11$ (Figure 3.12). Nevertheless, we ignore these cases since these instances are less symmetric and we do not understand their structure well.

By property 1, we may assume that the optimal fractional tour is isomorphic to $x_{i,j,i}$ with some fixed i, j. By symmetry, we can w.l.o.g. assume that the vertices

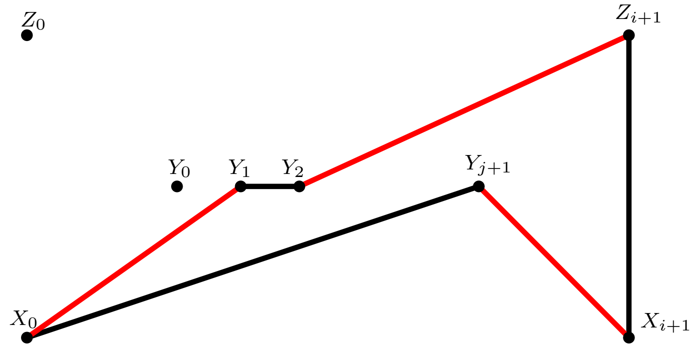


Figure 3.13: Construction of Y_{h+1} for $h+1=2$. The edges shown are that of $T^{\leftarrow} \triangle T_h^{\circ}$. The condition $\text{diffInner}_h = 0$ gives that the length of the red edges is equal to that of the black edges.

$X_0, X_{i+1}, Z_0, Z_{i+1}$ have the coordinates $(-b,-1),(b,-1),(-b,1),(b,1)$ for some $b>0$, respectively. The explicit value of b will be chosen later in the procedure. In the following we assume that an explicit value of b is given and describe how to determine the coordinates of the inner and outer vertices.

Inner Vertices

We first compute the coordinates of the inner vertices. By property 3, we know that the inner vertices lie on the x-axis and are symmetric to $(0,0)$. We first set the coordinates of Y_0 to $(-f,0)$ where f is a parameter to be determined later. With a given value for f the coordinates of Y_{h+1} for increasing h can be iteratively determined as follows: Assume we already know the coordinates of Y_h. Consider the difference of arbitrary shortcuts of the pseudo-tours $T^{\leftarrow}$ and T_h° (by symmetry all shortcuts have the same length) which is

$$\begin{aligned}\text{diffInner}_h := &\operatorname{dist}_2(X_0, Y_{j+1}) + \operatorname{dist}_2(X_{i+1}, Z_{i+1}) + \operatorname{dist}_2(Y_h, Y_{h+1}) - \operatorname{dist}_2(X_0, Y_h)\\ &- \operatorname{dist}_2(Y_{h+1}, Z_{i+1}) - \operatorname{dist}_2(X_{i+1}, Y_{j+1})\end{aligned}$$

(Figure 3.13). Since the coordinates of $X_0, X_{i+1}, Z_{i+1}, Y_h$ and Y_{j+1} are already known, this is equivalent to $\text{diffInner}_h := \operatorname{dist}_2(Y_h, Y_{h+1}) - \operatorname{dist}_2(Y_{h+1}, Z_{i+1}) + c$ for some constant c. Since Y_{h+1} lies on the x-axis right of Y_h and left of Z_{i+1} the condition $\text{diffInner}_h = 0$ from property 2 determines the position of Y_{h+1} uniquely as diffInner_h is monotonic increasing in the x-coordinate of Y_{h+1}. By symmetry, this also determines the position of $Y_{j+1-(h+1)}$.

For odd j we further know that the coordinate of $Y_{\frac{j+1}{2}}$ is by symmetry $(0,0)$. Therefore, we need to find f such that $\text{diffInner}_{\frac{j-1}{2}} = 0$. Similarly, for even j the coordinate of $Y_{\frac{j}{2}+1}$ is determined by symmetry and we need to find f such that $\text{diffInner}_{\frac{j}{2}} = 0$. Now, we could try every value of f up to a certain accuracy. To speed up the calculation, we make the following observation:

Observation 3.7.2. *The x-coordinate of Y_l for a fixed l with $1 \le l \le j$ determined by the procedure above is monotonically decreasing in f.*

This observation can be shown using induction and monotonicity arguments. Hence, we can use binary search to find the correct f such that the inner vertices are symmetric to the y-axis (Algorithm 7).

Algorithm 7 Inner Vertices Algorithm

Input: i, j, b, accuracy ϵ
Output: Coordinates of the inner vertices $Y_0, \ldots, Y_{j+1}$

if j is odd **then**:
  Set the coordinates of the vertex $Y_{\frac{j+1}{2}} := (0, 0)$
Binary search for f such that the inner vertices satisfy Observation 3.7.1:
for each value f we evaluate **do**
  Set the coordinates of $Y_0 := (-f, 0)$ and $Y_{j+1} := (f, 0)$
  for h from 0 to $\lfloor \frac{j}{2} \rfloor - 1$ **do**
    Determine the coordinates of Y_{h+1}
    Set by symmetry the coordinates of $Y_{j+1-(h+1)}$
  if $|\text{diffInner}_{\lfloor \frac{j}{2} \rfloor}| < \epsilon$ **then**
    return coordinates of $Y_0, \ldots, Y_{j+1}$
  else
    Increase or decrease f depending on the sign of $\text{diffInner}_{\lfloor \frac{j}{2} \rfloor}$ and repeat

Outer Vertices

The outer vertices are harder to compute since they are not uniquely determined by b in contrast to the inner vertices. By property 4, we know that X_0, X_{i+1}, Z_0 and Z_{i+1} and the other outer vertices lie on an ellipse. Unfortunately, five vertices are needed to determine an ellipse, therefore we need one more parameter. By property 3, the coordinate of the foci of the ellipse have the form $(-e, 0)$ and $(e, 0)$. We will take the value of e as an additional parameter.

Using the ellipse we make the observation that the coordinate of the other outer vertices are uniquely determined. We iteratively construct the coordinates of Z_{h+1} for increasing h. Assume that the coordinates of Z_h are already known. Consider the difference of the length of any non-intersecting shortcuts of $T^{\leftarrow}$ and $T_h^{\uparrow}$ (by symmetry all of the non-intersecting shortcuts have the same length). It is equal to

$$\begin{aligned}\text{diffOuter}_h := &\,\text{dist}_2(X_0, Y_0) + \text{dist}_2(Z_0, Y_{j+1}) - \text{dist}_2(X_0, Z_0) - \text{dist}_2(Z_h, Y_0) \\ &- \text{dist}_2(Z_{h+1}, Y_{j+1}) + \text{dist}_2(Z_h, Z_{h+1}).\end{aligned}$$

Moreover, we know the coordinates of X_0, Z_0, Z_h, Y_0 and Y_{j+1}. By property 2 we have $\text{diffOuter}_h = 0$ and therefore $\text{dist}_2(Z_h, Z_{h+1}) - \text{dist}_2(Z_{h+1}, Y_{j+1}) + c = 0$ for some constant c. This equation describes a hyperbola with foci Z_h and Y_{j+1}. It intersects the ellipse twice, once on each side of the line $Z_h Y_{j+1}$ (Figure 3.14). Thus, we can determine Z_{h+1} as the intersection of the hyperbola with the ellipse that lies on the same side as Z_{i+1}.

By symmetry, the coordinates of $Z_{i+1-(h+1)}, X_{h+1}, X_{i+1-(h+1)}$ are determined by that of Z_{h+1}. For odd i by symmetry the coordinate of the vertex $Z_{\frac{i+1}{2}}$ has x-coordinate 0 and has positive y-coordinate. Since the vertex lies on the ellipse, its coordinates are uniquely determined. Therefore, we want to choose e such that $\text{diffOuter}_{\frac{i-1}{2}} = 0$. For even i the coordinates of $Z_{\frac{i}{2}+1}$ are determined by symmetry and we want to choose e such that $\text{diffOuter}_{\frac{i}{2}} = 0$. It remains to determine the value(s) of e such that the condition above is satisfied. We could simply try every value of e up to a certain accuracy. To speed up the process, we make the following observations:

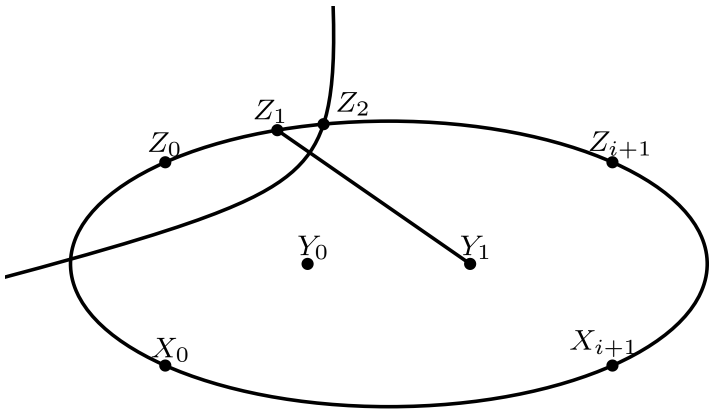


Figure 3.14: Construction of Z_{h+1} with $h+1=2$ and $j=0$ as the intersection of the ellipse with the hyperbola.

Observation 3.7.3. *The x-coordinate of $Z_{\lfloor \frac{i}{2} \rfloor}$ determined by the procedure above is monotonically increasing in e.*

Observation 3.7.4. $\mathrm{diffOuter}_{\lfloor \frac{i}{2} \rfloor}$ *is monotonically decreasing in e if the vertex $Z_{\lfloor \frac{i}{2} \rfloor}$ constructed by the procedure above has negative x-coordinate.*

Using Observation 3.7.3 and 3.7.4 we can speed up the process of finding the correct value of e by using binary search on e (Algorithm 8).

Algorithm 8 Outer Vertices Algorithm

Input: i, j, b, coordinates for the vertices Y_0, Y_{j+1}, accuracy ϵ
Output: Coordinates for the vertices $X_1, \ldots, X_i, Z_1, \ldots, Z_i$

Binary search for e such that the outer vertices are symmetric to the y-axis:
for each e we evaluate **do**
  if i is odd **then**
    Set $X_{\frac{i+1}{2}}$ and $Z_{\frac{i+1}{2}}$ to the intersections of $x=0$ with the ellipse through X_0 with foci $(-e,0)$ and $(e,0)$
  for h from 0 to $\lfloor \frac{i}{2} \rfloor - 1$ **do**
    Compute the position of Z_{h+1} as the intersection of the hyperbola $\mathrm{diffOuter}_h = 0$ and the ellipse through the outer vertices with foci $(-e,0)$ and $(e,0)$
    Set the position of $Z_{i+1-(h+1)}, X_{h+1}, X_{i+1-(h+1)}$ by symmetry
  if $Z_{\lfloor \frac{i}{2} \rfloor}$ has non-negative x-coordinate **then**
    Decrease e and repeat
  if $|\mathrm{diffOuter}_{\lfloor \frac{i}{2} \rfloor}| < \epsilon$ **then**
    return the coordinates of $X_1, \ldots, X_i, Z_1, \ldots, Z_i$
  else
    Increase or decrease e depending on the sign of $\mathrm{diffOuter}_{\lfloor \frac{i}{2} \rfloor}$ and repeat

Best Value for b

Let us denote the integrality ratio of the instance constructed by the inner and outer vertices algorithm with b as given parameter by $\mathrm{ratio}(b)$. Note that $\mathrm{ratio}(b)$ is not defined for every b: If b is too small or too large, there is no e such that the outer vertices can be constructed satisfying the properties 2 and 3. In this case the outer vertices algorithm fails to find suitable coordinates for the outer vertices. To find the instance with maximal integrality ratio by this construction we need to determine the value of b that maximizes $\mathrm{ratio}(b)$. We make the following observation that helps us to do this efficiently:

Observation 3.7.5. *The function* $\mathrm{ratio}(b)$ *is a concave function in* b*.*

Therefore, we can efficiently minimize a concave function to find the b maximizing $\mathrm{ratio}(b)$ instead of using brute force. Given a constructed instance with a fixed b the properties 1 and 2 allow us to speed up the computation of the integrality ratio: Instead of computing an optimal fractional tour and an optimal tour we can just compute the cost of $x_{i,j,i}$ and the length of any non-intersecting shortcut of a pseudo-tour in $\mathfrak{T}$. All in all, the above considerations result in the following algorithm we call the ellipse construction algorithm:

Algorithm 9 EUCLIDEAN TSP Ellipse Construction Algorithm

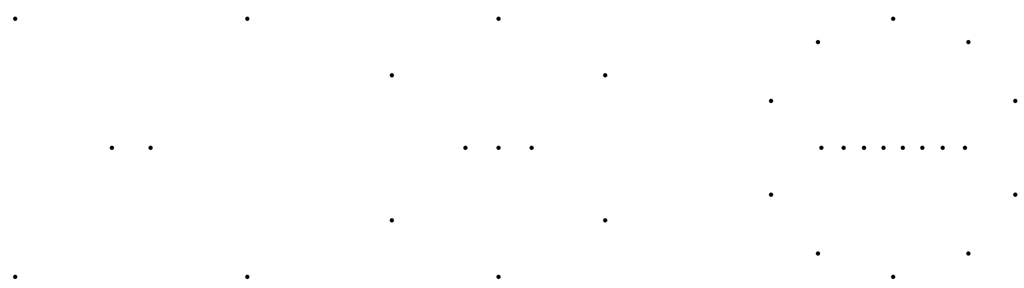
Input: i, j, accuracy ϵ
Output: EUCLIDEAN TSP instance

Optimize over a concave function to find b that maximizes $\mathrm{ratio}(b)$:
for each b we evaluate **do**
  Use the inner vertices algorithm to compute the inner vertices
  Use the outer vertices algorithm to compute the outer vertices
  Compute the length of any non-intersecting shortcut of a pseudo-tour in $\mathfrak{T}$
  Compute the cost of the fractional tour $x_{i,j,i}$
  Divide the two values to get the integrality ratio assuming properties 1 and 2.

3.7.2 Results of the Ellipse Construction Algorithm

In this subsection we describe the instances found by the ellipse construction algorithm. For $\epsilon = 10^{-9}$ and every $6 \leq n \leq 199$, $x_{i,j,i}$ with $i + j + i + 6 = n$ we executed the algorithm and took the best result for every n. The actual integrality ratio of the resulting instances have been computed using the `Concorde` TSP solver for $6 \leq n \leq 109$ vertices. The computation time was too high for the remaining instances, see Section 3.9 for more details on this phenomenon. Up to this point we could verify that property 2 of Observation 3.7.1 holds, i.e. the non-intersecting shortcuts of the pseudo-tours in $\mathfrak{T}$ are optimal tours. For higher number of vertices the integrality ratio was computed assuming property 2. Some of the instances the ellipse construction algorithm generated are shown in Figure 3.15.

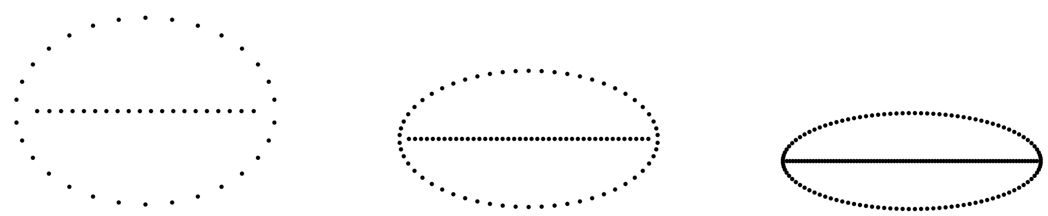
As in the rectilinear case we also see that vertices are not distributed equally. For large n there are more inner than outer vertices. This unequal distribution occurs first at $n = 18$ where we have 5+5 outer and 8 inner vertices. Moreover, for small n the ellipse is nearly a circle and becomes flatter for increasing n.



$n = 6, i = 0, j = 0$, ratio ≈ 1.0238

$n = 9, i = 1, j = 1$, ratio ≈ 1.060

$n = 18, i = 3, j = 6$, ratio ≈ 1.1319



$n = 50, i = 13, j = 18$, ratio ≈ 1.2263

$n = 100, i = 27, j = 40$, ratio ≈ 1.2695

$n = 199, i = 54, j = 85$, ratio ≈ 1.2970

Figure 3.15: Instances with n vertices constructed by the ellipse construction algorithm.

3.8 Comparing Integrality Ratio

In this section we compare the lower bounds on the integrality ratio we found in the previous sections for the TSP variants to each other and to the instances from the literature.

In the previous sections we showed lower bounds on the integrality ratio of $1 + \frac{1}{3+2\left(\frac{5}{j+1}+\frac{1}{i+1}+\frac{1}{k+1}\right)}$ and $1 + \frac{1}{3+2\left(\frac{1}{i+1}+\frac{1}{j+1}+\frac{1}{k+1}\right)}$ for the Rectilinear and Multidimensional Rectilinear / Metric TSP, respectively. As the deviation converges to 0 for $n \to \infty$, we discard in this subsection for simplicity the integrality constraints of i, j and k. In this case the bounds for the Rectilinear TSP and Multidimensional Rectilinear/ Metric TSP are $1 + \frac{1}{3+\frac{2(\sqrt{5}+2)^2}{n-3}}$ and $1 + \frac{1}{3+\frac{18}{n-3}}$, respectively. As we can see, both values converge to $\frac{4}{3}$ as $n \to \infty$. By a straightforward calculation, we get

$$1 + \frac{1}{3 + \frac{2(\sqrt{5}+2)^2}{n-3}} = 1 + \frac{1 + \frac{2}{3} \cdot \frac{(\sqrt{5}+2)^2}{n-3}}{3 + \frac{2(\sqrt{5}+2)^2}{n-3}} - \frac{\frac{2}{3} \cdot \frac{(\sqrt{5}+2)^2}{n-3}}{3 + \frac{2(\sqrt{5}+2)^2}{n-3}} = \frac{4}{3} - \frac{\frac{2}{3}\frac{(\sqrt{5}+2)^2}{n-3}}{3 + \frac{2(\sqrt{5}+2)^2}{n-3}}$$
$$= \frac{4}{3} - \frac{\frac{2}{3}(\sqrt{5}+2)^2}{3(n-3) + 2(\sqrt{5}+2)^2}.$$

Similarly, we get $1 + \frac{1}{3+\frac{18}{n-3}} = \frac{4}{3} - \frac{6}{3(n-3)+18}$. Hence, there are constants c_1, c_2 such that the lower bounds for the RECTILINEAR and MULTDIMENSONAL RECTILINEAR / METRIC TSP are $\frac{4}{3} - \frac{\frac{2}{9}(\sqrt{5}+2)^2}{n+c_1} \approx \frac{4}{3} - \frac{3.988}{n+c_1}$ and $\frac{4}{3} - \frac{2}{n+c_2}$, respectively. Since the additive constants c_1, c_2 are neglectable as $n \to \infty$, we see that the latter converges to $\frac{4}{3}$ roughly twice as fast as the former.

The tetrahedron instances for EUCLIDEAN TSP in [38] have an integrality ratio between $\frac{4n+\frac{4n}{\sqrt{3}}-69}{3n+\frac{3n}{\sqrt{3}}}$ and $\frac{4n+\frac{4n}{\sqrt{3}}-17}{3n+\frac{3n}{\sqrt{3}}-33}$. These bounds are too inaccurate to directly compare the rate of convergence. Figure 3.16 shows the integrality ratio which was explicitly computed in [37]. Note that for each fixed number of vertices the tetrahedron instances depend on two parameters and they were chosen in [37] to maximize the runtime of `Concorde` instead of the integrality ratio.

Unfortunately, we do not have a formula for the integrality ratio of the Euclidean instances found by the ellipse construction algorithm. So we cannot directly compare their rate of convergence but only the explicitly computed integrality ratios. Figure 3.16 shows the integrality ratio of the various instances described in the previous sections. Note that for the constructed Euclidean instances the integrality ratio was computed for $6 \leq n \leq 109$ vertices by `Concorde`. For instances with $110 \leq n \leq 199$ vertices the runtime of `Concorde` was too high to verify the integrality ratio, see Section 3.9 for more details on this phenomenon. For these cases we rely on property 2 of Observation 3.7.1 and assume the non-intersecting shortcuts of the pseudo-tours in $\mathfrak{T}$ are optimal tours. This property holds for the instances with $6 \leq n \leq 109$ vertices. As we can see from the plot, the integrality ratios of the constructed instances are higher than these of the tetrahedron instances. For this data they converge roughly two and four times slower than the rectilinear and metric instances, respectively.

3.9 Hard to Solve Instances

In this section we investigate the runtime of `Concorde` for the instances $I^3_{i,i-1,i+1}, I^3_{i,i-1,i+2}$ and $I^3_{i,i-1,i+3}$. The runtime of `Concorde` for solving these instances is much higher than for the known tetrahedron instances from [38].

First, we observe that symmetry seems to affect the computational results a lot. The instance $I^3_{10,10,10}$ can be solved by `Concorde` in less than a second. A small modification of the distribution of vertices on the three lines increases the runtime significantly: $I^3_{10,9,11}$ needs more than 1000 seconds to solve on the same hardware.

In the following we tested the runtime of the following instances: For $n = 3(i+2)$, $n = 3(i+2)+1$ and $n = 3(i+2)+2$ we solved the instances $I^3_{i,i-1,i+1}, I^3_{i,i-1,i+2}$ and $I^3_{i,i-1,i+3}$ with seed 1 by `Concorde`, respectively. The distances were multiplied by 1000 and rounded down to the nearest integer. `Corcorde-03.12.19` was compiled with `gcc 4.8.5` and using `CPlex 12.04` as LP solver. We used a single core of an `AMD EPYC 7601` processor for every run. The resulting runtimes are shown in Figure 3.17.

Using least-square fit regression of the logarithmic runtime we get the following exponential regression for the runtime

$$0.144 \cdot 1.304^n.$$

Since $1.304^3 > 2.2$, the estimated computation time more than doubles whenever n increases by 3. Based on this estimate an instance with 100 vertices would need more

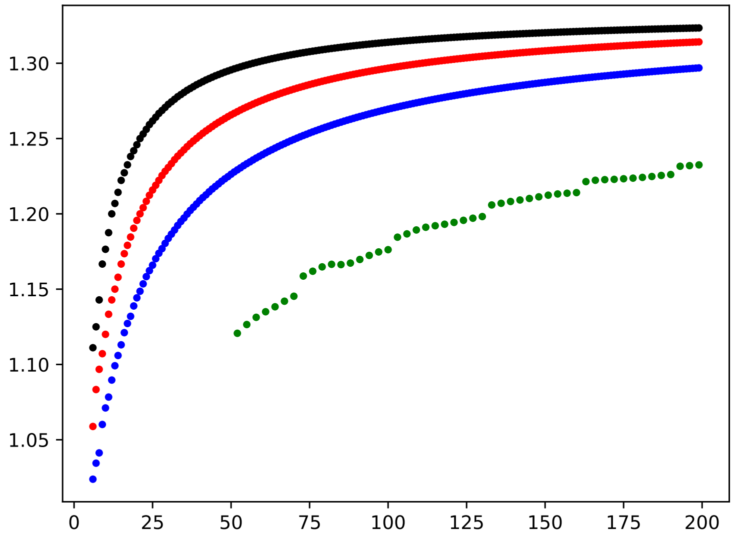


Figure 3.16: The integrality ratio for the TSP variants. The black, red and blue dots correspond to the lower bounds on the integrality ratio of the MULTIDIMENSIONAL RECTILINEAR TSP/METRIC TSP from [6], RECTILINEAR TSP and EUCLIDEAN TSP, respectively. The integrality ratio of the EUCLIDEAN TSP instances with $110 \leq n \leq 199$ vertices are computed assuming property 2 of Observation 3.7.1. The green dots correspond to the integrality ratio of the tetrahedron instances from [38].

than 1500 years. This runtime is much higher compared to the tetrahedron instances from [38]. For 52 vertices the runtime for the tetrahedron instances is about 10 seconds compared to over 2 days for the tested instances. It should be noted that the calculations in [38] were performed by a different processor which is about 20% faster. Nevertheless, comparing with the regression function $0.480 \cdot 1.0724^n$ for the tetrahedron instances we see that the exponential basis is much greater. This implies that the growth of runtime is also much faster.

The instances I_n^2 seem to have similar runtimes as the instances tested above. The runtime was a bit higher if the numbers of vertices in the outer two lines are not equal. It was not obvious which distribution of vertices maximizes the runtime.

In contrast the metric instances described in [6] with $n = 36$ can be solved by `Concorde` in a few seconds compared to more than half an hour for $I^3_{10,9,11}$ with the same number of vertices. Although the runtime seems to be exponential also in this case, modifying the number of vertices on the three lines still does not give similar high runtimes. The same holds for the Euclidean instances constructed by the ellipse construction algorithm.

Note that the results of Section 3.5 can be easily extended to determine the structure of the optimal tours of the instances $I^3_{i,i-1,i+1}$, $I^3_{i,i-1,i+2}$ and $I^3_{i,i-1,i+3}$. Using this knowledge the optimal tour can be computed in linear time.

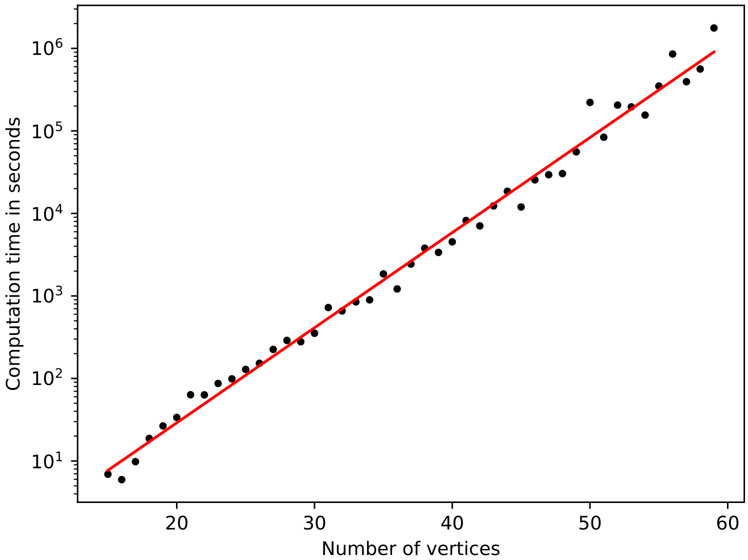


Figure 3.17: The `Concorde` runtime for the instances $I^3_{i,i-1,i+1}$, $I^3_{i,i-1,i+2}$ and $I^3_{i,i-1,i+3}$ in seconds. The red function is the least-square fit regression of the logarithmic runtime.

3.10 Integrality Ratio for Metric $s-t$ Path TSP

This section is based on [72]. We show the improved upper bound on the integrality ratio for METRIC $s-t$ PATH TSP of 1.5273. For this we use a theorem from [62]:

Theorem 3.10.1 (Theorem 5 in [62])**.** *Let $h : [0,1] \to [0,1]$ be an integrable function with*

$$\int_z^1 \max\{0, h(\sigma) - 1 + zh(\sigma)\} \mathrm{d}\sigma + \int_0^z (h(\sigma) - 1 - zh(\sigma)) \mathrm{d}\sigma \leq 0 \tag{3.7}$$

for all $z \in [0,1]$. Then, the best-of-many Christofides' algorithm with lonely edge deletion [58] computes a solution of cost at most $\rho^* c(x^*)$, where

$$\rho^* = 1 + \frac{1}{1 + \int_0^1 h(\sigma)\mathrm{d}\sigma}.$$

Traub and Vygen applied Theorem 3.10.1 for $h(\sigma) := \frac{4}{4+\sigma}$. We define our choice of h as follows:

Definition 3.10.2. h is a step function taking the value $\alpha := 0.971239$ in $[0,x)$ and the value $\beta := 0.873362$ in $[x,1]$ where $x := 0.236901$ (Figure 3.18), i.e.

$$h(\sigma) = \begin{cases} 0.971239 & \text{if } \sigma \in [0, 0.236901) \\ 0.873362 & \text{otherwise.} \end{cases}$$

In order to apply Theorem 3.10.1 we need to show that the condition (3.7) is satisfied.

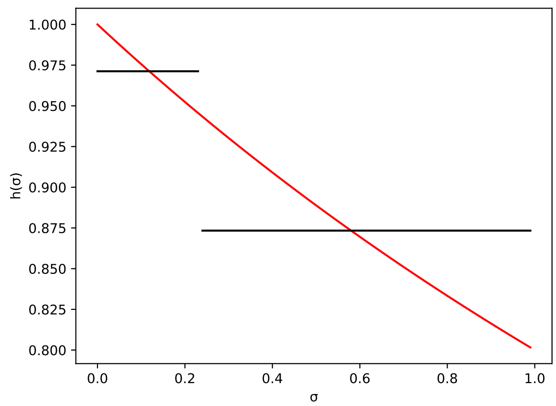


Figure 3.18: The black function shows our choice of h, the red function that chosen by Traub and Vygen in [62].

Lemma 3.10.3. *Our choice of h satisfies the condition of Theorem 3.10.1.*

Proof. Since by definition $h(\sigma) > 0$ for all $\sigma \in [0,1]$, we have $h(\sigma) - 1 + zh(\sigma) < 0$ if and only if $z < \frac{1}{h(\sigma)} - 1$. For our choice of h we have that $h(\sigma)$ can only take two values: α and β. Note that $0 < \frac{1}{\alpha} - 1 < 0.03$ and $0.145 < \frac{1}{\beta} - 1 < 0.146$. Thus, we can distinguish four cases: $z \in [0, \frac{1}{\alpha} - 1), z \in [\frac{1}{\alpha} - 1, \frac{1}{\beta} - 1), z \in [\frac{1}{\beta} - 1, x)$ and $z \in [x, 1]$.

The first case is trivial, since for $z \in [0, \frac{1}{\alpha} - 1)$ we have:

$$\begin{aligned} &\int_z^1 \max\{0, h(\sigma) - 1 + zh(\sigma)\}\mathrm{d}\sigma + \int_0^z (h(\sigma) - 1 - zh(\sigma))\mathrm{d}\sigma \\ =&\int_0^z ((1-z)h(\sigma) - 1)\mathrm{d}\sigma \le \int_0^z 0\mathrm{d}\sigma = 0. \end{aligned}$$

For $z \in [\frac{1}{\alpha} - 1, \frac{1}{\beta} - 1)$ we have:

$$\begin{aligned} &\int_z^1 \max\{0, h(\sigma) - 1 + zh(\sigma)\}\mathrm{d}\sigma + \int_0^z (h(\sigma) - 1 - zh(\sigma))\mathrm{d}\sigma \\ =&\int_z^x (\alpha - 1 + z\alpha)\mathrm{d}\sigma + z(\alpha - 1 - z\alpha) = (x-z)(\alpha - 1 + z\alpha) + z(\alpha - 1 - z\alpha) \\ =& x\alpha - x + x\alpha z - 2\alpha z^2. \end{aligned}$$

Similarly, for $z \in [\frac{1}{\beta} - 1, x)$ we have:

$$\begin{aligned} &\int_z^1 \max\{0, h(\sigma) - 1 + zh(\sigma)\}\mathrm{d}\sigma + \int_0^z (h(\sigma) - 1 - zh(\sigma))\mathrm{d}\sigma \\ =&\int_z^x (\alpha - 1 + z\alpha)\mathrm{d}\sigma + \int_x^1 (\beta - 1 + z\beta)\mathrm{d}\sigma + z(\alpha - 1 - z\alpha) \\ =&(x-z)(\alpha - 1 + z\alpha) + (1-x)(\beta - 1 + z\beta) + z(\alpha - 1 - z\alpha) \\ =& x\alpha - 1 + (1-x)\beta + (x\alpha + (1-x)\beta)z - 2\alpha z^2. \end{aligned}$$

Finally, for $z \in [x, 1]$ we have:

$$\begin{aligned}
&\int_z^1 \max\{0, h(\sigma) - 1 + zh(\sigma)\} \mathrm{d}\sigma + \int_0^z (h(\sigma) - 1 - zh(\sigma)) \mathrm{d}\sigma \\
=& \int_z^1 (\beta - 1 + z\beta) \mathrm{d}\sigma + \int_0^x (\alpha - 1 - z\alpha) \mathrm{d}\sigma + \int_x^z (\beta - 1 - z\beta) \mathrm{d}\sigma \\
=&(1 - z)(\beta - 1 + z\beta) + x(\alpha - 1 - z\alpha) + (z - x)(\beta - 1 - z\beta) \\
=&x\alpha - 1 + (1 - x)\beta + (-x\alpha + (x + 1)\beta)z - 2\beta z^2.
\end{aligned}$$

Hence, it is enough to show that for all $z \in \mathbb{R}$ we have:

$$\begin{aligned}
x\alpha - x + x\alpha z - 2\alpha z^2 &\leq 0 \\
x\alpha - 1 + (1 - x)\beta + (x\alpha + (1 - x)\beta)z - 2\alpha z^2 &\leq 0 \\
x\alpha - 1 + (1 - x)\beta + (-x\alpha + (x + 1)\beta)z - 2\beta z^2 &\leq 0.
\end{aligned}$$

The left hand sides are quadratic functions in z. Note that the leading coefficient is negative in all three cases. Hence, the inequalities hold if and only if the discriminants of the three quadratic functions are non-positive, that is:

$$(x\alpha)^2 + 8\alpha(x\alpha - x) \leq 0 \tag{3.8}$$
$$(x\alpha + (1 - x)\beta)^2 + 8\alpha(x\alpha - 1 + (1 - x)\beta) \leq 0 \tag{3.9}$$
$$(-x\alpha + (x + 1)\beta)^2 + 8\beta(x\alpha - 1 + (1 - x)\beta) \leq 0. \tag{3.10}$$

We can check these inequalities for our choice of x, α, β:

$$\begin{aligned}
(x\alpha)^2 + 8\alpha(x\alpha - x) &< -1.17266 \cdot 10^{-7} < 0 \\
(x\alpha + (1 - x)\beta)^2 + 8\alpha(x\alpha - 1 + (1 - x)\beta) &< -3.5346 \cdot 10^{-6} < 0 \\
(-x\alpha + (x + 1)\beta)^2 + 8\beta(x\alpha - 1 + (1 - x)\beta) &< -3.00596 \cdot 10^{-6} < 0.
\end{aligned}$$

□

Theorem 3.10.4. *The integrality ratio of the standard LP relaxation for the* METRIC $s - t$ PATH TSP *is at most 1.5273.*

Proof. By Lemma 3.10.3, our choice of h satisfies the condition of Theorem 3.10.1. Hence, we can apply Theorem 3.10.1 to get an upper bound on the integrality ratio of

$$\rho^* := 1 + \frac{1}{1 + \int_0^1 h(\sigma)\mathrm{d}\sigma} = 1 + \frac{1}{1 + x\alpha + (1 - x)\beta} < 1.5273.$$

□

Remark 3.10.5. The values for α, β and x we chose are approximate values of a solution for the system of equations we get by replacing the less-than-or-equal sign in (3.8), (3.9) and (3.10) by an equal sign. More precise values would probably lead to a better upper bound. Using a computer algebra system, we can solve that system of equations to get the exact values:

$$\alpha := \frac{1}{48}\left(34 + \frac{73}{\sqrt[3]{-377 + 18i\sqrt{762}}} + \sqrt[3]{-377 + 18i\sqrt{762}}\right)$$
$$\beta := \frac{2}{3}\left(-45 + 172\alpha - 128\alpha^2\right)$$
$$x := 8(\frac{1}{\alpha} - 1),$$

where the roots are principal roots and i is the imaginary unit. By definition, it is clear that the inequalities (3.8), (3.9) and (3.10) are fulfilled by this choice of values. This would lead to an upper bound on the integrality ratio of

$$\frac{-30(377i + 18\sqrt{762}) + (-377 + 18i\sqrt{762})^{\frac{2}{3}}(-249i + 28\sqrt{762}) + (-377 + 18i\sqrt{762})^{\frac{1}{3}}(-3975i + 206\sqrt{762})}{4((-377 + 18i\sqrt{762})^{\frac{2}{3}}(-44i + 7\sqrt{762}) - 16(377i + 18\sqrt{762}) + (-377 + 18i\sqrt{762})^{\frac{1}{3}}(-1088i + 47\sqrt{762}))} < 1.5273.$$

Remark 3.10.6. As already pointed out in [62] numerical computations indicate that the best choice of h gives an upper bound on the integrality ratio of approximately 1.5273. Hence, this suggests that our choice of h is near-optimal.

4 Conclusion and Open Problems

In this section we briefly summarize the results in this thesis and give an outlook to open problems. Instead of stating all open problems related to the TSP we will focus here on those closely related to the content of this thesis.

4.1 Approximation Ratio of the k-Opt and Lin-Kernighan Algorithm

For the 2-Opt algorithm the question of finding the exact approximation ratio was solved for Metric TSP. Moreover, we have shown that the approximation ratio of the k-Opt algorithm is between $\Omega(n^{\frac{2}{3k-3}})$ and $O\left(\sqrt[k]{n}\right)$ for the Metric TSP. If the Erdős girth conjecture holds, the upper bound would be tight. We do not expect that significant improvements can be made to close the gap without further understanding the behavior of $\mathrm{ex}(n, 2k)$.

Conjecture 4.1.1. *The approximation ratio of the k-Opt algorithm for the* Metric TSP *is $\Theta\left(\sqrt[k]{n}\right)$ where n is the number of vertices.*

Another open problem is finding lower bounds for the k-Lin-Kernighan algorithm. In contrast to the k-Opt algorithm the k-Lin-Kernighan algorithm considers changes involving an arbitrary number of edges. Hence, the key difficulty is that in this case it is not enough to construct an instance and argue that there are no improving alternating cycles up to a certain length.

Question 4.1.2. *Is the approximation ratio of the k-Lin-Kernighan algorithm asymptotically better than that of the k-Opt algorithm?*

For the Graph TSP we showed that the approximation ratio of the k-Opt algorithm is between $\Omega\left(\frac{\log(n)}{\log\log(n)}\right)$ and $O\left(\left(\frac{\log(n)}{\log\log(n)}\right)^{3.17}\right)$. For the upper bound we only analyzed the 2-Opt algorithm and carried the results over to the k-Opt and Lin-Kernighan algorithm as they also produce 2-optimal tours. It might be possible to analyze in a more complicated way the general k-Opt algorithm for the Graph TSP by using the same techniques and improve the approximation ratio for $k > 2$. However, we conjecture that the lower bound of $\Omega\left(\frac{\log(n)}{\log\log(n)}\right)$ is already tight for the 2-Opt algorithm. A refinement of our analysis or new techniques is needed to directly show this result.

Conjecture 4.1.3. *The approximation ratio of the 2-Opt algorithm for the* Graph TSP *is $\Theta\left(\frac{\log(n)}{\log\log(n)}\right)$ where n is the number of vertices.*

We showed that the approximation ratio of the k-Opt algorithm for the Euclidean TSP is $\Omega\left(\frac{\log(n)}{\log\log(n)}\right)$ which matches with the known upper bound and is asymptotically tight.

Although the approximation ratio of the k-Opt algorithm for the Graph TSP is not constant in the worst case, we were able to give a polynomial local search algorithm for the Graph TSP with constant approximation ratio. A natural question is if such an algorithm also exists for the Metric TSP and how good it would perform in practice.

Question 4.1.4. *Is there a local search algorithm for the* Metric TSP *with constant approximation ratio that needs polynomial time for every iteration?*

Another interesting question is the approximation ratio of the k-Opt and k-improv algorithm for the (1,2)-TSP. The approximation ratio of the 3-Opt algorithm is $\frac{11}{8}$ and we introduced the 3-Opt++ algorithm for the (1,2)-TSP with an approximation ratio of $\frac{4}{3}$. We have shown a lower bound of 1.1 on the approximation ratio of the k-Opt and k-improv algorithm for any fixed k. The current best upper bound is $\frac{8}{7}$ by Berman and Karpinski for the k-improv algorithm with $k = 15$. This is also the currently best approximation ratio for the (1,2)-TSP. It is expected that this upper bound can be further improved.

Question 4.1.5. *What is the approximation ratio of the k-Opt and k-improv algorithm with $k > 3$ for the* (1,2)-TSP*?*

4.2 Integrality Ratio of the Subtour LP

We have found Rectilinear TSP instances $(I_n^2)_{n\in\mathbb{N}}$ with high integrality ratio and shown certain properties of them. Assuming Conjecture 1.2.16 we were able to show that certain families of instances in the multidimensional rectilinear and metric case are the instances maximizing the integrality ratio. We conjecture that the families of instances $(I_n^2)_{n\in\mathbb{N}}$ and $(I_n^3)_{n\in\mathbb{N}}$ maximize the integrality ratio.

Conjecture 4.2.1. *The families of instances $(I_n^2)_{n\in\mathbb{N}}$ and $(I_n^3)_{n\in\mathbb{N}}$ maximize the integrality ratio for the* Rectilinear TSP *and the* Multidimensional Rectilinear TSP.

Note that this would also imply the $\frac{4}{3}$-Conjecture in the rectilinear and multidimensional rectilinear case as we have shown that the integrality ratios of I_n^2 and I_n^3 converge to $\frac{4}{3}$. By a local search algorithm, we found locally optimal instances with high integrality ratio for Euclidean TSP. We observe that these instances share similar structures as the instances with a small number of vertices maximizing the integrality ratio in the metric and rectilinear case. However, it is not clear if they maximize the integrality ratio and if so why they are embedded that way and have these observed properties.

Question 4.2.2. *What are the instances maximizing the integrality ratio among all instances with a fixed number of vertices for* Euclidean TSP*?*

We can further investigate locally optimal instances with other norms in arbitrary dimension with the local search algorithm we described.

Question 4.2.3. *How is the structure of instances with high integrality ratio found by the local search algorithm using the p-norm in $\mathbb{R}^d$?*

We improved the integrality ratio for the Metric $s-t$ Path TSP to 1.5273. The conjectured optimum in this case is 1.5. Since the choice of the auxiliary function h is already near-optimal, a new approach is needed to show this result.

Conjecture 4.2.4. *The integrality ratio of the standard LP relaxation for the* Metric $s-t$ Path TSP *is 1.5.*

Bibliography

[1] Anna Adamaszek, Matthias Mnich, and Katarzyna Paluch. New approximation algorithms for (1, 2)-TSP. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 9:1–9:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[2] Noga Alon, Shlomo Hoory, and Nathan Linial. The Moore bound for irregular graphs. *Graphs and Combinatorics*, 18(1):53–57, Mar 2002.

[3] Hyung-Chan An, Robert Kleinberg, and David B Shmoys. Improving Christofides' algorithm for the s-t path TSP. *Journal of the ACM (JACM)*, 62(5):1–28, 2015.

[4] David Applegate, Robert Bixby, Vasek Chvátal, and William Cook. *Concorde-03.12.19*, 2003 (last accessed March 19, 2020). `http://www.math.uwaterloo.ca/tsp/concorde/downloads/downloads.htm`.

[5] Richard Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.

[6] Genevieve Benoit and Sylvia Boyd. Finding the exact integrality gap for small traveling salesman problems. *Mathematics of Operations Research*, 33(4):921–931, 2008.

[7] Clark T. Benson. Minimal regular graphs of girths eight and twelve. *Canadian Journal of Mathematics*, 18:1091–1094, 1966.

[8] Jon J. Bentley. Fast algorithms for geometric traveling salesman problems. *ORSA Journal on computing*, 4(4):387–411, 1992.

[9] Piotr Berman and Marek Karpinski. 8/7-approximation algorithm for (1,2)-TSP. *Electron. Colloquium Comput. Complex.*, (069), 2005.

[10] Piotr Berman and Marek Karpinski. 8/7-approximation algorithm for (1, 2)-TSP. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 641–648. Society for Industrial and Applied Mathematics, 2006.

[11] Béla Bollobás. *Extremal graph theory.* Courier Corporation, 2004.

[12] Sylvia Boyd. *Vertices of the Subtour Elimination Polytope*, 2010 (accessed March 19, 2020). `http://www.site.uottawa.ca/~sylvia/subtourvertices/index.htm`.

[13] Sylvia Boyd and Robert Carr. Finding low cost TSP and 2-matching solutions using certain half-integer subtour vertices. *Discrete Optimization*, 8(4):525 – 539, 2011.

[14] Sylvia Boyd and Paul Elliott-Magwood. Structure of the Extreme Points of the Subtour Elimination Polytope of the STSP. In *RIMS Kôkyûroku Bessatsu B23*, pages 33–47, 2010.

[15] Ulrich A. Brodowsky and Stefan Hougardy. The approximation ratio of the 2-opt heuristic for the euclidean traveling salesman problem. *CoRR*, abs/2010.02583, 2020.

[16] William G. Brown. On graphs that do not contain a thomsen graph. *Canadian Mathematical Bulletin*, 9(3):281–285, 1966.

[17] Barun Chandra, Howard Karloff, and Craig Tovey. New results on the old k-opt algorithm for the traveling salesman problem. *SIAM Journal on Computing*, 28(6):1998–2029, 1999.

[18] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group, 1976.

[19] George Dantzig, Ray Fulkerson, and Selmer Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2(4):393–410, 1954.

[20] Vladimir G. Deineko, René van Dal, and Günter Rote. The Convex-Hull-and-Line Traveling Salesman Problem: A Solvable Case. *Information Processing Letters*, pages 141–148, 1994.

[21] Matthias Englert, Heiko Röglin, and Berthold Vöcking. Worst case and probabilistic analysis of the 2-opt algorithm for the TSP. *Algorithmica*, 68(1):190–264, Jan 2014.

[22] Paul Erdős and Alfréd Rényi. On a problem of graph theory. *Magyar Tudományos Akadémia Math. Kuató Int. Közl.*, 7:623–641, 1962.

[23] Paul Erdős, Alfréd Rényi, and Vera T. Sós. On a problem of graph theory. *Studia Scientiarum Mathematicarum Hungarica*, 1:215–235, 1966.

[24] Paul Erdős. Extremal problems in graph theory. In *Proc. Symp. Theory of Graphs and its Applications*, pages 29–36, 1963.

[25] Leonhard Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, pages 128–140, 1741.

[26] Merrill M Flood. The traveling-salesman problem. *Operations research*, 4(1):61–75, 1956.

[27] Lester R Ford Jr. Network flow theory. Technical report, Rand Corp Santa Monica Ca, 1956.

[28] Jacob Fox and Benny Sudakov. Decompositions into subgraphs of small diameter. *Combinatorics, Probability and Computing*, 19(5-6):753–774, 2010.

[29] M. R. Garey, R. L. Graham, and D. S. Johnson. Some NP-complete geometric problems. In *Proceedings of the Eighth Annual ACM Symposium on Theory of Computing*, STOC '76, pages 10–22, New York, NY, USA, 1976. ACM.

[30] Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

[31] Michel X Goemans and Dimitris J Bertsimas. Survivable networks, linear programming relaxations and the parsimonious property. *Mathematical Programming*, 60(1-3):145–166, 1993.

[32] Corinna Gottschalk and Jens Vygen. Better s–t-tours by Gao trees. *Mathematical Programming*, 172(1-2):191–207, 2018.

[33] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[34] JA Hoogeveen. Analysis of Christofides' heuristic: Some paths are more difficult than cycles. *Operations Research Letters*, 10(5):291–295, 1991.

[35] Stefan Hougardy. On the integrality ratio of the subtour lp for euclidean TSP. *Operations Research Letters 42*, pages 495–499, 2014.

[36] Stefan Hougardy, Fabian Zaiser, and Xianghui Zhong. The approximation ratio of the 2-opt heuristic for the metric traveling salesman problem. *Operations Research Letters*, 48(4):401 – 404, 2020.

[37] Stefan Hougardy and Xianghui Zhong. *Hard to Solve Instances of the Euclidean Traveling Salesman Problem*, 2018 (accessed October 06, 2020). `http://www.or.uni-bonn.de/%7Ehougardy/HardTSPInstances.html`.

[38] Stefan Hougardy and Xianghui Zhong. Hard to solve instances of the euclidean traveling salesman problem. *Mathematical Programming Computation*, 2020.

[39] David S. Johnson. Local optimization and the traveling salesman problem. In *International colloquium on automata, languages, and programming*, pages 446–461. Springer, 1990.

[40] Jovan Karamata. Sur une inégalité relative aux fonctions convexes. *Publications de l'Institut Mathématique*, 1(1):145–147, 1932.

[41] Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. A (slightly) improved approximation algorithm for metric TSP. *CoRR*, abs/2007.01409, 2020.

[42] Richard M. Karp. *Reducibility among combinatorial problems in: Raymond E. Miller, James W. Thatcher (Eds.), Complexity of Computer Computations*. Plenum Press, New York, 1972.

[43] Sanjeev Khanna, Rajeev Motwani, Madhu Sudan, and Umesh Vazirani. On syntactic versus computational views of approximability. *SIAM Journal on Computing*, 28(1):164–191, 1998.

[44] Dénes Kőnig. Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre. *Mathematische Annalen*, 77(4):453–465, 1916.

[45] Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer Publishing Company, Incorporated, 6th edition, 2018.

[46] Marvin Künnemann and Bodo Manthey. Towards understanding the smoothed approximation ratio of the 2-opt heuristic. In *International Colloquium on Automata, Languages, and Programming*, pages 859–871. Springer, 2015.

[47] Felix Lazebnik, Vasiliy A. Ustimenko, and Andrew J. Woldar. A new series of dense graphs of high girth. *Bulletin of the American mathematical society*, 32(1):73–79, 1995.

[48] Asaf Levin and Uri Yovel. Nonoblivious 2-opt heuristics for the traveling salesman problem. *Networks*, 62(3):201–219, 2013.

[49] Shen Lin and Brian W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2):498–516, 1973.

[50] Clyde L Monma, Beth Spellman Munson, and William R Pulleyblank. Minimum-weight two-connected spanning networks. *Mathematical Programming*, 46(1-3):153–171, 1990.

[51] Edward F Moore. The shortest path through a maze. In *Proc. Int. Symp. Switching Theory, 1959*, pages 285–292, 1959.

[52] Christos H. Papadimitriou. The Euclidean traveling salesman problem is NP-complete. *Theoret. Comput. Sci.*, 4(3):237–244, 1977.

[53] Ján Plesník. Bad examples of the metric traveling salesman problem for the 2-change heuristic. *Acta Mathematica Universitatis Comenianae*, 55:203–207, 1986.

[54] Gerhard Reinelt. TSPLIB—a traveling salesman problem library. *ORSA journal on computing*, 3(4):376–384, 1991.

[55] Gerhard Reinelt. *The traveling salesman: computational solutions for TSP applications*. Springer-Verlag, 1994.

[56] Daniel J Rosenkrantz, Richard E Stearns, and Philip M Lewis, II. An analysis of several heuristics for the traveling salesman problem. *SIAM journal on computing*, 6(3):563–581, 1977.

[57] András Sebő. Eight-fifth approximation for the path TSP. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 362–374. Springer, 2013.

[58] András Sebő and Anke Van Zuylen. The salesman's improved paths through forests. *Journal of the ACM (JACM)*, 66(4):1–16, 2019.

[59] A. I. Serdjukov. Some extremal bypasses in graphs [in Russian]. *Upravlyaemye Sistemy*, 17:76–79, 1978.

[60] Robert Singleton. On minimal graphs of maximum even girth. *Journal of Combinatorial Theory*, 1(3):306 – 332, 1966.

[61] Vera Traub and Jens Vygen. Approaching 3/2 for the s-t-path TSP. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1854–1864. SIAM, 2018.

[62] Vera Traub and Jens Vygen. An improved upper bound on the integrality ratio for the s–t-path TSP. *Operations Research Letters*, 47(3):225–228, 2019.

[63] Vera Traub, Jens Vygen, and Rico Zenklusen. Reducing path TSP to TSP. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 14–27. ACM, 2020.

[64] Jens Vygen. Reassembling trees for the traveling salesman. *SIAM Journal on Discrete Mathematics*, 30(2):875–894, 2016.

[65] Rephael Wenger. Extremal graphs with no C4's, C6's, or C10's. *Journal of Combinatorial Theory, Series B*, 52(1):113–116, 1991.

[66] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.

[67] David Paul Williamson. *Analysis of the Held-Karp heuristic for the traveling salesman problem (Master's Thesis)*. Massachusets Institute of Technology, 1990.

[68] Laurence A Wolsey. Heuristic analysis, linear programming and branch and bound. In *Combinatorial Optimization II*, pages 121–134. Springer, 1980.

[69] Rico Zenklusen. A 1.5-approximation for path TSP. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1539–1549. SIAM, 2019.

[70] Xianghui Zhong. On the Approximation Ratio of the k-Opt and Lin-Kernighan Algorithm for Metric and Graph TSP. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms (ESA 2020)*, volume 173 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 83:1–83:13, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[71] Xianghui Zhong. On the approximation ratio of the k-opt and Lin-Kernighan algorithm for metric and graph TSP. *CoRR*, abs/1909.12755, 2020.

[72] Xianghui Zhong. Slightly improved upper bound on the integrality ratio for the s-t path TSP. *Operations Research Letters*, 48(5):627 – 629, 2020.