

Master's Thesis
Masterstudiengang Physik



Phase Transitions of Disordered Travelling Salesperson Problems solved with Linear Programming and Cutting Planes

Hendrik Schawe

Betreuender Gutachter

Professor Dr. Alexander K. Hartmann

Gutachter

Professor Dr. Martin Fränzle

Oldenburg, November 6, 2015

Contents

List of Figures	5
1. Introduction	9
2. Complexity Classes	11
3. The Travelling Salesperson Problem	12
3.1. Introducing the Fuzzy Circle Ensemble	13
4. Linear Programming Approach to the TSP	16
4.1. Cutting Planes	21
4.2. Separation of Subtour Elimination Constraints	21
4.2.1. The Stör-Wagner Minimum Cut Algorithm	22
4.2.2. Karger’s Minimum Cut Algorithm	23
4.2.3. Implementation Details	24
4.3. Blossom Inequalities	24
4.4. Integer Programming	25
5. Short Introduction to Phase Transitions	29
5.1. Critical Exponents and Universality	29
5.2. Finite-Size Scaling	30
6. Results	32
6.1. Technical Details	32
6.2. Probability that LP Relaxations are Feasible	33
6.2.1. Degree LP Relaxation	34
6.2.2. SEC LP Relaxation	36
6.2.3. Fast Blossom LP Relaxation	36
6.3. Structural Properties	39
6.3.1. Difference to the Circle	39
6.3.2. Tortuosity	40
6.4. Is there Universality?	42
6.4.1. Gaussian Displacement	43
6.4.2. Spherical Displacement	43
6.4.3. Comparison to the Vertex Cover Transition	43
6.5. First Excitation: The Second Shortest Tour	45
6.6. The TSP Decision Problem	49
6.6.1. Uniformly Distributed Cities in High Dimensions	49
6.6.2. The Fuzzy Circle Ensemble	52

7. Conclusion	54
8. Outlook	54
A. Heuristics	56
A.1. Nearest Neighbor	56
A.2. Greedy	57
A.3. Farthest Insertion	57
A.4. Two-Optimal Tours	59
A.5. Analysis	59
B. Run Time Metrics	61
C. Program Flow Diagram	64
References	66
Glossary	72
Acronyms	74
Acknowledgments	77

List of Figures

2.1.	Venn Diagram of Complexity Classes	12
3.1.	Definition of the Displacement	13
3.2.	Example for the Displacement	14
	(a). $\sigma = 0$	14
	(b). $\sigma = 12$	14
	(c). optimized	14
3.3.	Examples of Tours for Different Values of σ	15
	(a). $\sigma = 0$	15
	(b). $\sigma = 10$	15
	(c). $\sigma = 20$	15
	(d). $\sigma = 40$	15
	(e). $\sigma = 80$	15
	(f). $\sigma = 160$	15
4.1.	Visualization of a Linear Program	17
4.2.	Example for the Explanaiton of the Constraints	19
	(a). No Constraints	19
	(b). Degree Constraints	19
	(c). Integer Constraint	19
	(d). Subtour Elimination Constraints	19
4.3.	Visualization of the Sets mentioned in the Blossom Inequalities	25
4.4.	Two Examples of Blossom Inequalities with $k = 3$	26
	(a). Blossom obeying the Inequality	26
	(b). Blossom violating the Inequality	26
4.5.	Visualization of an Integer Program	26
4.6.	Visualization of Cutting Planes	28
	(a). A Cutting Plane shrinks the Solution Space	28
	(b). A Cutting Plane leads to the Optimal Solution	28
6.1.	Probability p that the Degree LP Relaxation is Feasible	35
	(a). $p(\sigma)$	35
	(b). Data Collapse	35
	(c). $\text{Var}(p)$, inset: fit to $\sigma_{\max} = aN^{-b^{\text{lp}}} + \sigma_c^{\text{lp}}$ for $N \geq 512$	35
6.2.	Probability p that the SEC LP Relaxation is Feasible, i.e. Integer	37
	(a). $p(\sigma)$	37
	(b). Data Collapse	37
	(c). $\text{Var}(p)$, inset: fit to $\sigma_{\max} = aN^{-b^{\text{cp}}} + \sigma_c^{\text{cp}}$ for $N \geq 256$	37
6.3.	Probability p that the Fast Blossom LP Relaxation is Feasible, i.e. Integer	38
	(a). $p(\sigma)$	38

(b).	Data Collapse	38
(c).	Var(p), inset: fit to $\sigma_{\max} = aN^{-b^{\text{fb}}} + \sigma_c^{\text{fb}}$ for $N \geq 256$	38
6.4.	Difference of the Optimal Tour to the Initial Circle as a Function of σ	39
6.5.	Tortuosity of some Example Curves	40
(a).	Circle	40
(b).	Line	40
(c).	Two Half Circles	40
(d).	Two Three Quarter Circles	40
(e).	Four Half Circles	40
6.6.	Determining Segments for the Tortuosity Calculation	41
(a).	Determining Segments for the Tortuosity Calculation	41
(b).	Example for the Calculation of the Tortuosity τ	41
6.7.	Tortuosity for Different Values of N as a Function of σ	42
6.8.	Probability p that the SEC LP Relaxation is Feasible with Gaussian Displacement	44
(a).	$p(\sigma)$	44
(b).	Data Collapse	44
(c).	Var(p), inset: fit to $\sigma_{\max} = aN^{-b^{\text{cp,g}}} + \sigma_c^{\text{cp,g}}$ for $N \geq 256$	44
6.9.	Tortuosity for Different Values of N as a Function of σ with Gaussian Displacement	45
6.10.	Probability p that the SEC LP Relaxation is Feasible with Displacement in Three Dimensions	46
(a).	$p(\sigma)$	46
(b).	Data Collapse	46
(c).	Var(p), inset: fit to $\sigma_{\max} = aN^{-b^{\text{cp,3}}} + \sigma_c^{\text{cp,3}}$ for $N \geq 256$	46
6.11.	Tour Difference d between Shortest and Second Shortest Tour	48
(a).	Tour Difference d with Power-Law Fits	48
(b).	Exponents of the Power-Laws of (a)	48
6.12.	Cumulative Distribution of Tour Length L and Control Parameter k	50
(a).	Tour Length L	50
(b).	Control Parameter $k = L/(NA)^{1-1/D}$	50
6.13.	Finite-Size Scaling Analysis by Data Collapse	51
6.14.	Cumulative Distribution of Tour Length L and Visualization of the Crossover	52
(a).	$\sigma = 2$	52
(b).	$\sigma = 18$	52
6.15.	Finite-Size Scaling Analysis by Data Collapse	53
A.1.	Comparison of Different Heuristics	56
A.2.	Examples of Different Heuristics	58
(a).	Next Neighbor	58

(b).	Greedy	58
(c).	Farthest Insertion	58
(d).	Two-Opt Starting from Next Neighbor	58
(e).	Two-Opt Starting from Random	58
(f).	Optimal Tour	58
A.3.	Example of a 2-opt Swap	59
(a).	Before 2-opt Swap	59
(b).	After 2-opt Swap	59
A.4.	Solution Probability of Simple Heuristics	60
(a).	Farthest Insertion	60
(b).	Farthest Insertion and Two-Opt	60
B.1.	Number of Cuts	62
(a).	Number of Cuts at Different Values of σ for all Instances	62
(b).	Number of Cuts at Different Values of σ only if Solvable by LP	62
(c).	Number of Cuts at Different System Sizes	62
B.2.	Branch-and-Cut Run Time Statistics	63
(a).	Number of Nodes n_n over σ	63
(b).	Number of Cuts n_c over σ	63
(c).	Median of Number of Nodes n_n over N	63
C.1.	Program Flow to find Violated SEC	65

1. Introduction

According to Ref. [1] Karl Menger was the first mathematician to publish studies on the Travelling Salesperson problem (TSP) around the year 1930. It seems appropriate to let him define the TSP, which he called *Botenproblem* [2, p. 23].

We denote by messenger problem (since in practice this question should be solved by each postman, anyway also by many travelers) the task to find, for finitely many points whose pairwise distances are known, the shortest route connecting the points. Of course, this problem is solvable by finitely many trials. Rules which would push the number of trials below the number of permutations of the given points, are not known. The rule that one first should go from the starting point to the closest point, then to the point closest to this, etc., in general does not yield the shortest route.¹

In the first sentence Menger noted that the TSP is of great practical relevance. A good tour optimization can not only save money when used for vehicle routing, but also circuit boards can be produced faster and therefore cheaper if the drill uses the shortest way to drill the holes to contact back and front of the board, and there are many more real world applications [3]. It should be noted that the TSP has also a history as a testbed for optimization heuristics, e.g. the effectiveness of *simulated annealing* [4], *taboo search* [5] and *ant colony algorithms* [6] were shown early on examples of TSP instances. The TSP is suited for this purpose because its fame led to a rich collection of tours with their optimal solutions as a benchmark, notably the TSPLIB², to estimate the quality of the tested heuristic and – as the last sentence of the cite says – no effective algorithm is known to solve a TSP to optimality. In fact it is proven that the TSP is **NP-hard**³ which means that there is no easy way to solve the TSP yet and it is reasonable to think that it will never be the case. This is connected to the famous **P-NP** problem. Its solution would determine this uncertainty But for practical needs there exist good heuristics [6, 8, 9] and for the Euclidean case (which is still **NP-hard** [10]), i.e. the pairwise distances are the Euclidean distances, even a polynomial-time approximation scheme (PTAS) [11]. Also many polynomial-time algorithms for special corner cases are known [12].

The complexity class **NP-hard** is of great practical relevance since a wide range of important optimization problems belong to it, including spin glass ground-state

¹Translated from the German original by Alexander Schrijver in [1].

²TSPLIB is a collection of real world, typically difficult TSP problems and their optimal solutions [7].

³See Section 2.

calculations [13] which are relevant to condensed matter physics, integer programming [14] which is important to operations research and as already mentioned the TSP which needs to be optimized in every logistics context.

To gather insight in computational “hardness”, systems from the class **NP**-hard are studied empirically. Interestingly **NP**-complete⁴ problems, which are a subset of **NP**-hard, often show some kind of easy-hard phase transition regarding the typical behavior on suitably parameterized ensembles of random instances. Some of the classical **NP**-complete problems [14] from theoretical computer science were examined with respect to phase transitions with methods of statistical mechanics of unordered systems in [15–20] and many more. Especially, in Ref. [21] the TSP decision problem, if there exists a tour shorter than a given length, is under scrutiny and shows a phase transition.

This thesis searches for a phase transition in the TSP when changing the configurations of the cities the salesperson has to visit. To force some kind of transition the configurations start at an easy to solve instance, e.g. cities arranged on a circle. Each city is then displaced by a random distance dependent on a control parameter σ . A main point of reference for this thesis is Ref. [22], which examines the **NP**-hard Vertex Cover (VC) optimization problem on Erdős-Rényi graphs, which can be tuned with the *connectivity*, also known as *edge density*, parameter c . There the transition point obtained by analytic means [18] can also be observed when examining whether the linear programming (LP) has a valid solution or not. This will be the vector of attack to find a phase transition in this thesis, and allows to look at relatively large TSP realizations.

This thesis starts with the short introduction of complexity classes in Section 2, the Travelling Salesperson problem and the ensemble of realizations under scrutiny in this thesis in Section 3, linear programming in Section 4 and the methods of statistical mechanics of unordered systems in Section 5.

The main part Section 6 shows the results of the computations. Section 6.2 shows the occurrence of an phase transition from “easily solvable by LP” to “not solvable by LP” and Section 6.3 shows properties of optimal tours, which show some special behavior at the phase transition. Section 6.4 shows that this phase transition exhibits universality as expected. In Section 6.5 a look at ground-state excitations is taken and finally Section 6.6 reproduces and extends some results from earlier work before Sections 7 and 8 give a conclusion of this thesis and an outlook to possible ways to extend it.

⁴See Section 2.

2. Complexity Classes

The introduction mentioned that phase transitions are observable in many problems of the **NP**-complete complexity class. Since there are many misconceptions about complexity classes among – but not limited to – physicists, it seems appropriate to give a short introduction to complexity classes. Since it is far beyond the scope of this thesis to introduce formal details about Turing machines, only some non-rigorous, phenomenological definitions of some complexity classes are given, which should suffice for the understanding of this thesis⁵.

First a few fundamental definitions of complexity are needed. The most common way to compare different algorithms is the comparison of the *time complexity*, which basically counts the number of fundamental operations of the given algorithms. It is given as a function of the input size n for asymptotic big inputs with a Landau symbol, i.e. disregarding everything but the highest order term of the function – also commonly referred to as *big-Oh-notation*. For example, in the worst case a naive Bubble Sort⁶ of n values needs $\mathcal{O}(n^2)$ comparisons, whereas Merge Sort needs only $\mathcal{O}(n \log n)$. Therefore Merge Sort is – at least for large n – the superior algorithm. If the time complexity is bounded by a polynomial, like both examples, the algorithm has a *polynomial-time* complexity.

If a *decision problem*, i.e. a problem which solution is either “yes” or “no”, is solvable in polynomial time, it belongs to the **P** class, which stands for polynomial time. If there is a way to verify the solution of a decision problem in polynomial time given a *certificate* of the solution, it is in **NP**, which stands for non-deterministic polynomial time. For example the problem to decide whether a number is composite, i.e. not prime, is in **NP**⁷ because given the prime factors as the certificate one can multiply them in polynomial time to verify that the resulting number is indeed composite. So clearly

$$\mathbf{P} \subset \mathbf{NP}$$

holds, because if a problem is solvable in polynomial time, one can verify the solution in polynomial time by solving it again. Whether it is a proper subset or $\mathbf{P} = \mathbf{NP}$ is a famous open problem called **P-NP** problem, e.g. one of the *Millennium Problems* [27]. Though in the following it is assumed that **P** is a proper subset.

A subset of **NP** is **NP**-complete, which consists of decision problems to which

⁵For further reading Ref. [23] or similar books are recommended.

⁶This example is chosen because sorting seems to be the prime example for complexity analysis in every undergrad textbook. The definition of Bubble Sort and Merge Sort will not be given here, and the interested reader is referred to any textbook, e.g. Ref. [24] or for even more in depth analysis Ref. [25].

⁷Note that it is also in **P** because primality tests can be performed in polynomial time [26].

every other problem in **NP** can be reduced in polynomial time, especially all **NP**-complete problems can be reduced to each other. Therefore they are often referred to the hardest problems in **NP**, since if one of them can be solved in polynomial time, all of **NP** can be solved in polynomial time. **NP**-complete has as famous members as Boolean satisfiability, vertex cover, integer programming (IP) for binary variables without optimization and the decision versions of the TSP – is there a tour shorter than a given length – as proven in Ref. [14].

The last class with relevance in this thesis is **NP**-hard, which consists of not necessarily decision problems to which any **NP** problem can be reduced in polynomial time and therefore

$$\mathbf{NP}\text{-complete} = \mathbf{NP} \cap \mathbf{NP}\text{-hard}.$$

For example, the optimization versions of the vertex cover or the TSP, which solutions also answer the corresponding decision problems, which are in **NP**-complete. But also other practically relevant problems like ground states of spin glasses [13] or IP belong to **NP**-hard. Note that the solution of an **NP**-hard problem can only be checked for feasibility but not for optimality in polynomial time. The relations of these complexity classes are also sketched in Figure 2.1 for clarity.

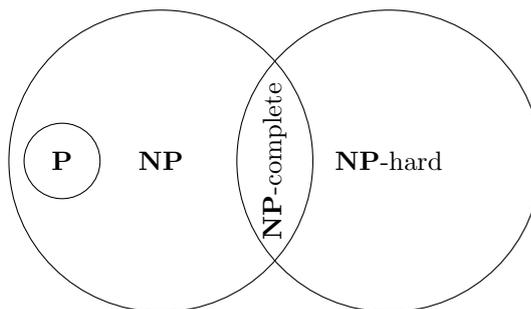


Figure 2.1: Venn diagram to clarify the relationships between the complexity classes that are important in the context of this thesis.

3. The Travelling Salesperson Problem

Instead of the historical formulation, a more formal definition of the TSP will be given in the context of graph theory. This has the advantage that short and precise terms can be used.

For a complete, undirected graph $G(V, E)$ with a given set of vertices V and given edge weights c_{ij} find the minimum weight Hamiltonian cycle, i.e. a closed tour visiting all vertices exactly once.

Each vertex corresponds to a city and the edge weights correspond to the pairwise distances. The length of the tour is the weight of the Hamiltonian cycle.

If the distance matrix c_{ij} is symmetric, i.e. the way $A \rightarrow B$ is as long as $B \rightarrow A$ thus $c_{AB} = c_{BA}$, it is called symmetric TSP. If the distances are derived from any metric and thence the triangle inequality holds, i.e. a tour $A \rightarrow B \rightarrow C$ is always longer or equal $A \rightarrow C$ thus $c_{AB} + c_{BC} \geq c_{AC}$, it is called metric TSP.

3.1. Introducing the Fuzzy Circle Ensemble

In this thesis a new random ensemble of city configurations, i.e. distance matrices c_{ij} , is introduced. I name it *Fuzzy Circle Ensemble (FCE)*, which loosely visualizes the realizations. The instances of this ensemble are Euclidean TSP, i.e. cities have coordinates on a plane (respectively in space for dimensions higher than two) and the edge weights are the Euclidean distances between them. This has the advantage that realizations are easy to visualize and comprehend. The FCE exhibits one parameter σ to tune the geometric properties of the cities. To construct realizations of the FCE, first the cities are arranged in a well defined start configuration and then randomly displaced depending on the parameter σ .

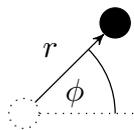


Figure 3.1: Definition of the displacement. The cities are displaced by a random distance $r \in U[0, \sigma]$ in a random direction $\phi \in U[0, 2\pi)$. Both random variables are independent and uniformly distributed.

The start configuration is chosen as N cities lying on a circle like Figure 3.2(a), because that seems to be the easiest configuration to spot the minimum length tour, as even the most simple heuristics, e.g. *nearest neighbor*, which is the heuristic proposed by Karl Menger cited in the introduction, finds the optimal tour. Further some polynomial-time solution methods work, since the circle fulfills the necklace condition [28] and all points are part of the convex hull [29].⁸

The radius of the circle is chosen as $R = \frac{N}{2\pi}$, such that the inter-city distance is approximately 1 and exactly 1 for $N \rightarrow \infty$. This establishes two length scales, the scale of the circle and the scale of the distance to the neighbor city.

⁸Of course these properties are also fulfilled by, e.g. a rectangle.

For each city a displacement is determined by two independent random variables from an uniform distribution. $\phi \in U[0, 2\pi)$ is treated as a displacement angle and $r \in U[0, \sigma]$ as a radius. This is sketched in Figure 3.1 and shown with an example in Figures 3.2(a) and 3.2(b).

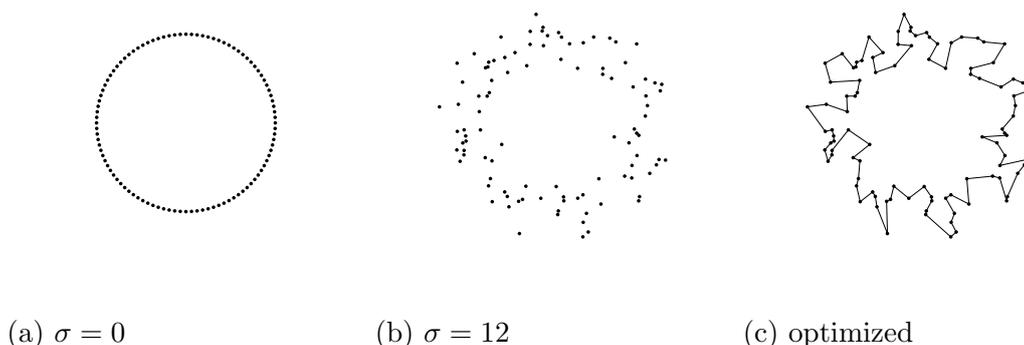


Figure 3.2: Configurations with $N = 180$ cities (a) before and (b) after the displacement step. (c) is the optimal tour of (b).

Though the choices of the start configuration and the modus of displacement are arbitrary, they should show no qualitative differences to, say, a rectangle as the starting configuration, as will be explained later in Section 5.1 and shown in Section 6.4.1, where instead of the uniformly distributed random variables r and ϕ , normal distributed Δx and Δy are used. Also instead of open boundary conditions, i.e. an infinite plane, periodic boundary conditions, i.e. a torus, could be chosen, which does not change the results fundamentally (which was indeed briefly tested, but will not be subject of this thesis).

To get a better understanding what the construction rules mean for the configurations and the optimal tours, Figure 3.3 shows optimal solutions for six different values of σ calculated by Concorde⁹. The examples have $N = 1024$ cities hence a radius of $R = \frac{N}{2\pi} \approx 163$. Therefore Figure 3.3(f) with a maximal displacement of $\sigma = 160$ closes the “hole” in the middle. The most fundamental change happens between Figures 3.3(a) and 3.3(b) where the solution changes from obvious to possibly difficult. The tours in Figures 3.3(c) to 3.3(f) seem not to be fundamentally different from Figure 3.3(b). The expectation is therefore that the transition from easy to hard – if it exists – happens in the range $0 < \sigma < 10 \ll R$ for this example with $N = 1024$, which suggests that the structure of the big circle and therefore

⁹Concorde is a state of the art solver for the TSP, free for academic use and open source. <http://www.math.uwaterloo.ca/tsp/concorde.html>

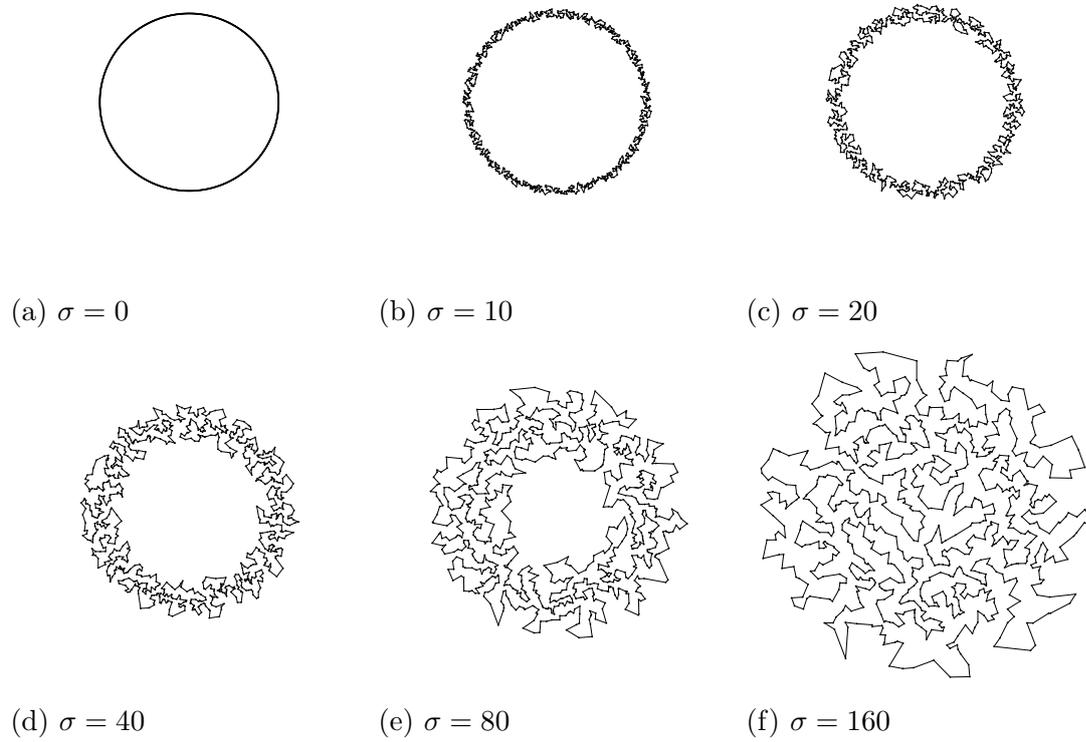


Figure 3.3: Evolution of a $N = 1024$ and $R = \frac{N}{2\pi} \approx 160$ system with increasing σ .

N is not important for this transition. Therefore the same values of σ will be examined for every considered number of cities N .

4. Linear Programming Approach to the TSP

LP is a term for techniques used to solve optimization problems which minimize or maximize a linear *objective function* with respect to restrictions formulated using *linear inequalities*, called *constraints*. The most general formulation using the coefficient vector \mathbf{c} , the coefficient matrix \mathbf{A} , the bound vector \mathbf{b} and the variable vector \mathbf{x} is

$$\begin{aligned} & \text{maximize} && \mathbf{c}^T \mathbf{x} \\ & \text{subject to} && \mathbf{A} \mathbf{x} \leq \mathbf{b}. \end{aligned}$$

Note that equalities and “ \geq ” can also be expressed by this formulation e.g.

$$\begin{aligned} -ax_1 \leq -b & \Leftrightarrow ax_1 \geq b \\ \left. \begin{array}{l} ax_1 \leq b \\ ax_1 \geq b \end{array} \right\} & \Leftrightarrow ax_1 = b \end{aligned}$$

The constraints divide the solution space in feasible and infeasible regions. The feasible region is always a convex polytope due to the linearity of the constraints. An example for

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} \frac{4}{9} & 1 \\ 1 & \frac{1}{5} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 5 \\ 2.5 \end{pmatrix}$$

is sketched in Figure 4.1.

The solution lies always on the boundary, i.e. the sides, of the polytope. Because the objective function is linear, it is sufficient to examine the corners, i.e. vertices, of the polytope to find the optimum [30]. The most intuitive way to find the optimum would be to start at one vertex and move from there to adjacent better vertices until the optimum is obtained, because no adjacent vertex is better. Since the polytope is convex, this algorithm can not be stuck in a local minimum. This algorithm is called *simplex algorithm*. Because I did not implement one myself for this thesis, I reference Ref. [30] for the reader interested in technical details, e.g. how it can be assured that it does not get stuck in a cycle through equally good vertices.

Although the simplex algorithm does not have a polynomial worst-case run time, it is of great practical relevance, since the typical run time is not bad and it offers *warmstart* capabilities [31] which are useful if many closely related problems need to be solved, which is a common case as will be shown in Section 4.1. It should further be noted, that there are other algorithms and that the *ellipsoid* algorithm

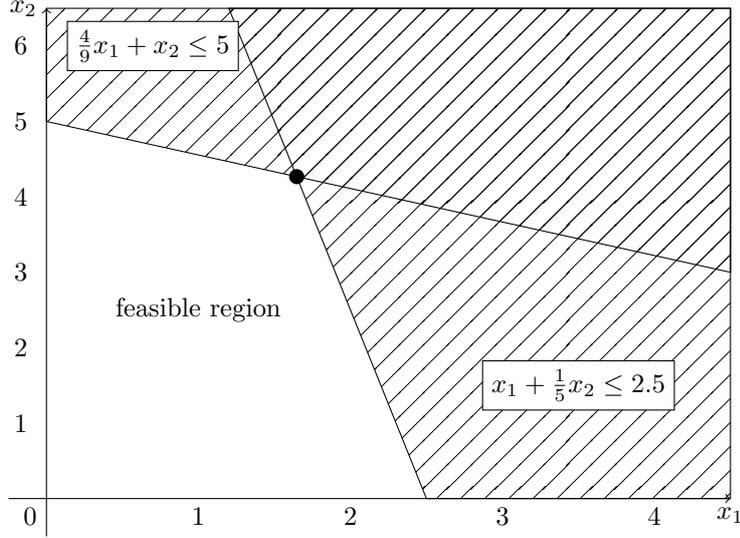


Figure 4.1: Visualization of a LP. The hatched region is infeasible. All solutions which maximize a linear objective function lie on the lines defined by $\frac{4}{9}x_1 + x_2 = 5$, $x_1 + \frac{1}{5}x_2 = 2.5$. In fact it is sufficient to examine the intersections, to find the maximum, which is what the simplex algorithm does. The maximum of $x_1 + x_2$ is marked with a dot.

[30] and some *interior-point* algorithms [24] solve the same problem in polynomial time, which is especially important for the complexity analysis. The careful reader will now note, that the TSP which is **NP**-hard thus, if $\mathbf{P} \neq \mathbf{NP}$ holds, can not be solved by LP alone. The portion of the solution process which needs more than polynomial run time will be introduced in Section 4.4.

One possible formulation of the symmetric TSP as a LP is given in Equations (4.1) to (4.4) as first introduced by Ref. [32]. This formulation will be explained in detail in the remainder of this section. To simplify the notations in this section, the variables are identified $x_{ij} \equiv x_{ji}$ due to the symmetry.

$$\text{minimize} \quad \sum_i \sum_{j < i} c_{ij} x_{ij} \quad (4.1)$$

$$\text{subject to} \quad x_{ij} \in \{0, 1\} \quad (4.2)$$

$$\sum_j x_{ij} = 2 \quad \forall i \in V \quad (4.3)$$

$$\sum_{i \in S, j \notin S} x_{ij} \geq 2 \quad \forall S \subset V, S \neq \emptyset, S \neq V \quad (\text{SEC}) \quad (4.4)$$

The idea is to take all possible connections between city i and j as the variables x_{ij} . Thus \mathbf{x} is the adjacency matrix of the tour. If cities i and j are directly connected in the tour, the variable x_{ij} should be 1, else 0. Because the graph symbolizing the

tour is undirected, the adjacency matrix is symmetric, i.e. $x_{ij} = x_{ji}$. The cyclic tour (1, 2, 4, 3, 5) is represented by

$$x_{ij} = \begin{pmatrix} \cdot & 1 & 0 & 0 & 1 \\ 1 & \cdot & 0 & 1 & 0 \\ 0 & 0 & \cdot & 1 & 1 \\ 0 & 1 & 1 & \cdot & 0 \\ 1 & 0 & 1 & 0 & \cdot \end{pmatrix}.$$

Because it is not clear what a non-integer entry in an adjacency matrix means, Equation (4.2) restricts x_{ij} to the integers 0 and 1. (Values greater than 1 are actually forbidden by the combination of constraints of type Equations (4.3) and (4.4), but for clarity this bound is given separately.)

Since the TSP searches the minimum length tour, the objective function to minimize Equation (4.1) calculates the tour length, with c_{ij} being the distance between city i and j , i.e. $c_{ij} = \text{dist}(i, j)$.

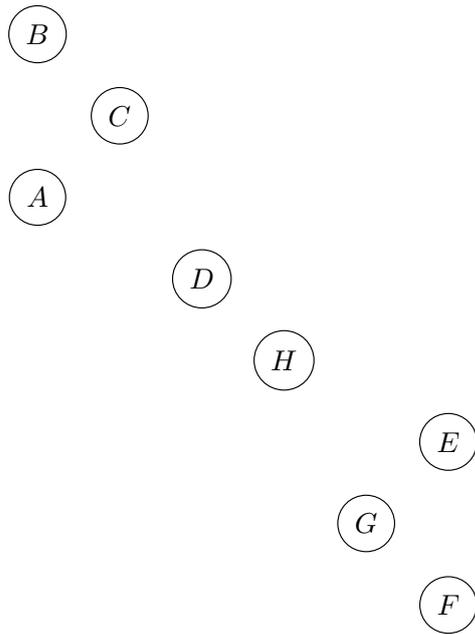
Obviously every city needs to be connected to two other cities, such that the salesperson can enter and leave it. Formulated for the matrix representation, the sum over the i -th row needs to be 2 as stated by the *degree equations* Equation (4.3). Because of the symmetry also the sum over the i -th column needs to equal 2. However, not only one single tour satisfies this constraint, but several not connected loops, called *subtours*, do, too. An example is sketched in Figures 4.2(b) and 4.2(c).

The subtour elimination constraints (SECs) Equation (4.4) suppress the subtours by ensuring that for every non-empty, proper subset $S \subset V$ of the cities a minimum of two ways connect the cities in the subset $i \in S$ with cities not in subset $j \notin S$. This introduces a new problem, as the “for all subsets” part hints at an exponentially increasing number of constraints for larger system sizes N . In fact not all SECs are formulated in the beginning. The strategy to add them on demand will be explained in Section 4.1. As an example for the way the SECs work, take Figure 4.2(c) which violates the SEC with the set $S = \{A, B, C, D\}$ because

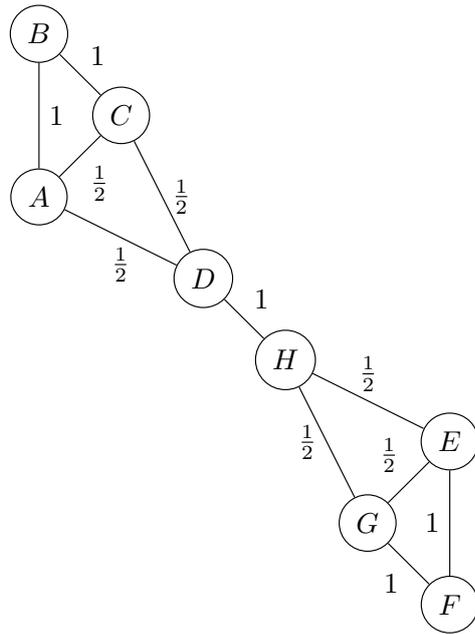
$$\sum_{i \in S, j \notin S} x_{ij} = 0 \not\geq 2.$$

As shown in Figure 4.2(b) the SECs also suppress other invalid configurations, which can arise when disregarding the integer constraints Equation (4.2). In that example the SEC for the set $S = \{A, B, C, D\}$ is violated because

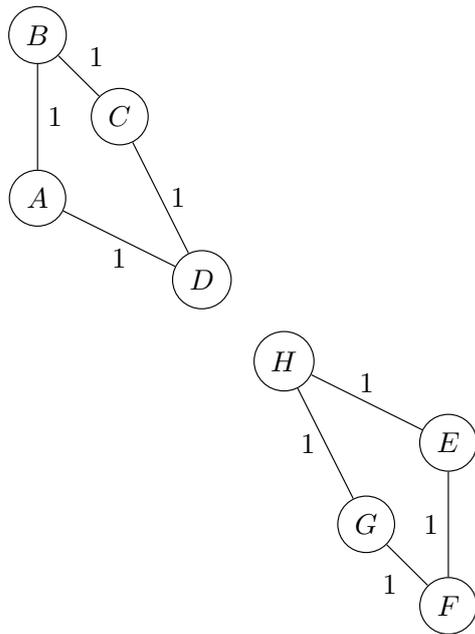
$$\sum_{i \in S, j \notin S} x_{ij} = 1 \not\geq 2.$$



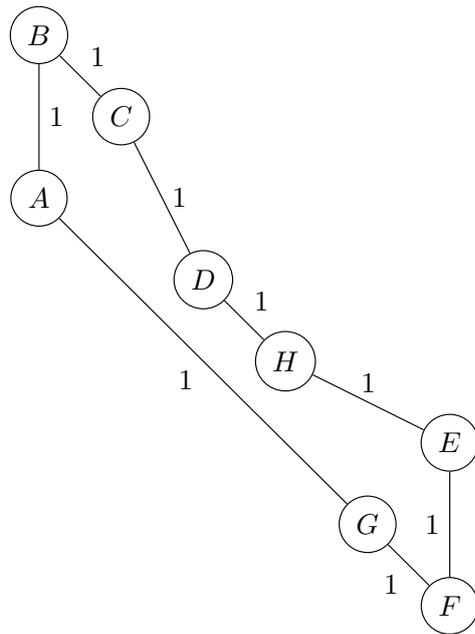
(a) $L = 0$



(b) $L = 4 + 4\sqrt{2} + 2\sqrt{5} \approx 14$



(c) $L = 4 + 2\sqrt{2} + 4\sqrt{5} \approx 16$



(d) $L = 4 + 7\sqrt{2} + 2\sqrt{5} \approx 18$

Figure 4.2: Optimal solutions obeying subsets of the constraints, to be precise (a) is without any constraints, (b) has only the degree constraints, (c) additionally the integer constraints, (d) complies also with the SECs and is an optimal tour. The lengths L are given with the distance $\overline{AB} = 2$.

But beware that the SECs can not suppress all non-integer solutions.

There are two other formulations equivalent to the SECs [32], which are very similar to each other and are mentioned here just for completeness sake.

$$\sum_{i < j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset V, S \neq \emptyset, S \neq V \quad (4.5)$$

$$\sum_{i < j \notin S} x_{ij} \leq N - |S| - 1 \quad \forall S \subset V, S \neq \emptyset, S \neq V \quad (4.6)$$

The idea behind Equation (4.5) is that the number of edges between the vertices of a subset S – under the premise that the degree inequalities are satisfied – is only equal to the number of nodes in this subset $|S|$ if there is a loop. This can easily be verified at Figure 4.2(c) with $S = \{A, B, C, D\}$

$$\sum_{i, j \in S} x_{ij} = 4 \not\leq 3 = |S| - 1.$$

Also note that this relation is violated by the same S in Figure 4.2(b)

$$\sum_{i, j \in S} x_{ij} = 3.5 \not\leq 3 = |S| - 1.$$

The second Equation (4.6) has the same idea supplemented by the fact that the solution is a loop and needs therefore N edges. The examples how these constraints are violated in Figures 4.2(b) and 4.2(c) are left as exercise for the reader.

The major problem of this formulation is the integer constraint Equation (4.2). Since it is not linear, the LP mechanism can not handle it. Until a strategy to mitigate this shortcoming is presented in Section 4.4, instead the IP is *relaxed*, by ignoring the integer constraints Equation (4.2), i.e. replace it by

$$x_{ij} \in [0, 1]. \quad (4.7)$$

The resulting LP problem is called LP relaxation. Since there are a few other relaxations, generated by dropping other constraints or adding further inequalities, under scrutiny in this thesis, this relaxation will be called SEC LP relaxation for clarity. One can verify that the solution of any LP relaxation is always better or equal than the solution with integer variables and thus is a lower bound to the minimum tour length. If a solution of the SEC LP relaxation version has integral x_{ij} , it is also a solution to the TSP.

For completeness it should be mentioned that the SECs are not the only way to prevent subtours in the solution. An easy to implement alternative is the Miller-Tucker-Zemlin formulation [33]. It only needs a polynomial amount of constraints, but enforces the integrality constraint Equation (4.2) at an earlier point in the solution process and is thus not suited for the kind of analysis this thesis provides.

4.1. Cutting Planes

The idea behind cutting planes is to not add all constraints at the definition of the problem but during the solution process. They are added in an iterative way. First the relaxed problem without some class of constraints is solved, e.g. by the simplex method, then a constraint is found that invalidates the region of the solution space containing the current solution. This process is called *separation* and the constraint is the *cutting plane* – or short *cut*. It is added to the LP relaxation and the procedure is repeated. A famous example are *Gomory Cuts* [30] which add constraints which ensure that all variables are integer after sufficiently many iterations, but unfortunately need often too many iterations to be useful. An example where cutting planes lead to the solution of a simple integer program is shown in Figure 4.6.

To use the SECs as cutting planes, a SEC violated by the current solution needs to be found. The separation of SECs will be explained in Section 4.2.

There do exist more cutting planes for the TSP, which can be added to the problem to enforce the integrality constraints like *Comb Constraints* [34–36]. But there are no polynomial-time algorithms known yet, to separate all of them, but only subsets like *simple comb inequalities* [37, 38]. Nevertheless in more sophisticated solvers like Concorde those advanced techniques are used, as stated by Ref. [39].

There are also generic cuts which can be used for most IP problems and are often implemented in sophisticated solvers to reduce the solution space. Although the CPLEX¹⁰ version used offers 10 different types of cuts, including fractional Gomory, clique and zero-half cuts, none of them will be used here. They are clearly inferior to SECs and using only SECs makes it possible to evaluate statistics of the cuts, which would be diluted by generic cuts otherwise.

4.2. Separation of Subtour Elimination Constraints

If one divides the map on which the TSP lives into two parts, the SECs enforce that connections with the summed value ≥ 2 have to cross the border between those parts. If formulated in the context of graphs it becomes apparent how to find violations of this rule. The *global minimum cut*¹¹ of the corresponding graph

¹⁰CPLEX is a commercial LP and mixed integer programming (MIP) solver from IBM, which can be used as a C++ library and is free for academic use. <http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud/>

¹¹Though the names are deceptively similar, this may not be confused with the aforementioned cutting planes.

must be obtained. Once the minimum cut of the graph is found, the sum of the weights of the cut edges, also called the *weight of the cut*, indicates whether a SEC is violated. The two sets on either side of the cut indicate which SEC is violated. If the value is ≥ 2 there are no SECs violated, else a cutting plane is added with the violated SEC and the LP relaxation is solved again. Fortunately the minimum cut problem is well studied and there exist a plethora polynomial-time algorithms which can find the global minimum cut. In this study the Stör-Wagner algorithm [40] explained in Section 4.2.1 is used, provided by the Boost Graph Library (BGL).

The special case of a minimum cut of 0, i.e. a graph with more than one connected component, can be detected by a simple depth first search (DFS) [41] in time $\mathcal{O}(|V| + |E|)$ where $|V|$ is the number of vertices and $|E|$ the number of edges in the graph.

To speed up the solution process, it is advantageous to find more than one violated SEC in every iteration, since it may reduce the total number of needed iterations. Therefore Karger’s algorithm [42] explained in Section 4.2.2 will be used to find some random small cuts.

The logical conclusion of this line of thinking would be to find all minimum cuts between all pairs of nodes. Therefore a Gomory-Hu-Tree [43] provided by the Library for Efficient Modeling and Optimization in Networks (LEMON) was considered, but tests showed that this strategy added too many cuts slowing down the solution process, therefore the Gomory-Hu-Tree is not constructed in the final program. The details of the chosen implementation are explained in Section 4.2.3.

Note that because the separation needs just polynomial time, the solution of this LP relaxation is therefore also possible in polynomial time using the ellipsoid algorithm [44–46]. Also Appendix B shows numerical statistics of the number of cuts used to generate the results of this study and shows that in fact only polynomial many are needed in the simulations.

4.2.1. The Stör-Wagner Minimum Cut Algorithm

The Stör-Wagner algorithm [40] finds for an undirected edge-weighted graph $G(V, E)$ with non-negative weights the global minimum cut in time

$$\mathcal{O}(|V||E| + |V|^2 \log |V|)$$

without using the famous *max-flow min-cut* theorem. The idea is that either the global minimum cut of G separates the nodes s and t , then the minimum s - t -cut, i.e. the minimum cut which separates the vertices s and t in distinct sets, is the

global minimum cut, or the global minimum cut of G is the global minimum cut of G where s and t are merged into one vertex.

The main part is the so called *minimum cut phase*, which finds an arbitrary s - t -min-cut, i.e. s and t are not predetermined, by a *maximum-adjacency search*. For the maximum-adjacency search a set A with an arbitrary seed vertex a is constructed. The vertex most tightly connected to A is added to A , i.e. the vertex $v_0 \notin A$ whose sum of the edge weights to vertices $v \in A$ is maximal. This is repeated until all but one vertex of G are in A , the last added vertex is called s and the remaining t . Note that this does not alter the graph structure. The sum of the weights of the edges incident to t is the minimum s - t -cut, called *cut-of-the-phase*, which is stored. Now the vertices s and t are merged into one new vertex u . The weights of the edges connecting u to the remainder of the graph are

$$w(\{u, i\}) = w(\{s, i\}) + w(\{t, i\}) \quad \forall i \in V \setminus \{s, t\}$$

This is repeated $|V|$ times until all vertices are merged into one. The lightest encountered *cut-of-the-phase* is the global minimum cut.

4.2.2. Karger's Minimum Cut Algorithm

Karger's algorithm [42] is a randomized algorithm that finds some small cut in time

$$\mathcal{O}(|E| \log^2 |V|)$$

which is especially suited for this application, since there are only $|E| = \mathcal{O}(|V|)$ edges in the graph because of the degree constraints. It works on undirected multigraphs, i.e. a graph on which multiple edges between the same two vertices are allowed, but no self loops. Also all edge weights need to be 1 – integer edge weights can be incorporated by multiple edges.

For the application in this thesis also fractional edge weights are used, which will invalidate the estimate of the probability to find the global minimum cut, which is not important in the scope of this application.

It chooses a random edge $e = \{i, j\} \in E$ and merges the two incident vertices i and j discarding all edges between them but preserving all other. This is repeated until only two vertices are left. The sum of the edges between them is the weight of the small cut. Intuitively this will preferentially generate small cuts, because small cuts are only connected by very few edges, so that the probability to contract one of those edges is low.

To keep track which vertices are contracted to determine the sets which are divided by the cut, this algorithm is supplemented by a *Union-Find* structure (also called *disjoint-set forest*) with weighting and path compression [24, 47, 48].

4.2.3. Implementation Details

First the connected components are identified by a DFS. If there are more than one connected component, one SEC for every connected component S is added, else the Stör-Wagner algorithm is used to check whether there is a global minimum cut with weight less than 2. If not, all SECs are fulfilled and all needed cutting planes are already generated, else the violated constraint is added and Karger’s algorithm is used $K = \mathcal{O}(N)$ times to find some additional cuts less than 2. The additional SECs generated this way are by no means necessary but speed up the process, because some SECs are generated in this iteration which would otherwise be generated in the next which would result in additional calls to the simplex solver. But there is also the possibility that SECs are introduced which are unnecessary, which leads to longer run times of the simplex solver. K is chosen empirically by run time measurements on some TSPLIB instances as $K = N/20$. This program flow is also depicted in Appendix C Figure C.1.

4.3. Blossom Inequalities

This section will introduce some advanced cutting planes, which are not in focus of this thesis as only Section 6.2.3 is concerned with a specific subset of the here introduced inequalities. Though because of their importance to the solution of TSP by linear programming and being the obvious way to extend this study, they are mentioned. For further details Ref. [39] is a good starting point.

Blossom inequalities (also called blossoms) [49] are inequalities, which are fulfilled by every TSP tour. If a LP relaxation violates one of them, the LP can be tightened by adding the violated inequality as a cutting plane. The blossoms are defined by Equation (4.13) given the definitions from Equations (4.8) to (4.12).

Let S_0, S_1, \dots, S_k be proper, non-empty subsets of V , with

$$k \text{ odd} \tag{4.8}$$

$$S_i \cap S_j = \emptyset \quad \forall i, j \in \{1, \dots, k\} \tag{4.9}$$

$$S_0 \cap S_i \neq \emptyset \quad \forall i \in \{1, \dots, k\} \tag{4.10}$$

$$S_i \setminus S_0 \neq \emptyset \quad \forall i \in \{1, \dots, k\} \tag{4.11}$$

$$|S_i| = 2 \quad \forall i \in \{1, \dots, k\} \tag{4.12}$$

An example of possible S_i for $k = 5$ is sketched in Figure 4.3. This sketch

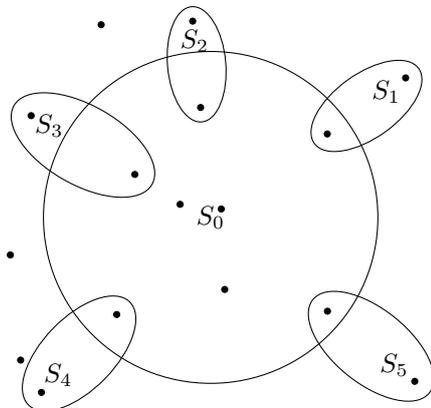


Figure 4.3: Visualization of a possible configuration for $k = 5$ of the sets S_0, S_1, \dots, S_5 . Note how the 5 sets S_1, \dots, S_5 resemble the petals and S_0 the carpels of a blossom [50].

resembles loosely a blossom with 5 leaves, which explains the name. Given the set of sets $\{S_0, S_1, \dots, S_k\}$, the blossom inequalities are defined as

$$\sum_{m=0}^k \sum_{i \in S_m, j \notin S_m} x_{ij} \geq 3k + 1, \quad (4.13)$$

i.e. one inequality for every distinct set of sets $\{S_0, S_1, \dots, S_k\}$. Sloppy formulated, the weights of the edges leaving those $k + 1$ sets are summed and have to be at least $3k+1$. Obviously the number of constraints poses an even worse problem as the SEC, and again the solution lies in polynomial-time separation algorithms [51].

A superset of the blossom inequalities are the *comb inequalities*¹² which do not have to fulfill Equation (4.12), but for which no polynomial-time separation is known yet.

4.4. Integer Programming

With the chosen formulation only integer x_{ij} are meaningful. Thus to solve the TSP, a simple LP solution, which can be obtained in polynomial time, is not sufficient. Instead an IP problem, which is known to be **NP**-hard, needs to be solved. In Figure 4.5 the example introduced in Figure 4.1 is visualized as an IP.

¹²The name originates from the interpretation of S_0 as the *handle* of the comb and S_1, \dots, S_k as the *teeth*, such that a sketch of those sets resembles a comb.

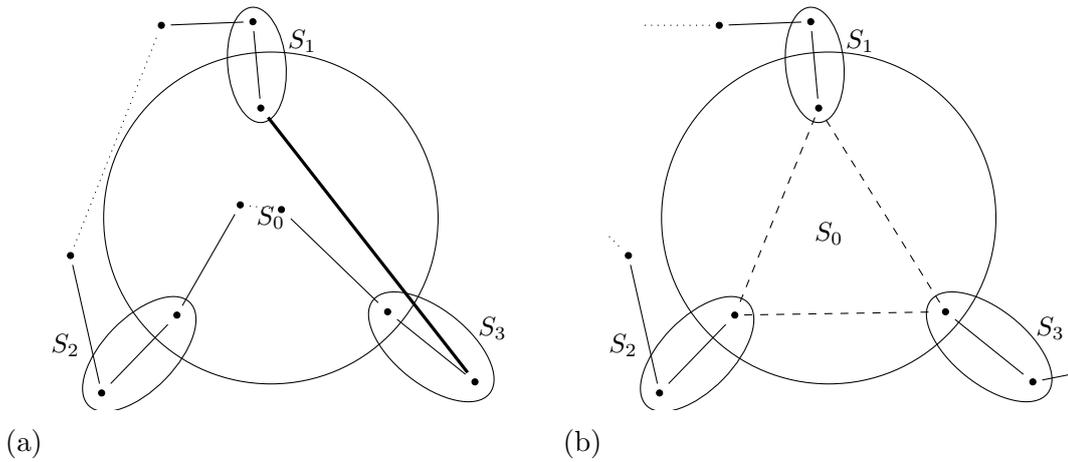


Figure 4.4: Two examples of blossom inequalities with $k = 3$. (a) shows a valid tour obeying a blossom inequality with, $\sum_{m=0}^k \sum_{i \in S_m, j \notin S_m} x_{ij} = 10 \geq 3k + 1 = 10$. Mind that the thick line is counted three times, for leaving S_0 , S_1 and S_3 . Dotted lines do not leave any set, and thus do not appear in the summation. (b) shows a part of a non-integer tour violating the shown blossom inequality. The dashed line have weight $1/2$ and therefore $\sum_{m=0}^k \sum_{i \in S_m, j \notin S_m} x_{ij} = 9 \not\geq 3k + 1 = 10$. Mind that the dashed lines are counted twice (with weight $1/2$) as they are leaving, e.g. S_1 and S_3 .

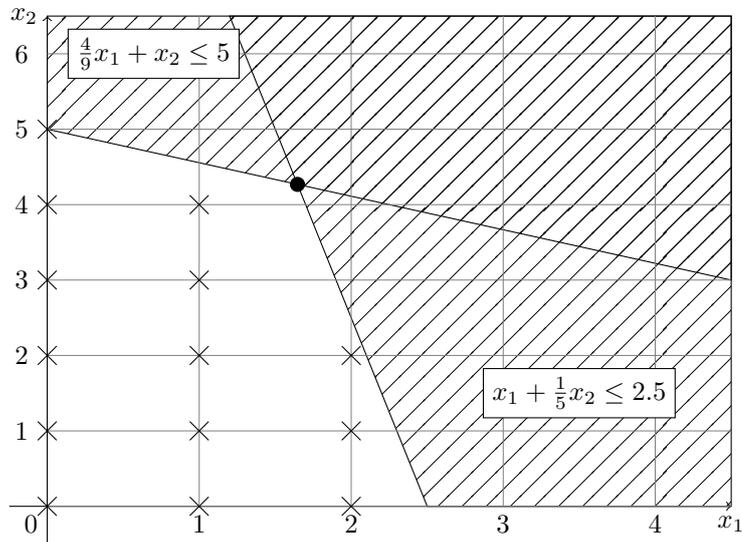


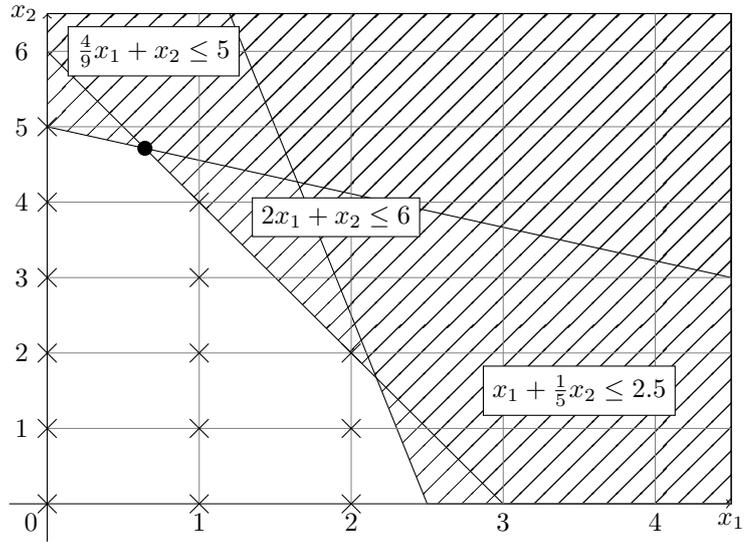
Figure 4.5: Visualization of an IP. Only solutions with integer variables marked by crosses are feasible solutions. The optimum of the LP relaxation marked by a dot is not a feasible solution.

Only the points marked with a cross are feasible integer solutions. To find the best one, we can not use a mathematical elegant way like the simplex algorithm, but have to try some, possibly exponentially many, points. The state of the art method for this purpose is called *branch-and-cut* [52].

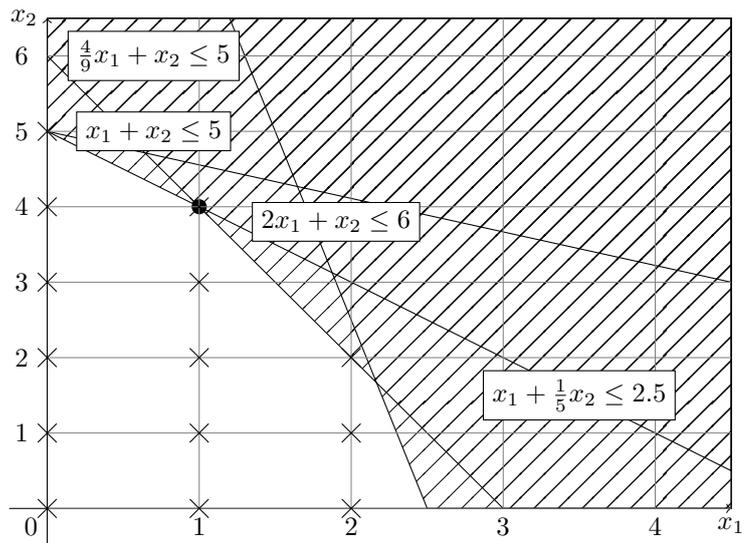
The *branch* part of the name refers to the building of a tree which is used to find the best solution by backtracking. The branching is done by adding an additional constraint which divides the solution space in two half-planes, e.g. $x_1 \leq 1$ respectively $x_1 \geq 2$ in the example Figure 4.5. For each subtree a lower bound (for maximization problems an upper bound) can be calculated, so if the current best solution is less than the lower bound of any subtree, that subtree is discarded.

The *cut* part of the name refers to cutting planes which are introduced at every node in the tree, as explained in Section 4.1. In Figure 4.6(a) a cutting plane is added to the IP from Figure 4.5 and the LP relaxation is solved again, which tightens the lower bound of that subtree. In Figure 4.6(b), after another cutting plane, the LP relaxation is integer, therefore the optimum is found. Note that those cuts are possibly valid for the whole problem and can be reused in other parts of the tree.

Because this is again not implemented by myself, I will not go beyond these basic ideas behind the algorithm. The interested reader is again referred to the literature [52, 53].



(a)



(b)

Figure 4.6: Visualization of the cutting plane method. Cutting planes shrink the solution space without forbidding feasible integer solutions by adding additional constraints iteratively such that the current LP relaxation optimum is in an infeasible region after the cut. Therefore a LP relaxation has to be solved again after each added cutting plane. In (a) one cut is added to the example from Figure 4.5 and in (b) a second cut is added such that the LP relaxation is integer and the optimal solution of the integer problem is obtained.

5. Short Introduction to Phase Transitions

A phase transition is the sudden change of an observable in a *thermodynamic system*. A thermodynamic system is a system consisting of many – which means infinitely many in the *thermodynamic limit* – elements following some (often simple) rules, which lead to emergent collective behavior of the whole system. For example a gas consists of many molecules, which have an energy and interact with each other approximately according to a Lennard-Jones potential, but shows emergent properties like density or temperature, which are undefined for single molecules. Coupling this system to a heatbath and reducing the temperature below some critical point will lead to the condensation of the gas to a liquid. At the point of this transition the *order parameter*, in this case the density, changes in a non-continuous way. This is a *first order* phase transition – the archetypal phase transition in the context of classical thermodynamics.

But in addition to first order phase transitions, there are also *continuous* or *second order* phase transitions, where the order parameter changes continuously and without latent heat. A real world example is the phase transition of ferromagnetic materials to paramagnetic above the Curie temperature T_c of that material, i.e. above T_c magnets fall off of the fridge. Note that in both examples there are in the order of $N \approx 10^{23}$ particles involved, which are enough that the results for the thermodynamic limit $N \rightarrow \infty$ are applicable without any measurable deviation.

Although the term “thermodynamic” and both examples may suggest that phase transitions are somehow connected to temperature, this is not necessarily the case. Many systems for which no temperature is defined show phase transitions. The best known example for such a model is probably *percolation* [54], which is a simple model on a lattice where sites on the lattice are occupied with probability p . At a critical p_c there exists a cluster of adjacent occupied sites spanning the whole lattice. Also many of the classical **NP**-complete problems [15–21] show phase transitions as already mentioned in the introduction.

5.1. Critical Exponents and Universality

A property shared by all second order phase transitions is that observables like the order parameter, e.g. the magnetization m for the ferromagnet example, are power-laws near the critical point, e.g.

$$m(T) \approx |T - T_c|^\beta. \quad (5.1)$$

Since power-laws can be characterized by their exponent, the critical behavior, i.e. the behavior near the phase transition, can be expressed by very few *critical exponents* [55].

One remarkable property of second order phase transitions – probably the one that makes their study so worthwhile – is that the critical exponents do not depend on details of the model, like boundary conditions or lattice structure (if the model involves a lattice), but only on a few properties, often the dimension of the lattice. This led to the classification of problems in *universality classes*. The members of a class share the same properties near the phase transition, i.e. they share the same critical exponents. One can model very simple systems but still get results that are applicable to a whole range of critical phenomena in real world materials, e.g. the three dimensional Heisenberg model can predict the critical magnetic behavior of a wealth of otherwise unrelated alloys [56].

5.2. Finite-Size Scaling

The transitions mentioned above are only sharp for the thermodynamic limit, e.g. for percolation on finite lattices there obviously exist instances without a spanning cluster even for $p > p_c$. Such deviations are called *finite-size effects* and are inherent to all computer simulations of thermodynamic systems, since unfortunately computers do only have finite memory, therefore direct simulations of the thermodynamic limit $N \rightarrow \infty$ or real-world systems $N \approx 10^{23}$ are impossible. But the properties for infinite systems can be extracted from simulations of finite systems. The prerequisite is the *scaling hypothesis* [56], which is that observables scale in vicinity of the transition point with power laws as a function of the system size N – which is well established by numerical work on second order phase transitions.

Then finite-size scaling (FSS) [56, 57], which is used throughout this thesis (and many publications), can be used to make predictions about infinite systems. In fact, this can be used to rescale measurements on different system sizes such that they are independent of the system size and thus collapse on one curve, when plotted together. This can be used to validate the estimated exponents and even to get estimates of the exponents (which are two, because both axes may have to be rescaled). This can be written as

$$O = N^\beta \tilde{O} [N^\alpha (p - p_c)] \quad (5.2)$$

with p being the parameter to tune, e.g. often temperature, and some unknown *scaling function* $\tilde{O}[\cdot]$. Hence the simulations will be done for a wide range of different system sizes N . This form is usually deduced using the correlation function

ξ between the elements in the system. For this model, however, it is not really clear – at least to the author – what the equivalent to ξ would be in this context. Therefore Equation (5.2) is used without any kind of attempt to make it plausible, besides the assurance that it works very well for problems of statistical mechanics [55–57]. Later the results will show that this leap of faith does yield some very interesting results, as this technique is often used throughout this thesis to find critical points and exponents.

However, since the power-law behavior is not an exact result, often correction terms need to be added, to describe the results with greater fidelity. These are called *corrections to scaling* and have bigger influence on small N . In this thesis no corrections to scaling are considered, instead FSS analyses are performed for system sizes large enough for the corrections to be negligible.

6. Results

This section will state the results of the computations, compare them with the literature, crosscheck with other methods and give interpretations. First the technical details needed to reproduce the results and give an idea of the quality of the data is given in Section 6.1. In Sections 6.2 to 6.4 the main result of this thesis is explained in detail, and Section 6.5 states some minor result. Finally in Section 6.6 some work is done to reproduce and extend results of older studies using the data already generated for the main part. This part comes therefore effectively without cost and does not only confirm the old results with more data, but also increases the confidence of the correctness of the main results. Some results which were generated during the study and helped the author to develop a deeper understanding, but are not – or just loosely – connected to the core of this thesis, are banished to Appendices A and B.

6.1. Technical Details

The results are mainly calculated for 11 different system sizes $N \in [32, 180]$ which are rounded powers of $\sqrt[4]{2}$ times 80 different values of $\sigma \in [0, 60]$ each time averaged over 5000 random realizations. Where exact results are not needed, the SEC LP relaxation for 10 more sizes up to $N = 1448$ cities and the degree LP relaxation up to $N = 2048$ are solved. Further, if the branch and cut tree grows beyond a memory threshold of 900 MB, the optimization is aborted and the configuration not considered. This never happened in the region $\sigma \leq 4$, but at $N = 180$ and $\sigma \approx 10$ about 20% of all configurations exceeded this memory threshold. Therefore results for exact tours at $\sigma > 4$ could deviate from the real properties, because a significant fraction of the instances were not solved to optimality. This is no problem for this analysis, because the points of interest are at $\sigma \leq 4$ as will be shown in Section 6.

All errorbars and error estimates in this thesis are either estimates obtained via bootstrap resampling [58–60] (with 200 bootstrap samples) or for fits the asymptotic standard error as given by Gnuplot¹³ normalized by χ as proposed by [59] if not noted otherwise. Symmetric errors are given as $a = 1.23(4)$ where the number in braces denotes the error of the last digit, i.e. the above value is equivalent to $a = 1.23 \pm 0.04$.

Some of the following plots will show lines, which are just linear interpolations as guides to the eye if not explained otherwise, e.g. fits are explicitly mentioned in

¹³Gnuplot is a free and open source graphing utility. <http://gnuplot.info/>

the caption and the text.

The program is written in C++ while the data is evaluated with Python scripts. The source is available upon request¹⁴. All calculations were performed on the High-End Computing Resource Oldenburg (HERO) or on a simple workstation.

CPLEX 12.6 is used as a framework to formulate and solve LP problems, i.e. its simplex and branch-and-cut implementation are used. This is a black box in the solving process, which is not optimal for the method dependent parts of the analysis. In fact CPLEX uses a bunch of generic cutting planes and heuristics which are partly considered trade secrets. Both are disabled when properties of the LP process are under scrutiny. A few optimizations are done using Concorde [39] – mainly for big instances used in images and for the analysis in Section 6.2.3. Note that Concorde needs a LP solver to find optimal tours, therefore Concorde was linked against CPLEX 12.6.

6.2. Probability that LP Relaxations are Feasible

As noted before, a phase transition from an easy circle to a hard distorted circle should be provoked. As a measure of hardness the formal complexity class of the algorithm is used in the following sense. If a LP relaxation is feasible and the separation of the constraints can be done in polynomial time, then the instance can be solved in polynomial time [44–46] and is therefore “easy”. Three different LP relaxations will be under scrutiny in this section, first the degree LP relaxation, a relaxation with only the N degree constraints Equation (4.3) and the N^2 bounds Equation (4.7) in Section 6.2.1, second the SEC LP relaxation, a formulation additionally respecting all SECs Equation (4.4). And last the fast Blossom LP relaxation which adds some Blossom inequalities Equation (4.13) as constraints to the problem. Note that this could be extended to simple comb inequalities [37, 38] which are separable in polynomial time.

As the “order parameter” to determine to which phase the σ at hand belongs to, the probability p of the system being solvable by the chosen LP relaxation is under scrutiny. For small σ , in the easy domain, it should be $p = 1$, i.e. always solvable, and in the hard domain at large σ it should be $p = 0$, i.e. never solvable. In between a transition is expected, which will occur for $N \rightarrow \infty$ at σ_c and have some critical exponent b . This p is plotted in Figures 6.1(a), 6.2(a) and 6.3(a), where it is easy to see, that the “effective transition points” of the systems with finite N are size dependent as the slope is steeper for larger system sizes. To estimate σ_c these effective transition points can be extrapolated to $N \rightarrow \infty$. Here

¹⁴hendrik.schawe@uni-oldenburg.de

the effective transition point is defined as the position of the peak of the variance of p , i.e.

$$\sigma_{\max} = \arg \max_{\sigma} \{\text{Var}(p(\sigma))\}.$$

which is plotted in Figures 6.1(c), 6.2(c) and 6.3(c). Technically the position is estimated by fitting polynomials of second order

$$f(x) = a(x - x_0)^2 + b(x - x_0) + c$$

to 5 points near the maximum of $\text{Var}(p)$ and taking its maximum's position at $\sigma_{\max} = x_0 - \frac{b}{2a}$. The errorbar of the maximum's position is estimated by bootstrap resampling, i.e. 200 sets of fit parameters are generated from bootstrap samples to estimate their errorbars [59]. Note that the value of the maximum of the variance is always $\text{Var}(p(\sigma_{\max})) = \frac{1}{4}$, which can easily be explained, because single realizations are either soluble or not, such that the single measurements $p_i \in \{0, 1\}$ follow a binomial distribution, for which the variance is defined as $\text{Var}(p) = p(1 - p)$ which reaches its maximum at $\text{Var}(p) = \frac{1}{4}$.

As mentioned in Section 5, continuous phase transitions show power-law scaling behavior, such that the power-law

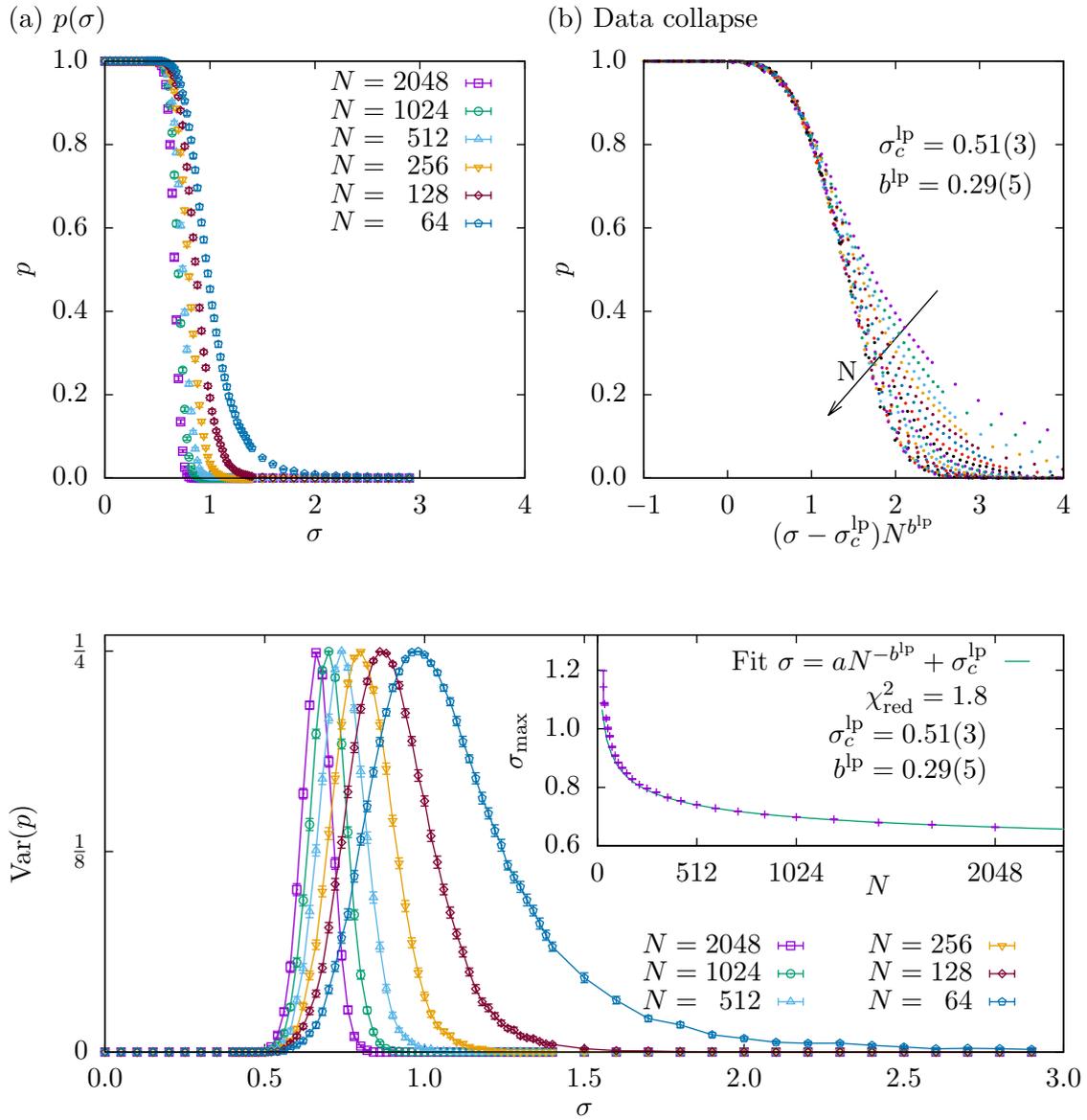
$$\sigma_{\max} = aN^{-b} + \sigma_c \tag{6.1}$$

is fitted to the obtained maxima as shown in the insets of Figures 6.1(c), 6.2(c) and 6.3(c).

As a cross check, to validate the values of the fit, a data collapse is performed, utilizing the property from Section 5.2, such that one can rescale the σ -axis as $(\sigma - \sigma_c)N^b$ to let all measurements collapse on one curve – at least in vicinity of the critical point, i.e. near 0 on the rescaled axis. The p -axis does not need to be rescaled, since p is a dimensionless quantity. Those collapses are displayed in Figures 6.1(b), 6.2(b) and 6.3(b). They omit errorbars for better readability and look very acceptable. The collapse works best for large values of N , where corrections to scaling have less influence. This can be observed in every collapse. Since the extrapolation does not consider errors of the single points, the errors given, are just a rough estimate, which seem to be overestimated, because the collapse does yield drastically worse results for some values still within the errorbars.

6.2.1. Degree LP Relaxation

The degree LP relaxation is the weakest studied relaxation, in fact for a solution of this relaxation to be feasible it is not sufficient to be integer, since subtours can



(c) $\text{Var}(p)$, inset: fit to $\sigma_{\text{max}} = aN^{-b^{\text{lp}}} + \sigma_c^{\text{lp}}$ for $N \geq 512$

Figure 6.1: (a) Probability p that the degree LP relaxation is feasible, i.e. the x_{ij} are integer and there are no subtours present. (b) The same p plotted on a rescaled axis such that the measurements collapse on one curve, with σ_c^{lp} and b^{lp} obtained from (c). (c) Extrapolation of the transition point by a power-law through the peak positions of $\text{Var}(p)$ yielding $\sigma_c^{\text{lp}} = 0.51(4)$ and $b^{\text{lp}} = 0.29(6)$.

still occur. Therefore once an integer solution is found, a DFS is performed to ensure that the graph representing the tour is connected. If it is, the solution is feasible. Note that this additional DFS is crucial for this kind of relaxation. In fact, attempts to collapse the probability p of all integer solutions, regardless of them containing subtours, fail.

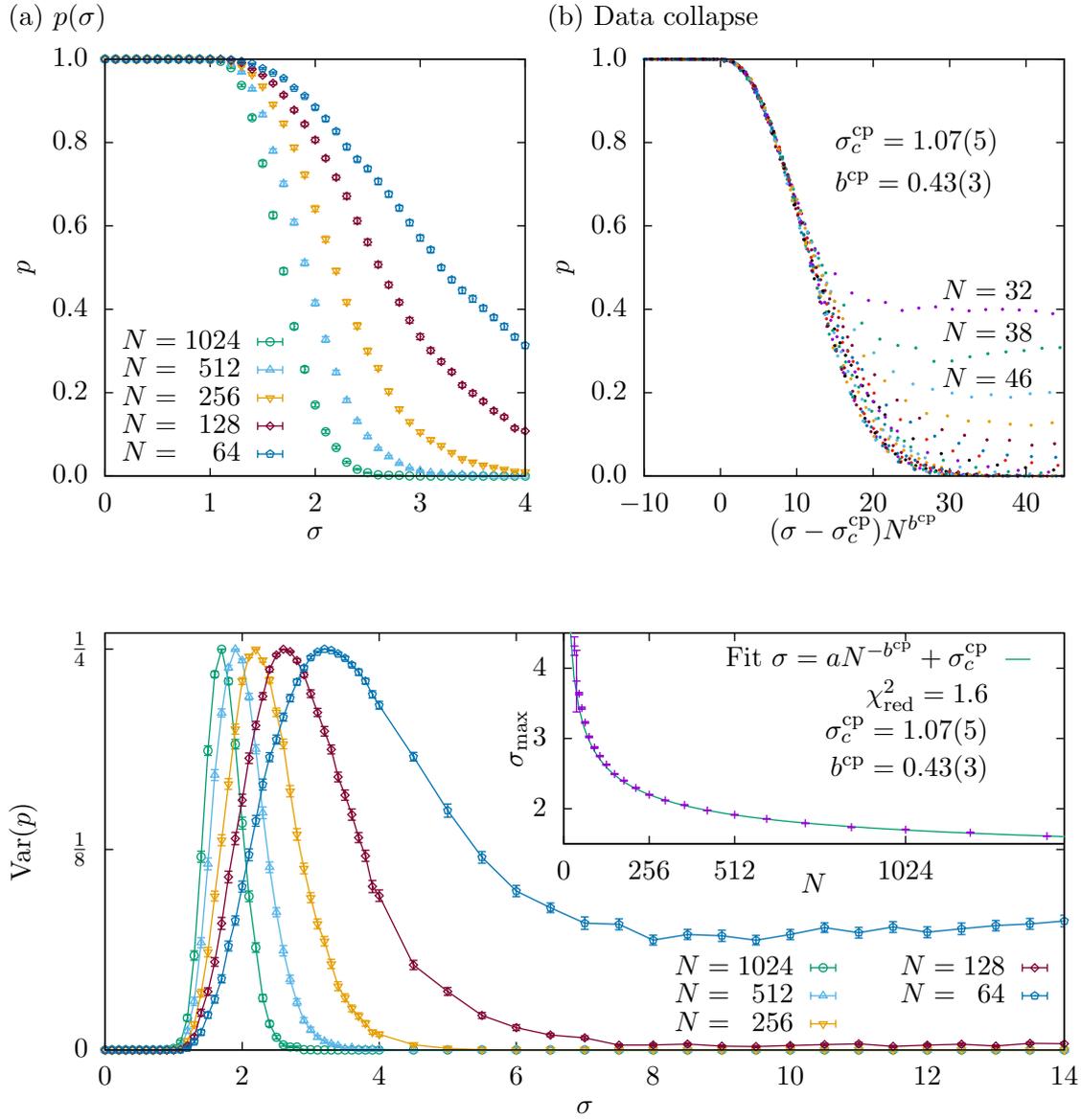
6.2.2. SEC LP Relaxation

The SEC LP relaxation is the most natural relaxation of the TSP IP, since it just drops the integer constraints. As evident from Figure 6.2(a) its solution probability p is always greater than the one for the degree LP relaxation in Figure 6.1(a), which is easily understandable because it uses a superset of constraints. The data collapse in Figure 6.2(b) shows major corrections to scaling for the small systems. This can be made plausible since for small systems there is no hard region, where no realization can be solved, i.e. for $N \leq 6$ every realization is solvable by the SEC LP relaxation independent of σ and even for $N = 64$ there are always some instances that can be solved, as evident from the non-zero variance in Figure 6.2(c).

6.2.3. Fast Blossom LP Relaxation

To get an even tighter LP formulation, some blossom inequalities (compare Section 4.3) are added as cutting planes. The separation is done using heuristics, which do not find all violated blossom inequalities, though polynomial-time algorithms exist for this purpose. Here the Concorde TSP solver is used for the separation of *fast blossom inequalities*, and the analysis whether the solution of this relaxation is a feasible TSP tour or not. Here arise some small deviations from the analyses done before, because Concorde rounds the distances to integer values, which is mitigated by scaling the whole system by a sufficiently large factor to reach a comparable precision as with the floating point distances in all other calculations. Further the class of fast blossoms is not well defined, i.e. the set of found fast blossoms is dependent on internal choices for the heuristics – in fact Concorde does this dependent on a seed with a pseudo-random number generator. But since this study just looks at stochastic averages, this is not a serious problem. The heuristics themselves are well defined and the reader is referred to [39]. I will call this relaxation by a non-canonical name *fast Blossom LP relaxation*.

As expected, the solution probability p is always higher than for the SEC LP relaxation and therefore the critical point moves. But the critical exponent seems to stay the same.



(c) $\text{Var}(p)$, inset: fit to $\sigma_{\text{max}} = aN^{-b^{\text{cp}}} + \sigma_c^{\text{cp}}$ for $N \geq 256$

Figure 6.2: (a) Probability p that the SEC LP relaxation is feasible, i.e. x_{ij} are integer. (b) The same p plotted on a rescaled axis such that the measurements collapse on one curve, with σ_c^{cp} and b^{cp} obtained from (c). (c) Extrapolation of the transition point by a power-law through the peak positions of $\text{Var}(p)$ yielding $\sigma_c^{\text{cp}} = 1.07(5)$ and $b^{\text{cp}} = 0.43(3)$.

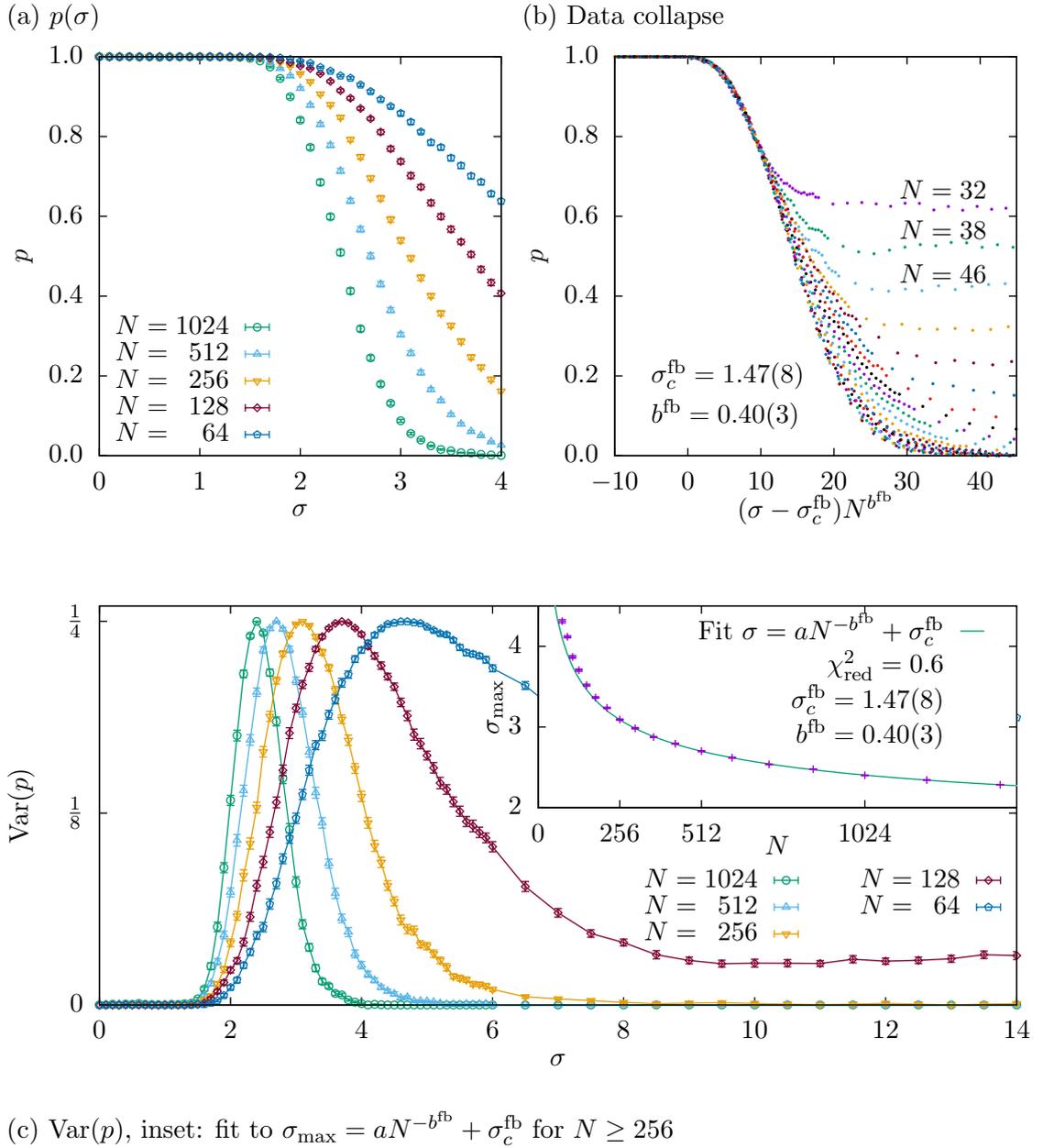


Figure 6.3: (a) Probability p that the fast Blossom LP relaxation is feasible, i.e. x_{ij} are integer. (b) The same p plotted on a rescaled axis such that the measurements collapse on one curve, with σ_c^{fb} and b^{fb} obtained from (c). (c) Extrapolation of the transition point by a power-law through the peak positions of $\text{Var}(p)$ yielding $\sigma_c^{\text{fb}} = 1.47(8)$ and $b^{\text{fb}} = 0.40(3)$.

6.3. Structural Properties

While Section 6.2 is about the solution method, this section is about actual optimal solutions. If the solution to the SEC LP relaxation is not feasible, i.e. not integer, the optimal tour is found by a branch-and-cut search as described in Section 4.4. The aim is to look at the properties of the optimal tours, which are in no way related to the solution process and find observables which hint at the phase transitions found in Section 6.2, which are extracted from observations of the solution process. In other words, seek correlation between hardness and physical properties.

This succeeds for the degree LP relaxation (see Section 6.3.1) and SEC LP relaxation (see Section 6.3.2) but no physical property exhibiting a transition coinciding with the fast Blossom LP relaxation transition was found. Anyway, since the fast Blossom LP relaxation is heavily based on heuristics, it is not a huge setback. For further studies a look at a relaxation implementing all blossom inequalities or, say, simple comb inequalities, would be promising.

6.3.1. Difference to the Circle

Comparing the optimal solution tour x_{ij}^* to the optimal tour at $\sigma = 0$, i.e. the tour along the circle x_{ij}^o , it is expected that they are similar at small σ . A way to

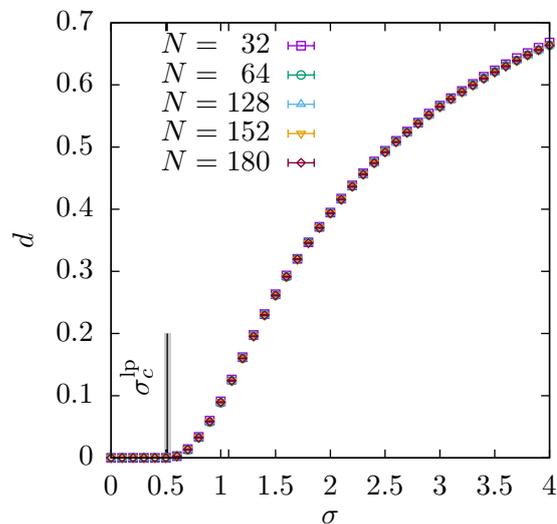


Figure 6.4: Difference of the optimal tour to the initial circle d as a function of σ . The difference d is defined as a suitable normalized Hamming distance (see text).

measure this similarity is to look at the number of edges occurring in the one tour but not in the other, i.e. the Hamming distance [61]. The tour difference d shown in Fig. 6.4 is the Hamming distance normalized by $2N$, such that two tours with no common edges would result in $d = 1$ while two tours visiting the cities in the same sequence would result in $d = 0$. This observable seems to be independent of N . Fig. 6.4 suggests that realizations are solvable by the degree LP relaxation alone as long as $d = 0$, i.e. the solution is the same sequence as the circle $x_{ij}^* = x_{ij}^\circ$.

6.3.2. Tortuosity

Pictures of optimized tours as shown in Figure 3.3 show more meandering courses with increasing σ . As a measure of this “entanglement”, the *tortuosity* τ is introduced in Ref. [62]. It was originally created to evaluate images of blood vessels in the retina to detect vascular diseases. To calculate τ , the tour is segmented into n segments of same-sign curvature. Let the *arc length* L_i be the length of the segment i along the tour and let the *chord length* S_i be the distance between the first and last city of the segment i . L is the total length of the tour. Then the tortuosity is defined as

$$\tau = \frac{n-1}{L} \sum_{i=1}^n \left(\frac{L_i}{S_i} - 1 \right). \quad (6.2)$$

That means that for a high tortuosity τ , detour-like structures, which connect two points close to each other with a long bow leading to big $\frac{L_i}{S_i} - 1$ values, and many of those, to increase the factor $n - 1$, are needed.

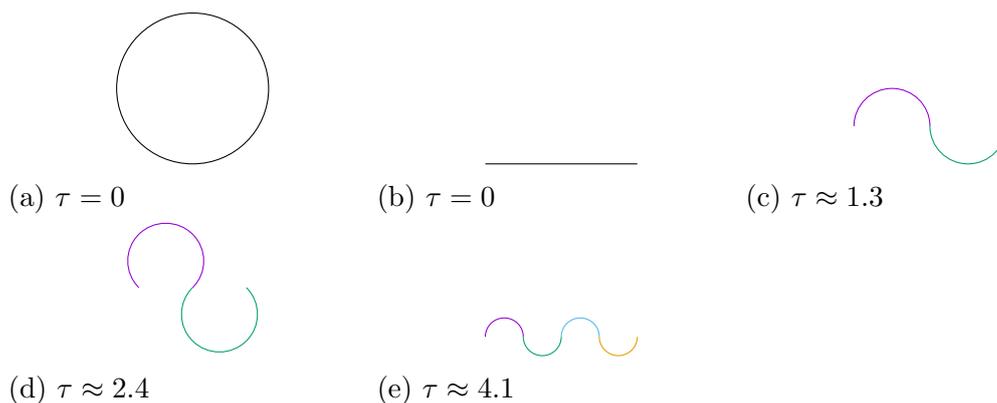


Figure 6.5: Examples of the tortuosity of different curves. The length of (b) is 2, all other curves are to scale. The segments of same sign curvature are indicated by different colors. (c) consists of two half circles, (d) of two 3/4 circles and (e) of four half circles.

The straight line and the circle have both a tortuosity of $\tau = 0$ while the shape of a meandering river will yield a high τ . Also many small meanders will typically yield a higher τ than few big meanders, because of the prefactor $n - 1$. Examples for those cases are depicted in Figure 6.5.

Although this is originally a measure for continuous curves, it is easy to interpret it for piece-wise linear curves. The segments are created by walking along the

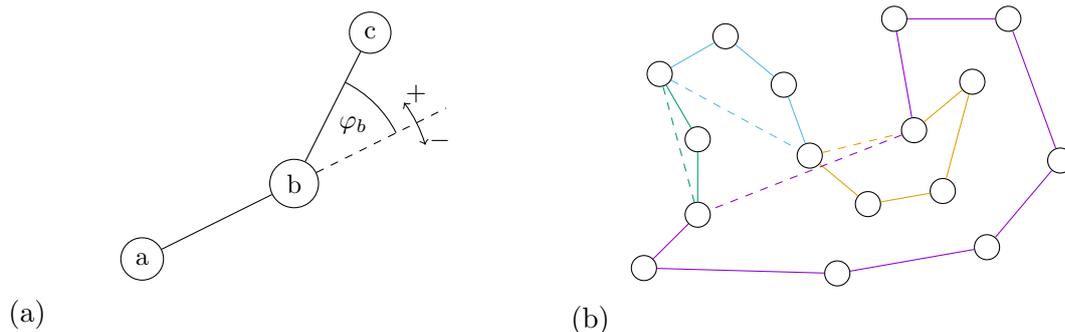


Figure 6.6: (a) Definition of the bearing φ . (b) An example tour. The $n = 4$ (clockwise) segments are marked with different colors. For the i -th segment the arc length L_i is the length along the solid lines, the chord length S_i along the dashed line.

tour and each time the sign of the bearing φ defined in Figure 6.6(a) changes, a new segment is started. But mind that the segments depend on the orientation of the walk, i.e. whether it is clockwise or counterclockwise. This, however, should not influence the mean value over many realizations. Note that there are other possibilities to define the tortuosity for piece-wise linear curves, e.g. first converting it to a continuous curve with spline interpolation. In Figure 6.6 an example of a (non-optimal) tour consisting of $N = 16$ cities and $n = 4$ segments is pictured.

Because there are single instances, where two cities are so close to each other that the tortuosity is orders of magnitude larger for those realizations than the mean, the two maximal measurements, corresponding to 0.04% of all, are discarded for the calculation of the mean. Alternatively the median could be used, which does look qualitatively similar, but seems yield slightly worse results in the following analysis.

In Figure 6.7 the tortuosity peaks at roughly σ_c^{cp} . The positions of the peaks are estimated by fits of straight lines left and right of the peak, as indicated by black lines in Figure 6.7 for the example of $N = 180$. Their intersection is used as the estimate for the peak position and extrapolated to $N \rightarrow \infty$ using a power-law

$$\tau_{\text{max}} = aN^{b\tau} + \sigma_c^\tau$$

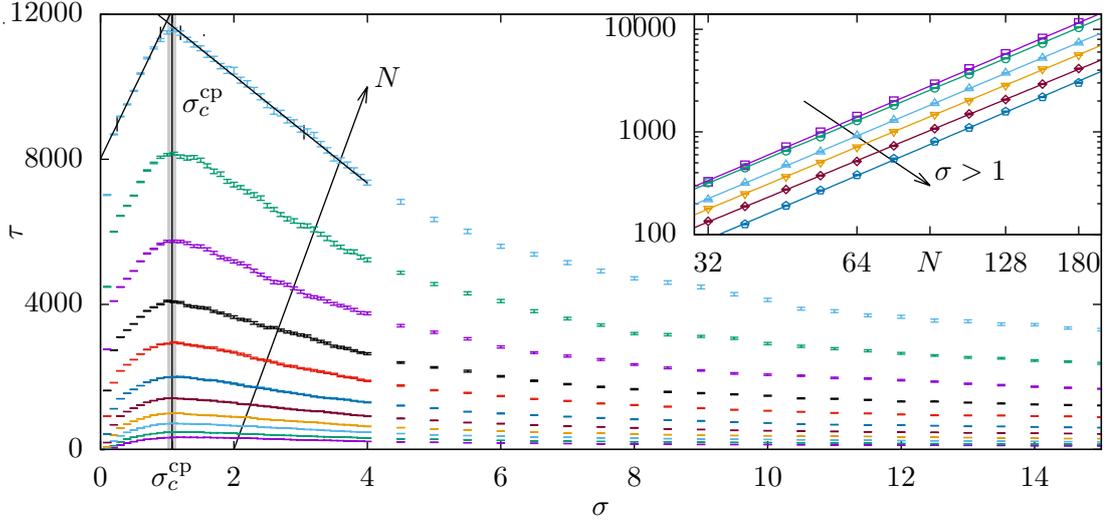


Figure 6.7: Tortuosity for different values of $N \leq 180$ and σ . At $\sigma_c^\tau = 1.06(23)$ the tortuosity peaks. For $\sigma \gtrsim 1$ the tortuosity scales according to the power-law $\tau \propto N^{2.02(2)}$ with the system size as shown for $\sigma \in \{1, 2, 4, 6, 10, 18\}$ in the inset.

similar to Section 6.2. This yields $\sigma_c^\tau = 1.06(23)$, which confirms that the tortuosity indicates the phase transition found before. Mind that the large uncertainty is caused by the very rough method to estimate the peak position. Unfortunately the exponent is not as robust and b^τ obtained by this fit has a very large uncertainty $b^\tau = 0.67 \pm 1.59$, such that it can not be estimated with this method.

6.4. Is there Universality?

A property of phase transitions of second order is the fact that they can be categorized in different universality classes which are explained in more detail in Section 5.1. To test whether the TSP on the FCE establishes an universality class, one can change some details and see if the critical exponents stay the same. Therefore in Section 6.4.1 the distribution from which the displacement of the cities is drawn is changed and in Section 6.4.2 the dimension in which the cities are distributed is changed. The analysis is the same as in Section 6.2. A comparison of all critical points and exponents is also given as a conclusion in Table 6.1.

6.4.1. Gaussian Displacement

If the displacement is not drawn from the uniform distribution $r \in [0, \sigma]$ but instead the displacement in x and y direction is drawn from a Gaussian distribution $\Delta x \in G(0, \sigma)$ with zero mean and width σ , the resulting city positions are surely different. For example arbitrary long displacements have a non-zero probability. The probability that the SEC LP relaxation yields an integer solution is shown in Figure 6.8 in the same way as before in Figure 6.3. The extrapolation in Figure 6.8(b) yields a different critical point $\sigma_c^{\text{cp,g}}$, but a critical exponent $b^{\text{cp,g}} = 0.45(5)$ which is in good agreement with the one from the SEC LP relaxation $b^{\text{cp}} = 0.43(3)$ thus suggesting that universality is indeed given.

Further, the tortuosity τ is calculated in the same way and for the same system sizes N as in Section 6.3.2 and plotted in Figure 6.9, where one can see that it peaks at $\sigma_c^{\tau,\text{g}} = 0.44(8)$ which is compatible with the critical point $\sigma_c^{\text{cp,g}} = 0.47(3)$. This simple test should mainly give confidence that τ is somehow connected to the critical point of the SEC LP relaxation.

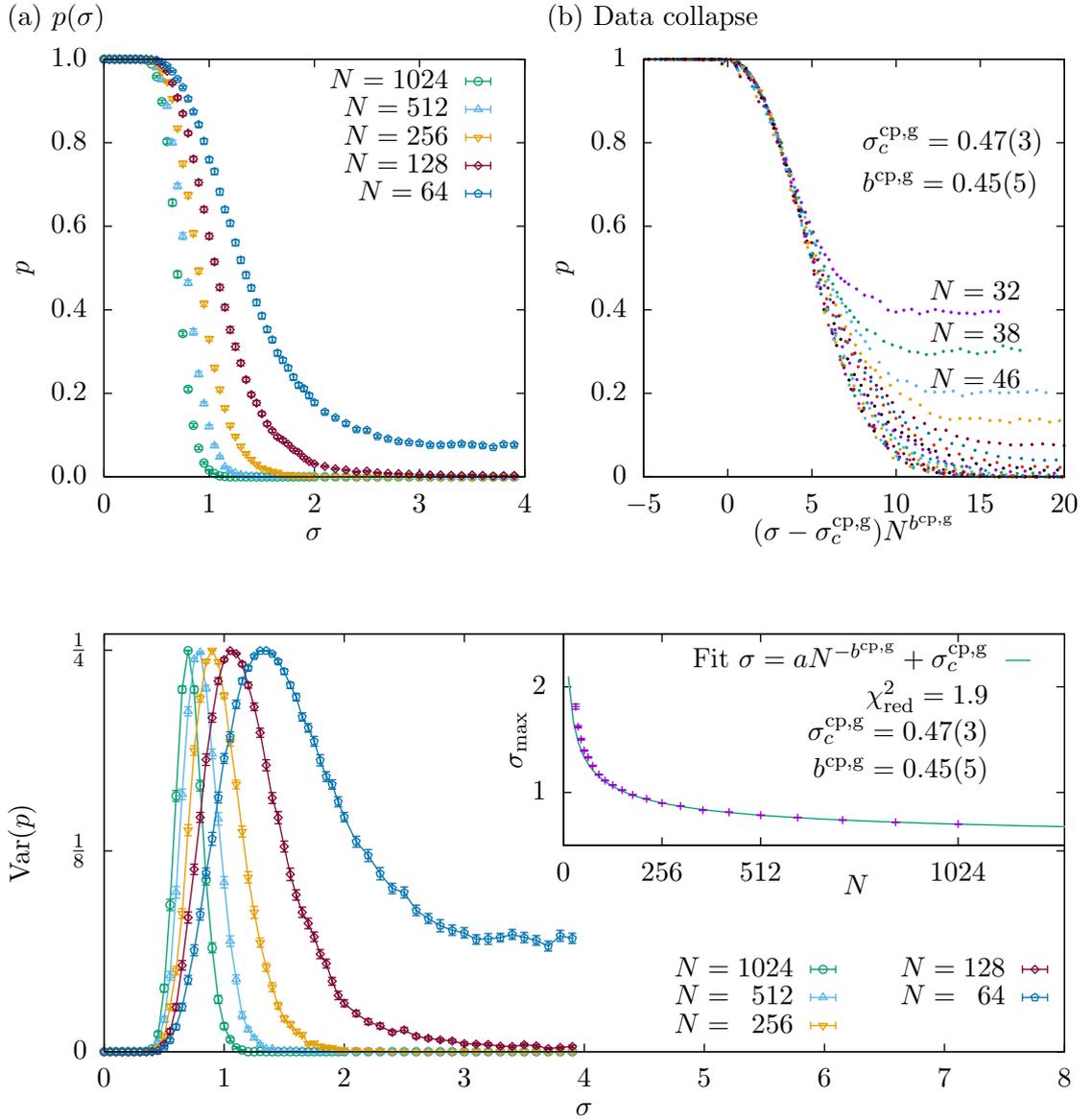
6.4.2. Spherical Displacement

Often the universality class changes if the dimension of the model is changed. Though one can define the Euclidean TSP in higher dimensions, the starting configuration of the FCE can not be generalized easily to higher dimensions. For example, cities on a sphere instead of a circle are not easy to solve realizations anymore, which is the rationale for the circle. Anyway, starting from a circle a displacement of the cities inside a sphere of radius σ is tested next.

In Figure 6.10 the results of simulations for a narrow range of $\sigma \leq 4$ are shown. This range is sufficient to evaluate the peak-positions for $\sigma \geq 256$ and get the exponent $b^{\text{cp,3}} = 0.40(4)$, but too short to extend the data collapse Figure 6.10(b) to large rescaled values to demonstrate the deviations there from the FSS as in the collapses before.

6.4.3. Comparison to the Vertex Cover Transition

In Ref. [22] the same analysis is conducted, but for Vertex Cover optimization on Erdős-Rényi graphs, where the tuning parameter analogous to the σ in this thesis is the *connectivity* c . There a LP approach with heuristic cutting planes led to a transition from “solvable by LP” to “not solvable by LP” at $c = 2.62(17) \approx e$



(c) $\text{Var}(p)$, inset: fit to $\sigma_{\text{max}} = aN^{-b^{\text{cp,g}}} + \sigma_c^{\text{cp,g}}$ for $N \geq 256$

Figure 6.8: (a) Probability p that the SEC LP relaxation with the Gaussian displacement (see text) is feasible, i.e. x_{ij} are integer. (b) The same p plotted on a rescaled axis such that the measurements collapse on one curve, with $\sigma_c^{\text{cp,g}}$ and $b^{\text{cp,g}}$ obtained from (c). (c) Extrapolation of the transition point by a power-law through the peak positions of $\text{Var}(p)$ yielding $\sigma_c^{\text{cp,g}} = 0.47(3)$ and $b^{\text{cp,g}} = 0.45(5)$.

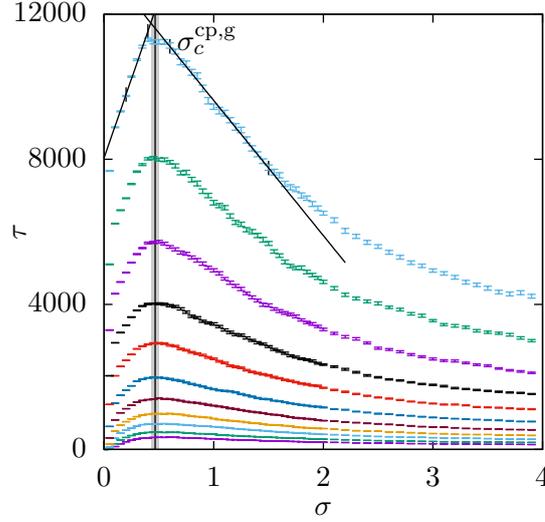
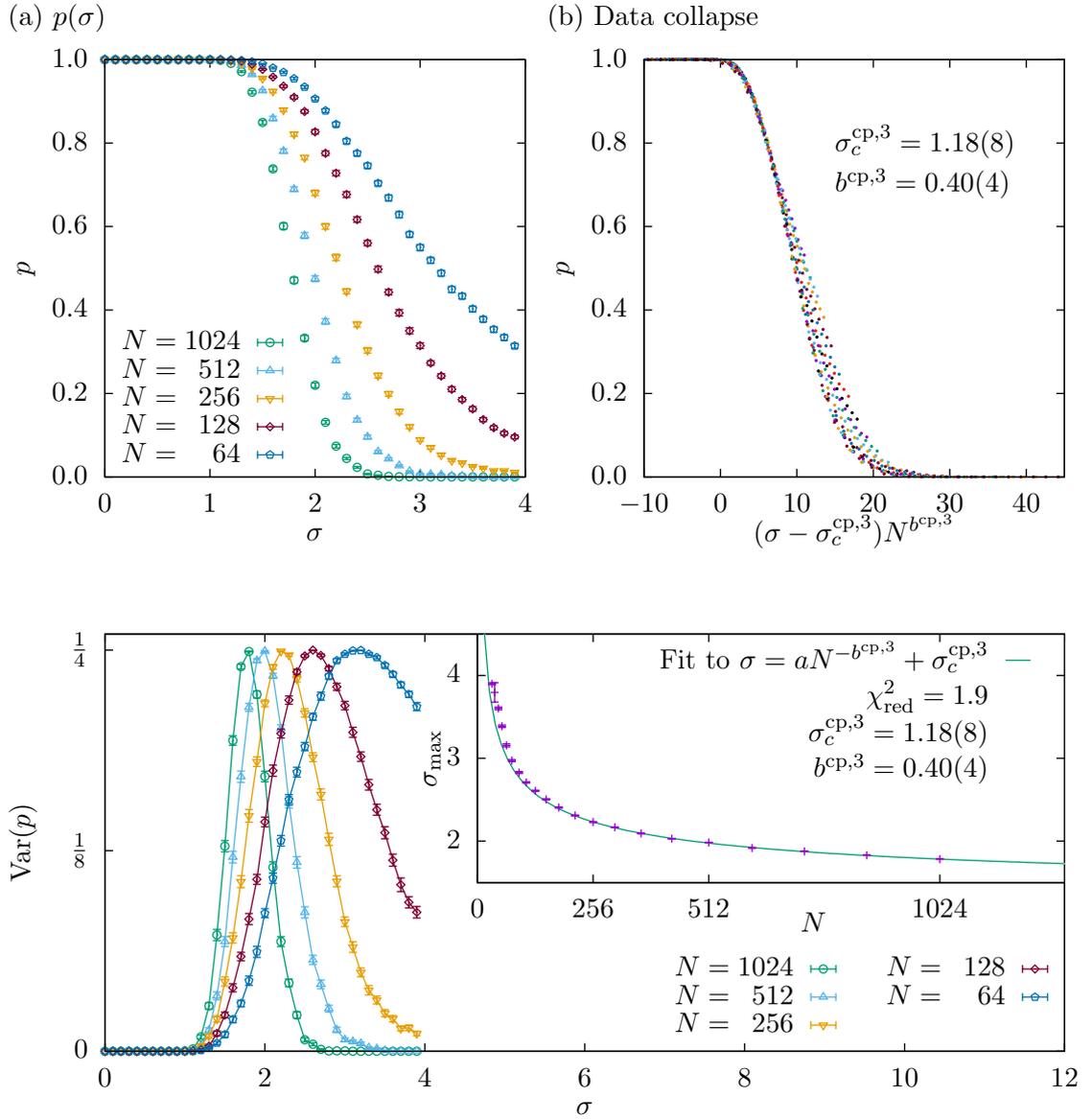


Figure 6.9: Tortuosity for different numbers of cities $N \leq 180$ and σ . At $\sigma_c^{\tau, \text{g}} = 0.44(8)$ the tortuosity peaks.

where replica symmetry breaks [18]. Though the critical exponent $b = 0.55(5)$ [63] suggests, that these transitions do not belong to the same universality class.

6.5. First Excitation: The Second Shortest Tour

Glassy systems such as spin glasses or actual glasses (from which windows are made) show a very rough and complicated *energy landscape*, which makes it especially hard to find the ground state, i.e. the configuration with minimal energy. Interpreting the tour length L as an energy, the energy landscape of the TSP does probably exhibit similar properties. To examine the energy landscape, excitations of the ground state are created and their “distance” in configuration space to the ground state is tracked. If two tours with a small energy/length difference are “far away” in configuration space, this is a hint for a very rough and complicated energy landscape with many local minima in which local searches can be trapped. As a measure of distance in the configuration space, the tour difference d as introduced in Section 6.3.1 is used again. Note that this measure is resilient against reversal of slices of the tour. Since the used solution method enables exact solutions, instead of some random excitation, the first excitation, i.e. the second shortest tour, can be obtained. For every of the N edges $\{i, j\}$ of the optimal tour, one edge at a time is fixed to $x_{ij} = 0$ by adding this equality as a further constraint. Of the N solu-



(c) $\text{Var}(p)$, inset: fit to $\sigma_{\max} = aN^{-b^{\text{cp},3}} + \sigma_c^{\text{cp},3}$ for $N \geq 256$

Figure 6.10: (a) Probability p that the SEC LP relaxation with the three-dimensional displacement (see text) is feasible, i.e. x_{ij} are integer. (b) The same p plotted on a rescaled axis such that the measurements collapse on one curve, with $\sigma_c^{\text{cp},3}$ and $b^{\text{cp},3}$ obtained from (c). (c) Extrapolation of the transition point by a power-law through the peak positions of $\text{Var}(p)$ yielding $\sigma_c^{\text{cp},3} = 1.18(8)$ and $b^{\text{cp},3} = 0.40(4)$.

	σ_c	b
degree LP relaxation	$\sigma_c^{\text{lp}} = 0.51(4)$	$b^{\text{lp}} = 0.29(6)$
SEC LP relaxation	$\sigma_c^{\text{cp}} = 1.07(5)$	$b^{\text{cp}} = 0.43(3)$
	$\sigma_c^{\tau} = 1.06(23)$	–
	$\sigma_c^{\text{cp,g}} = 0.47(3)$	$b^{\text{cp,g}} = 0.45(5)$
	$\sigma_c^{\tau,\text{g}} = 0.44(8)$	–
	$\sigma_c^{\text{cp,3}} = 1.18(8)$	$b^{\text{cp,3}} = 0.40(4)$
fast Blossom LP relaxation	$\sigma_c^{\text{fb}} = 1.47(8)$	$b^{\text{fb}} = 0.40(3)$

Table 6.1: Values of critical points and exponents grouped by different types of relaxations.

tions¹⁵, the solution with minimum length is the second shortest tour. Obviously this method is very expensive, because N IP problems have to be solved.

The analysis is performed on configurations where the cities are uniformly, randomly distributed inside the unit hypercube, e.g. the unit square in two dimensions, instead of the FCE. Since this ensemble is the most well researched one, it is reasonable to use it instead of the FCE. This also enables to look at proper defined higher dimensional realizations.

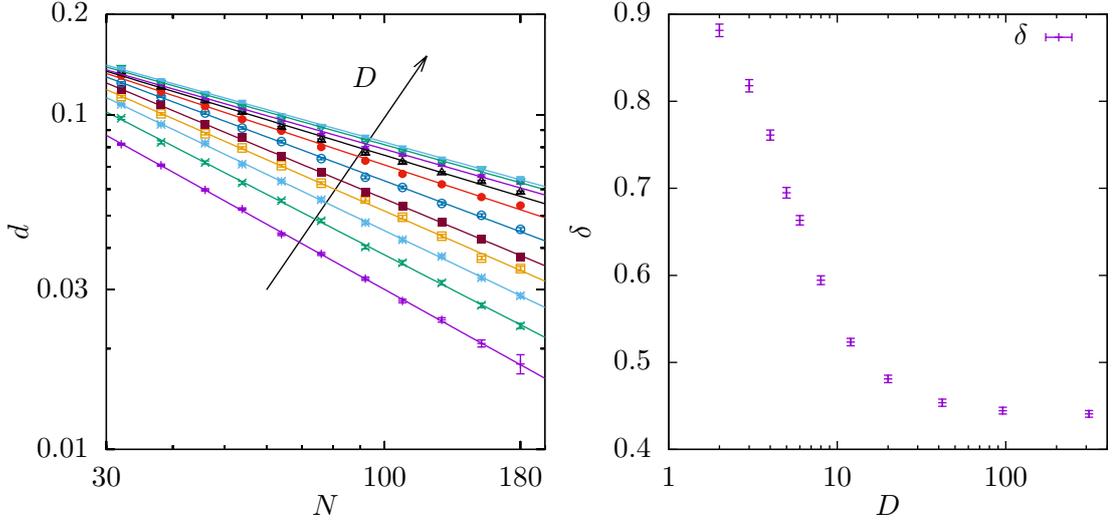
First, the configurations in high dimensions give more freedom to the second shortest tour, which is evident from the constantly higher d , i.e. more dissimilar first excitation. Surprisingly the difference to the optimal tour d seems to follow the power-law $d = aN^{-\delta}$ with positive δ for all tested dimensions $D \in \{2, 3, 4, 5, 6, 8, 12, 20, 42, 96, 312\}$. Also the exponent δ seems to approach a limit of

$$\lim_{D \rightarrow \infty} \delta \approx 0.4$$

as shown in Figure 6.11(b). This means that for infinitely large systems the fraction of different edges between the shortest and second shortest tour likely approaches $d = 0$ and confirms the assumption of many local search heuristics for the TSP – that good tours are similar.

A similar analysis of low excitations – but not first excitations – is conducted in Ref. [64] for the random-field Ising model (RFIM). There the *overlap* q of the ground state and an excited state are measured. The overlap q is an observable very similar to the tour difference d , as it basically counts the number of flipped

¹⁵Note that subsequent solution processes do not need to start from scratch, because all cutting planes stay valid and most are still relevant.



(a) Tour difference d with power-law fits (b) Exponents of the power-laws of (a)

Figure 6.11: For $N = 180, D \in \{2, 3\}$ and $N = 152, D = 2$ there are less samples taken, therefore the errorbars are bigger. (a) shows a log-log plot of the tour difference d between the shortest and second shortest tour for different system sizes N and different dimensions D in which the cities are distributed. For infinite systems the difference d approaches 0. (b) The exponent δ seems to reach a limit for big D of $\delta \approx 0.4$. The x -axis is logarithmically scaled for better readability.

spins between two systems, $q = 1$ being identical spin configurations, $q = 0$ being half of the spins flipped and $q = -1$ the inverse spin configuration, i.e. all spins flipped. $q = 1$ is therefore analogue to $d = 0$ and $q = 0$ is analogue to $d = 1$. But in contrast to the result of this section where d goes to zero for $N \rightarrow \infty$, Ref. [64] shows that the distributions of the overlap and therefore the mean q shift to $q < 1$ for large systems (though the distribution get narrower with larger system sizes which hints that the RFIM is not glassy). This is a hint that – at least the Euclidean TSP – is not a glassy system.

Further study could be carried out for other random ensembles that are not based on the Euclidean metric and possibly not even constrained by triangle inequalities, as even the high-dimensional cases might not give enough freedom in that regard. Another field for further study could be a look at the FCE and first excitation properties as a function of sigma, since the length difference of the best and second best tour would be about 1 for $\sigma = 0$ and shrink for larger sigma. Possibly the phase transitions from Section 6.2 could be observed this way, too.

6.6. The TSP Decision Problem

As noted in Section 2 **NP**-hard problems can be reduced to every problem in **NP**. Especially the solutions of the TSP can be trivially reduced to solutions of the corresponding decision problem. Since the TSP gives the optimal length L^* of a tour, the answer to the question, if there is a tour shorter than some threshold L_t , can be answered after a single comparison of the optimal length and the threshold. Thus the results from Ref. [21] can be reproduced easily. There the probability $p_<$ that a solution shorter than a given length L_t exists was plotted over a control parameter $k = \frac{L}{\sqrt{NA}}$ and rescaled until the curves of different system sizes N collapse on one curve. Ref. [21] examines configurations created by a Poisson process in the two dimensional unit square, i.e. with area $A = 1$.

6.6.1. Uniformly Distributed Cities in High Dimensions

In Ref. [21] the TSP decision problem was just solved in two dimensions for $N \in \{6, 12, 18, 24, 30\}$ with a branch-and-bound algorithm. In this thesis not only a far more efficient method is used but there is also far more computing power available than in 1996 when it was published, therefore the same analysis is performed for $N \in [32, 180]$ cities which are uniformly distributed in $D \in \{2, 3, 4, 12, 42, 312\}$ dimensions, which are already optimized for Section 6.5. In [65–68] the control parameter k is given for arbitrary dimensions D as

$$k = \frac{L}{(NA)^{1-1/D}}.$$

Example plots of the cumulative distribution function (CDF) of the tour length L and the control parameter k are given in Figure 6.12(a) respectively Figure 6.12(b). The probabilities $p_<$ are estimated by a 20 bin histogram that is integrated using a simple sum. In Figure 6.13 the critical k_c and the scaling exponent ν are determined by a FSS analysis comparable. The collapse is produced in an semi-automated way by `autoscale.py` [69], which also offers an estimate of the quality of the collapse S and an estimate of the error of the obtained values – k_c and ν in this case – further references are given in Ref. [69]. The lower the value of S , the better the quality of the collapse, though because of statistical fluctuations a value of $S \approx 1$ is considered as a good value. Because small instances are more susceptible to deviations from the scaling behavior, only instances of $N \geq 92$ were considered to determine the collapse and smaller instances are omitted in the plots. As noted above, the scaling assumptions are only valid near the critical point, i.e. near zero on the rescaled abscissa, therefore only data points near zero in the range of $[-0.5, 1.5]$ on the rescaled axis are used to determine the collapse.

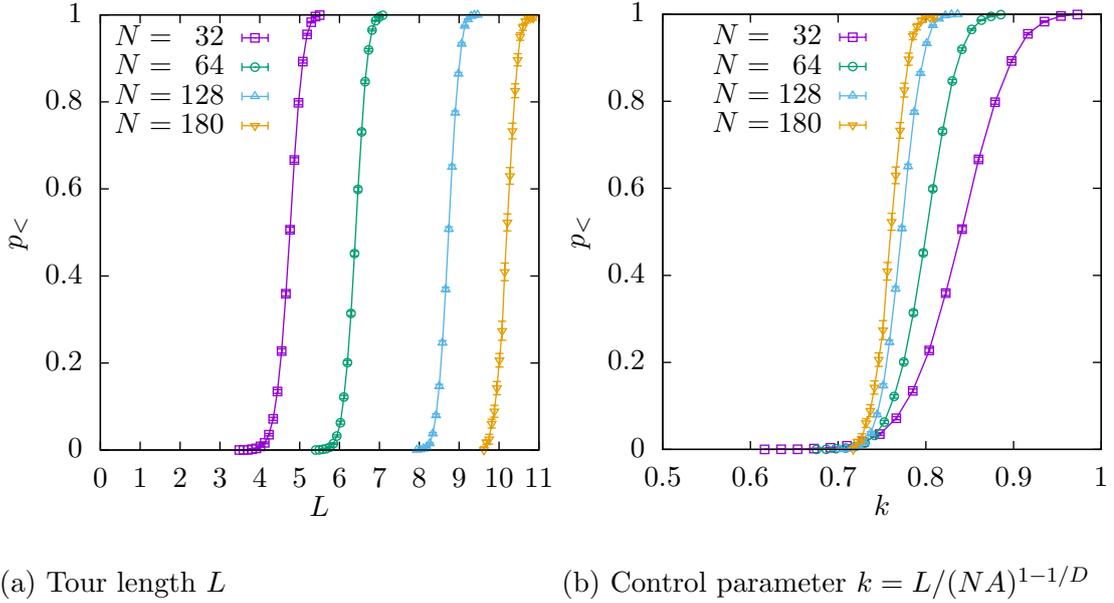


Figure 6.12: CDF of tour length L and control parameter k plotted for two dimensional uniformly distributed cities on the unit square. If more cities have to be visited, the tour gets longer. The control parameter k , compensates this effect and shows an intersection point of all system sizes N at a critical k_c .

The value of $k_c = 0.721(4)$ for $D = 2$ is lower than in Ref. [21] which could be caused by corrections-to-scaling-effects of the very small system sizes in used there. In fact a value of $k_c = 0.7124(2)$ is expected, which is the $N \rightarrow \infty$ asymptotic optimal tour-length parameter k for the Euclidean TSP [66]. A far worse deviation is visible for the obtained values for three- and four-dimensional configurations. Though Ref. [66] states that k_c should increase (though not strictly) with the dimension, the values observed in Figure 6.13 are not in good agreement with the expected ones $k_c^3 = 0.6980(3)$ and $k_c^4 = 0.7234(3)$. This could be caused by more severe finite-size effects as for the quite small sizes used $N \leq 180$ the fraction of cities affected by the open boundary conditions is quite high. Therefore the deviation from the values for $N \rightarrow \infty$ are plausible.

But the values for ν are in very good agreement with the values from Ref. [21] and are within errorbars the same for all D which hints that this model is in the same universality class for all dimensions.

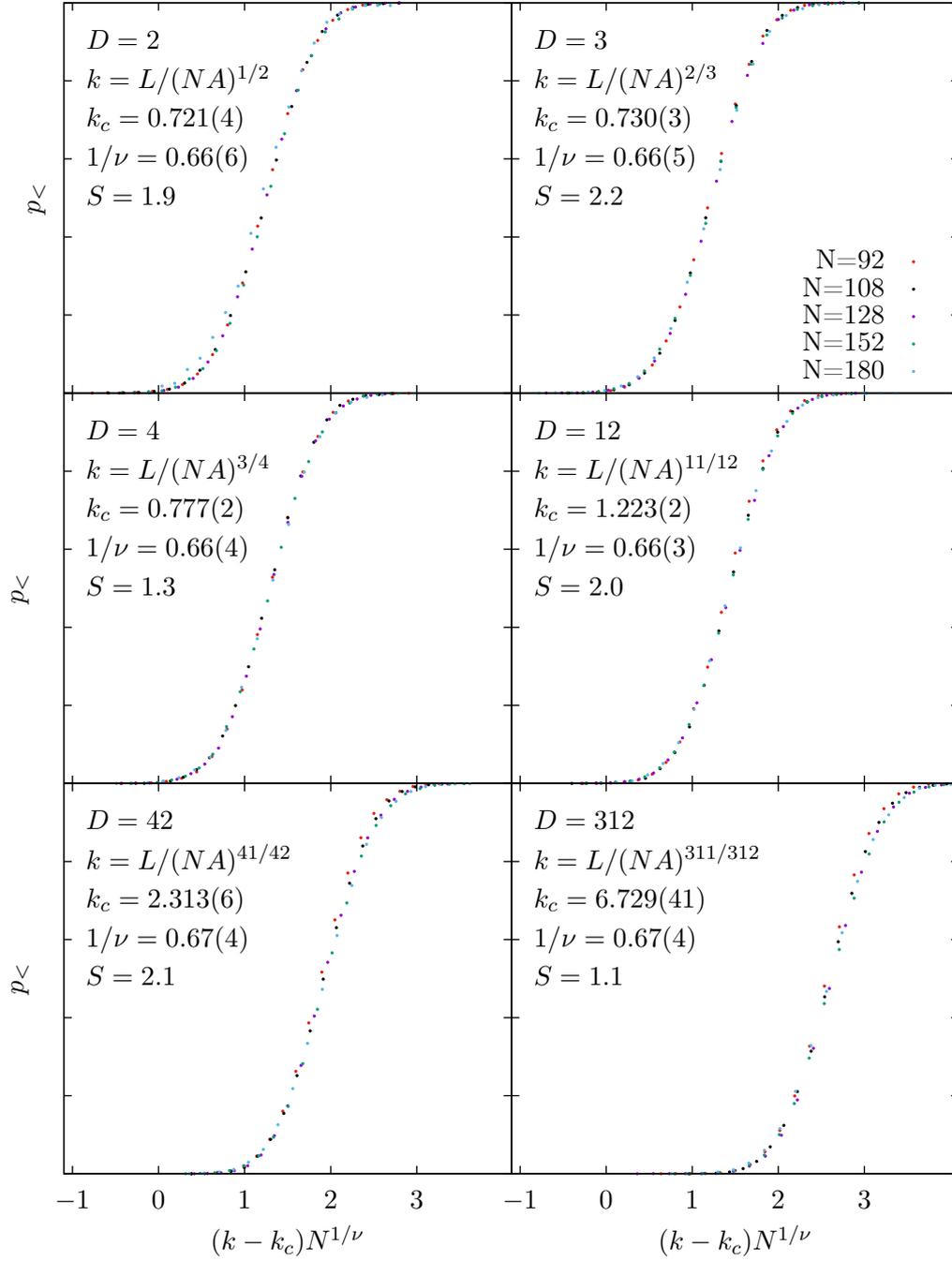


Figure 6.13: FSS analysis of TSP decision version. Here the probability $p_<$ that a tour shorter than L_t exists is plotted over a rescaled parameter $(k - k_c)N^{1/\nu}$ with $k = L/(NA)^{1-1/D}$.

6.6.2. The Fuzzy Circle Ensemble

Clearly for the FCE, the geometrical size grows with N since $R = \frac{N}{2\pi}$. To adapt it to the aforementioned ensemble, this chapter will rescale the results, such that the initial circle has $R = 1$ and thus the tour length L at $\sigma = 0$ is always $L = 2\pi$. This simultaneously shows the need for a new control parameter. Here L instead of k is chosen and is expected to get worse for configurations that deviate strongly from a circle, i.e. small N at moderate σ . There a crossover to the control parameter k is expected, since those realizations resemble a random point process (compare Figure 3.3(f)). This effect is demonstrated in Figure 6.14 where a nice phase transition with an intersection of all sizes at a critical L_c is observable for $\sigma = 2$ in Figure 6.14(a), while in Figure 6.14(b) for $\sigma = 18$ the smallest sizes do show major deviations, as an example of the crossover. However for $N \rightarrow \infty$ this has no influence.

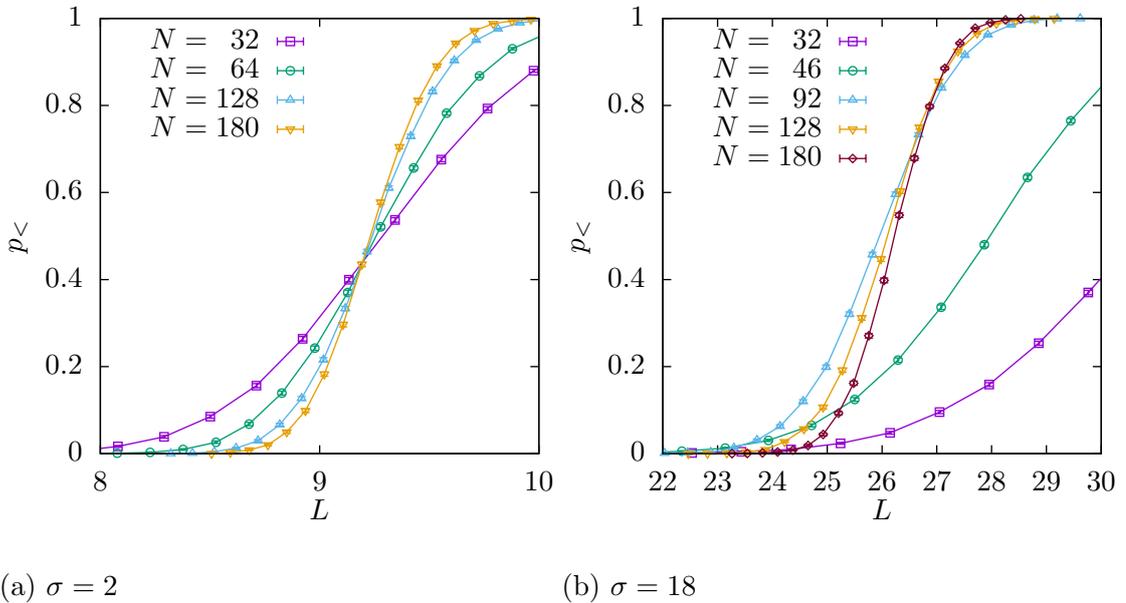


Figure 6.14: The control parameter for the FCE seems to be L for small σ . The intersection in (a) is in the same place for all N . For big σ and small N , i.e. for configurations where $\sigma \approx R = N/2\pi$ and the configuration is more similar to a random point process than a circle, L can not be used as a control parameter anymore and a crossover to some other is visible. In (b) this can be seen for $N = 32$ and $N = 46$.

In Figure 6.15 in the lower right at $\sigma = 18$ the crossover is not visible, since again only instances of $N \geq 92$ are plotted and considered to determine the collapse. The bad S values for $\sigma \in \{6, 18\}$ are probably an effect of this crossover. Considering

that, the collapses seem acceptable, though the uncertainty of $1/\nu$ is ludicrously big, despite the values are very compatible with the expectation, so that the high uncertainty is again probably caused by the effects of the crossover.

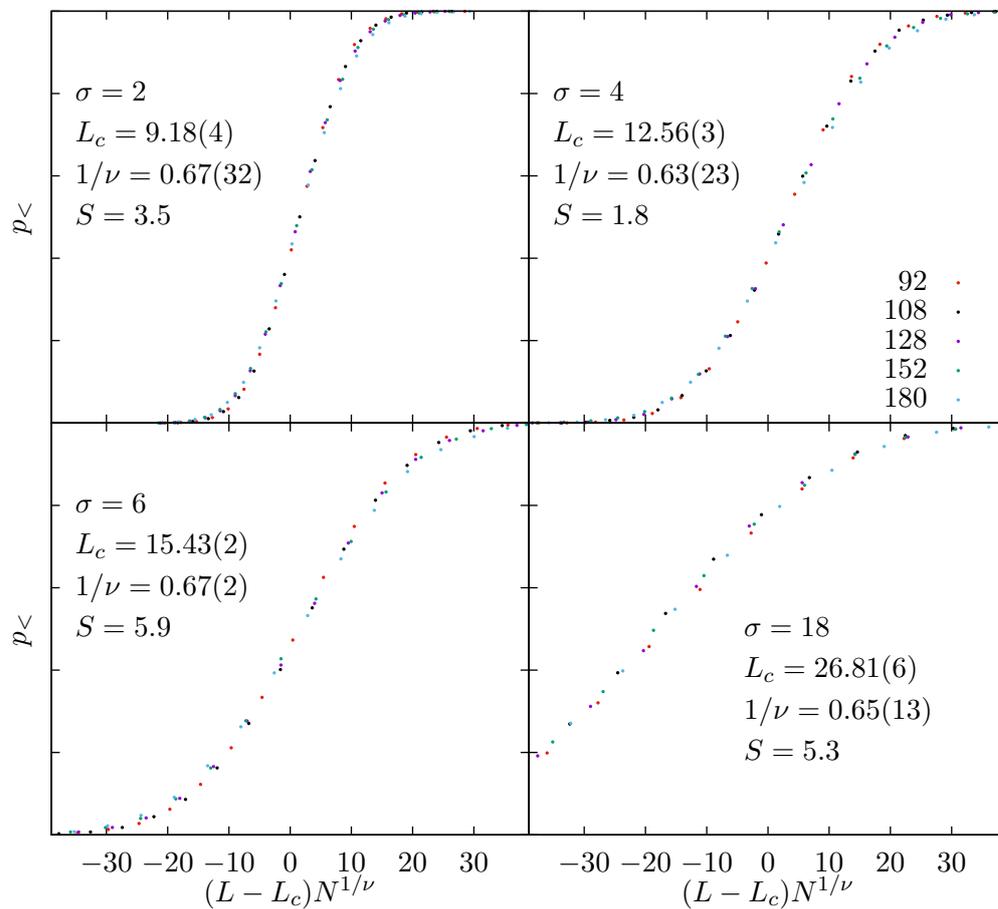


Figure 6.15: Data collapse of the CDF of the tour length L . The obtained ν confirm the values from Figure 6.13 and the collapse works acceptable well. The $p_<$ axis goes from 0 to 1.

7. Conclusion

This thesis shows that the Fuzzy Circle Ensemble driven by the parameter σ of Euclidean TSPs has a continuous phase transition which can be characterized by a critical exponent b . This seems to depend on the type of cuts used, since the exponents from the degree LP relaxation differ from the other two relaxations. The transitions observed in the solvability statistics of the degree LP relaxation and SEC LP relaxation can also be observed in structural properties of the optimal tours, i.e. in the Hamming distance to the initial circle tour x_{ij}° , respectively in the tortuosity τ .

The second result of this thesis is about the first excitation of the ground state, i.e. the second shortest tour, to test whether the energy landscape of this problem shows similarities with glassy systems, where low excitations lead to finite fractions of changes in the configuration. This result shows that for the first excitation of the TSP this is not the case as the fraction of difference, measured as Hamming distance, between the shortest and second shortest tour vanishes in the $N \rightarrow \infty$ limit.

At last some results concerning the decision version of the TSP from older publications are reproduced and extended for higher dimensions. Since the data was already generated for the other topics, this comes as a consistency check at no cost. Also an attempt to extend those results for the FCE is shown and gives plausible results.

8. Outlook

Some possible directions for further study are already laid out in the subsections of the results Section 6, which will not be repeated here. Instead further suggestions which extend this thesis in directions which are not explored here are listed.

In Section 6.2.3 the same critical exponent is observed as in Section 6.2.2. The obvious way to extend this work, is thus to test, if the same critical exponent occurs, for even tighter LP relaxations, like with all blossom inequalities, or with all simple comb constraints [38]. It would be interesting if those would establish a further phase transition at higher σ and if the critical exponent b stays the same.

A further interesting question for further study would be finding an answer to why the tortuosity τ peaks at σ_c . Unfortunately τ is quite complex to measure which likely decreases the chance to analyse it in depth. Therefore the search for a simpler observable showing the transition would be of equal interest.

A more approachable and promising way for further analysis would be properties of the distance matrices c_{ij} themselves, because they are defining the instances completely and start at a pretty simple structure, as for $N \rightarrow \infty$ and $\sigma = 0$ the distance matrix has the form of

$$c_{ij} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & \dots \\ 1 & 0 & 1 & 2 & 3 & \dots \\ 2 & 1 & 0 & 1 & 2 & \dots \\ 3 & 2 & 1 & 0 & 1 & \dots \\ 4 & 3 & 2 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \quad (8.1)$$

It would be interesting if one could look at the properties of the matrices for different σ and maybe find hints for the phase-transition similar to the tour difference from Section 6.3.1 or the tortuosity from Section 6.3.2 but with the advantage that no optimal solution is needed for the evaluation [70].

The same reasoning is behind a deeper look at simple heuristics. Since they are easy to obtain, it would be interesting if their solutions give any hints on the phase transitions from Section 6.2. I already did a small first step in the context of this thesis, which are shortly presented in Appendix A.

A. Heuristics

In practical applications the global optimal tour is usually not needed, but a local optimum within a few percent of the optimal length is sufficient. Therefore a plethora heuristic schemes for the TSP were developed [4–6, 8]. Some of the most simple heuristics will be compared to the optimal solution in Figure A.1 and shortly explained in Appendices A.1 to A.4.

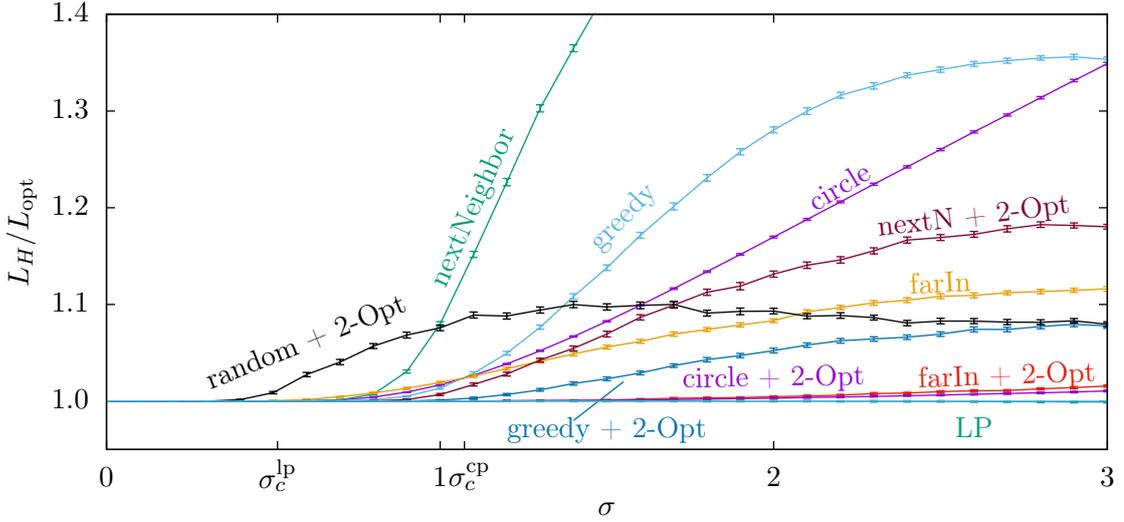


Figure A.1: Comparison of the mean length of a solution of a heuristic L_H compared to the optimal solution L_{opt} for 5000 samples of system size $N = 180$. The exact solution is therefore a horizontal line at $L_H/L_{\text{opt}} = 1$.

The obvious result is that all except “random + 2-Opt” of these simple heuristics yield the optimal solution for $\sigma < \sigma_c^{\text{lp}}$ and start to deviate significantly afterwards. This is a further confirmation, that the approach of determining the simplicity of a city-configuration by looking at the solubility by LP yields plausible results. Also note that the SEC LP relaxation “LP” is always very close to the optimal tour and is an excellent lower bound for the optimal length therefore, though the gap increases for $\sigma \gg 3$ which is not shown in this plot.

A.1. Nearest Neighbor

The *Nearest Neighbor Neighbor* heuristic is a greedy algorithm, which constructs a tour by starting at a random city A and adding the city B nearest to A which

is not already in the tour T .

$$B = \arg \min_{\{i \in V, i \notin T\}} \text{dist}(A, i)$$

So for every of N cities the nearest neighbor from the remaining $N - i$ cities has to be found, which results in a $\mathcal{O}(N^2)$ runtime. This is the method mentioned in the opening quote of Karl Menger. An example tour with $N = 100$ cities optimized by the Nearest Neighbor heuristic is depicted in Figure A.2(a). The start city is marked by an additional circle around the city.

A.2. Greedy

The *Greedy*¹⁶ heuristic for the TSP follows a similar idea as Kruskal's algorithm [71] for minimum spanning trees. It chooses the shortest edge and connects its endpoints if their degree is afterwards not higher than two and the loop would not be closed by this edge, i.e. if it is possible to still create a valid tour after this action. The test whether that edge would construct a loop, can be done in almost constant time using a Union-Find data structure [47, 48] and the test whether the degree of a city does not exceed 2 at any time, is trivial in constant time. Therefore the overall time complexity is determined by the sorting of the N^2 edges according to their length, which is $\mathcal{O}(N^2 \log N)$. An example is shown in Figure A.2(b).

A.3. Farthest Insertion

The *Farthest Insertion* heuristic starts at a city and adds the city farthest away from any city in the tour, i.e.

$$\arg \max_{i \in V \setminus T} \min_{j \in T} \text{dist}(i, j)$$

to the position in the tour such that the resulting tour has a minimal length. This results in a time complexity of $\mathcal{O}(N^3)$. An example is shown in Figure A.2(c). There are also variations which take the nearest city instead of the farthest or take the city which is cheapest to insert, i.e. increases the tour length the least. Because they are conceptually similar, only Farthest Insertion is considered here¹⁷

¹⁶The name Greedy is a bit misleading because, e.g. Nearest Neighbor Appendix A.1 is also greedy.

¹⁷Which is, by the way, the favorite insertion heuristic of William Cook, the maintainer of Concorde [3, p. 69].

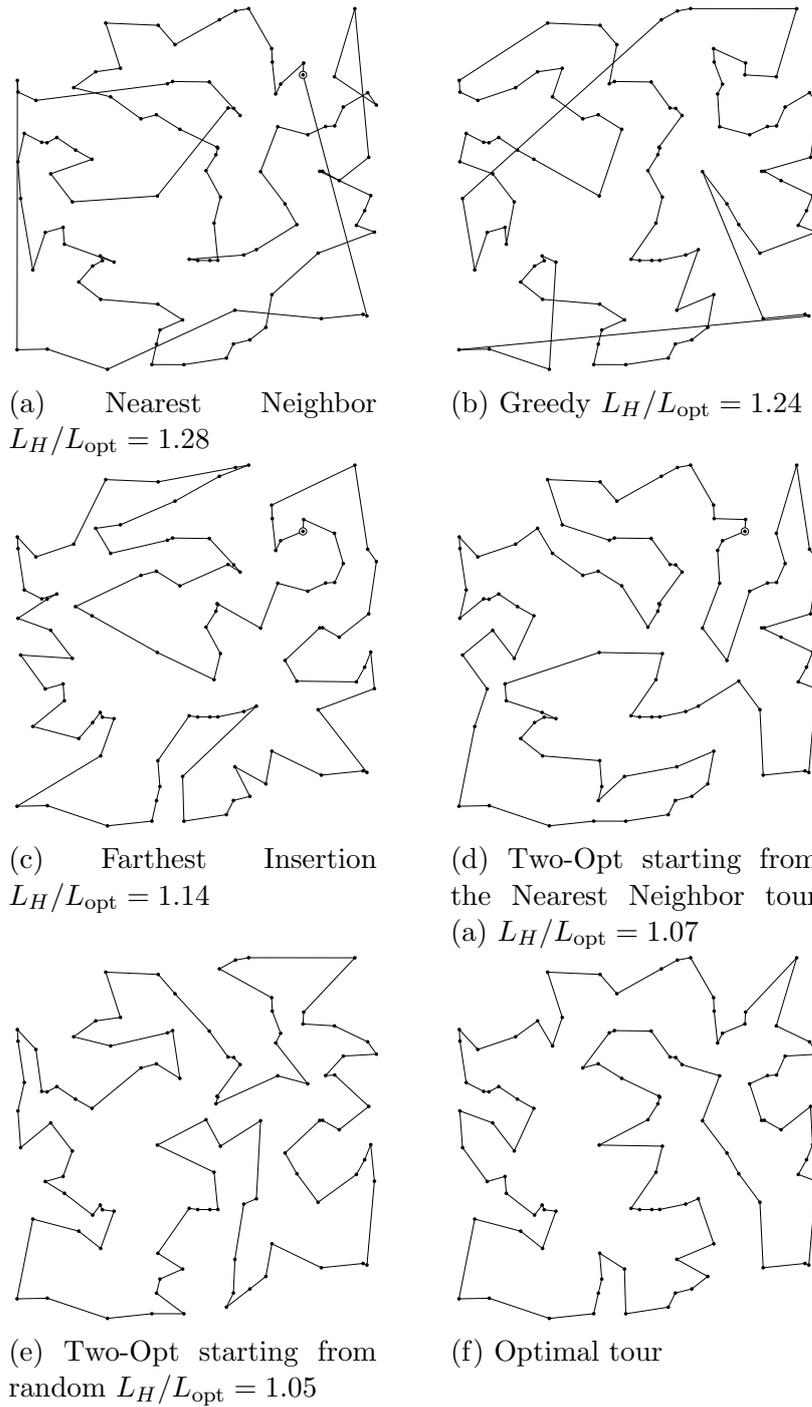


Figure A.2: Examples of the same $N = 100$ configuration optimized by different heuristics. The fraction L_H/L_{opt} is the result for this one example and not suitable to judge the quality of the corresponding heuristic. The starting city, if applicable, is marked with a circle. There is also a simple Python program for this kind of visualization at <https://github.com/surt91/TSPview>.

A.4. Two-Optimal Tours

The Two-Opt heuristic (also 2-Opt or 2-opt) performs a local search. It does not construct a tour from scratch like the ones above, but improves an existing tour. In Figure A.1 the above mentioned heuristics are also improved by Two-Opt and labeled as “Greedy + 2-Opt”. The Two-Opt heuristic chooses two edges $\{a, b\}$ and $\{c, d\}$ and changes their endpoints, such that the tour stays connected, if the resulting tour gets shorter, i.e.

$$\text{dist}(a, c) + \text{dist}(b, d) < \text{dist}(a, b) + \text{dist}(c, d).$$

An example of a swap is depicted in Figures A.3(a) and A.3(b). This step is

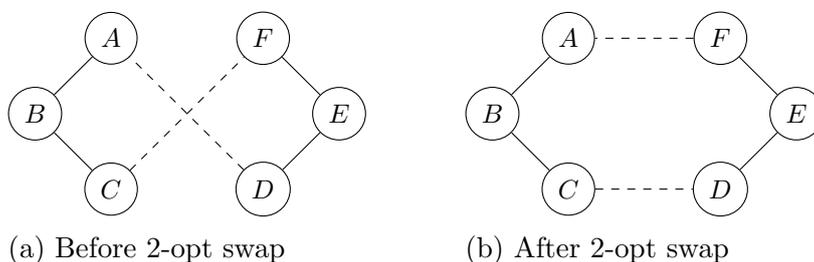
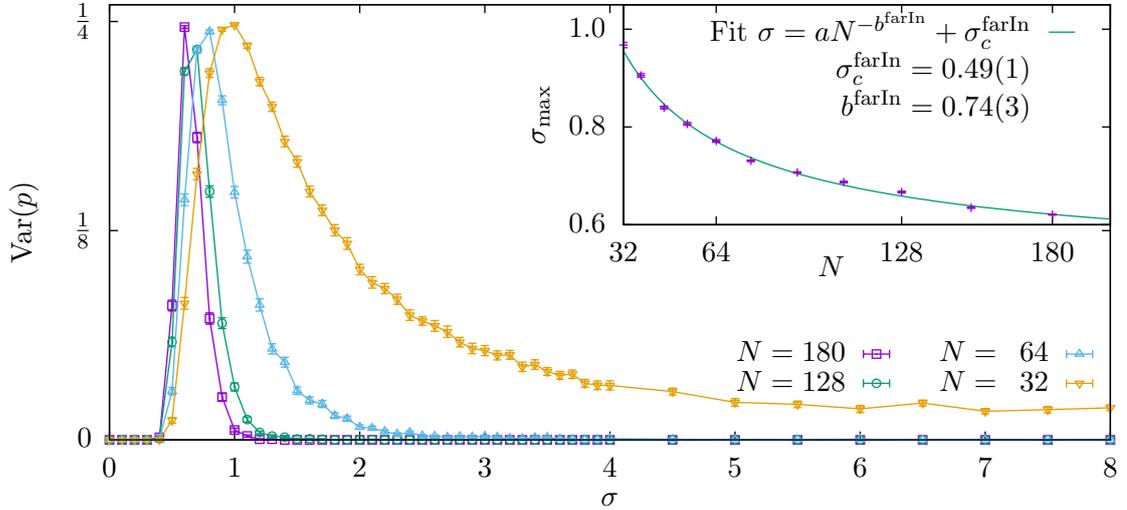


Figure A.3: The two dashed edges from (a) $\{C, F\}$ and $\{A, D\}$ are longer than $\{F, A\}$ and $\{C, D\}$, therefore they are exchanged resulting in a 2-opt tour in (b).

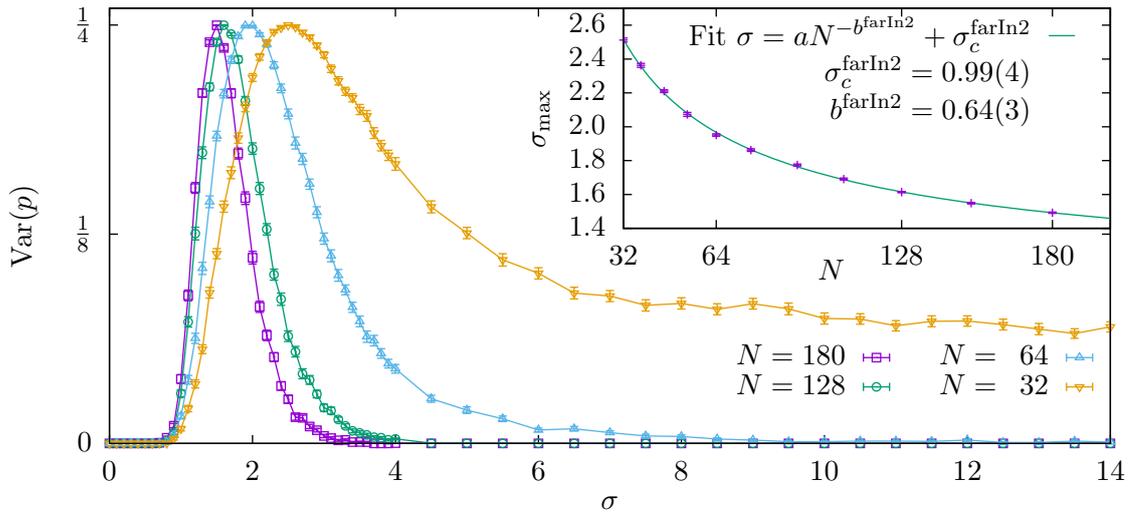
repeated for all $\mathcal{O}(N^2)$ pairs of edges in the tour. If a pair of edges is found to improve the tour, the swap is performed and the algorithm starts over again – that means that exponentially many restarts may have to be performed in the worst case, though starting from a tour created by other heuristics takes typically only very few steps. Two-Opt gets easily trapped in local minima, hence it is dependent on the initial conditions. It often yields good results for its simplicity and is often combined with metaheuristics [72–74]. In particular for metric TSP there are no edges that cross each other in a Two-Opt tour. An example where a random initial tour is optimized is shown in Figure A.2(e). Also note that this heuristic can be extended to change 3 instead of 2 edges to the Three-Opt heuristic [75] or more general the k -Opt heuristics, which change k edges.

A.5. Analysis

Note that the same kind of analysis as in Section 6.2 can also be performed for these heuristics – but then the problem has to be solved to optimality, otherwise there is no way to determine if the heuristic found the optimal solution. Therefore it is



(a) Farthest Insertion



(b) Farthest Insertion and Two-Opt

Figure A.4: Variance of the probability that a simple heuristic finds the optimal solution at different σ and positions of maximal $\text{Var}(p)$ fitted by a power-law in the inset. Compare to Figure 6.2(c).

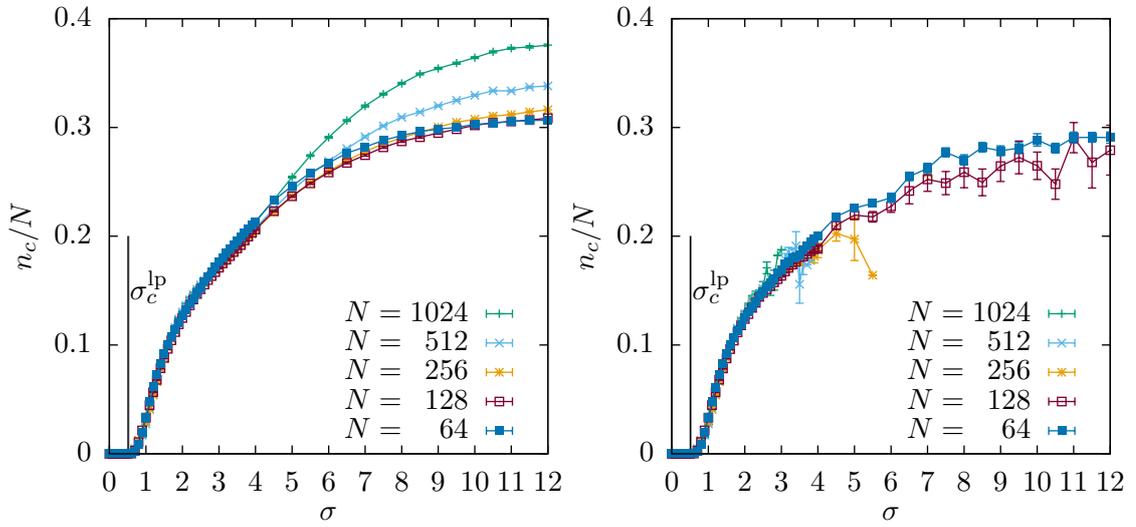
only feasible for small systems and therefore especially susceptible to corrections to scaling – mind that for the corresponding fits in Section 6.2 only $N \geq 256$ are considered, while here the largest is $N = 180$, therefore the obtained values should be handled with care. Nevertheless those heuristics show transitions but with different critical points and exponents.

B. Run Time Metrics

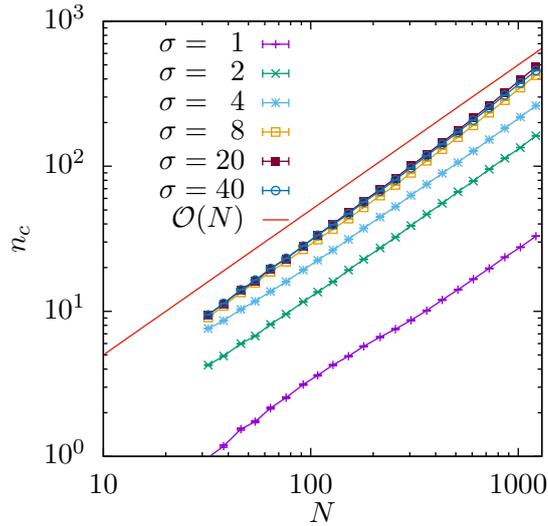
In Figure B.1(c) the number of used cutting planes n_c is plotted against the system size N with the purpose of showing that only polynomial many are needed. For this plot all cuts added are counted. Some of the added cuts can be redundant, especially some of the SECs generated by Karger’s minimum cut algorithm could chop off a subset of the solution space already constraint by another SECs from the same iteration, i.e. those cuts are *not tight*. Also note that many of the SECs are generated by an analysis of the connected components, that is as long as the graph is not connected, for each connected component a SEC is added, which also could introduce a few not-tight cuts. For more details see Section 4.2 and especially Figure C.1. Despite CPLEX is able to discard some cuts which are not tight, the analysis will focus on all cuts, to minimize the influence of the *black box* CPLEX. All in all the number of cuts plotted here is not the number of strictly needed SECs but an upper bound on that, which is sufficient to show the approximately polynomial growth of the needed cuts.

A quite obvious method to measure hardness of a problem would be to count the seconds a (sufficiently good) program needs to solve it. In practice this poses some difficulties, because it is not only dependent on complicated things like compiler flags, hardware, the system load of the compute-cluster and especially the skill of the programmer, but also gives no real insight. The next best thing is to count some fundamental steps of the algorithm. In this study a branch-and-cut algorithm is used with well defined possible cuts, the SEC. As a measure of hardness we could thus count the number of cuts n_c done which is visualized in Figure B.2(b), or the number of nodes n_n added to the search tree until the optimal solution is found in Figure B.2(a). Note that the number of nodes could be underestimated, because during the run instances which search trees grow beyond a certain threshold are discarded. Note that n_c is divided by N , because Figure B.1(c) shows $n_c \approx N$.

As a crosscheck σ_c^{cp} and σ_c^{lp} are also displayed. σ_c^{lp} should mark the spot where the number of cuts is first $n_c > 0$, because beyond this point the degree LP relaxation will start to yield infeasible solutions and needs to be augmented with the SEC as cutting planes. Similarly σ_c^{cp} should mark the spot where the number



(a) (b)



(c)

Figure B.1: (a) and (b) show the number of cuts added to the formulation. (a) shows the statistic for all instances and (b) is limited to the instances that are solvable by the SEC LP relaxation. (c) shows the number of SECs added to the problem via cutting planes until no SECs are violated by the LP relaxation. This – though not exactly linear – seems to not grow substantially faster than the linear function $f(N) = \frac{N}{2}$.

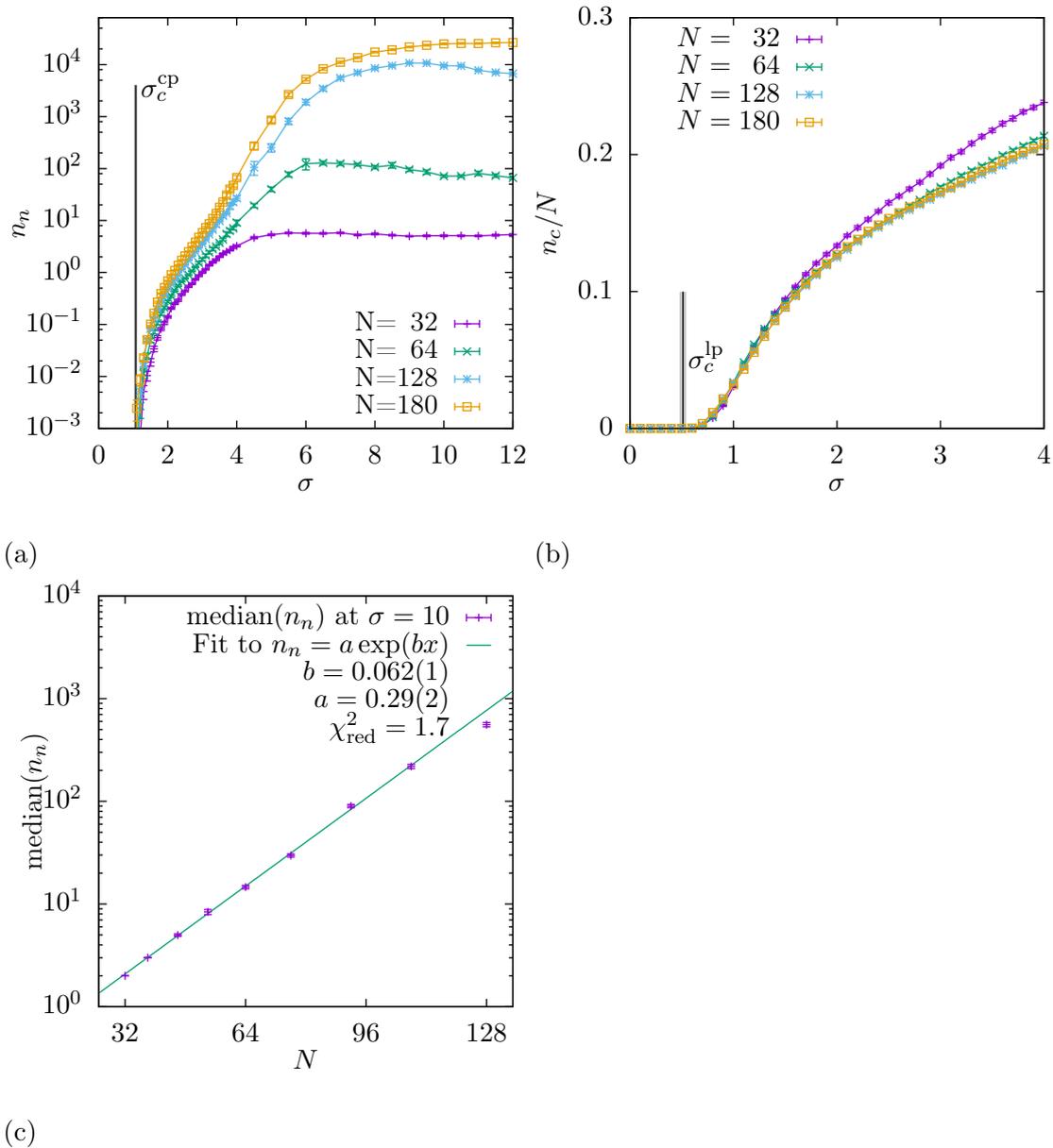


Figure B.2: Number of nodes n_n in the search tree of the branch-and-cut to find the optimal tour. In (a) as a function of the tuning parameter σ and in (c) the median of the number of nodes as a function of the system size N showing the exponential growth of the run time. (b) shows the number of cuts in the branch-and-cut.

of nodes first exceeds 0 (not counting the root node), since before all instances are solvable by the SEC LP relaxation and afterwards the exhaustive search has to be started. Though both statements are only strictly true for infinite N , both effects are quite well observable in Figures B.2(a) and B.2(b). It is also expected that the running time increases exponentially with N on the hard side of the transition, this is actually well observable in the number of nodes as shown in Figure B.2(c) for $\sigma = 10$ where a straight line in a half-logarithmic plot shows the exponential relationship. Note that the largest system sizes do not appear in this plot, since the statistic of those is skewed since the instances with many nodes are aborted because of memory constraints.

Especially interesting is that phase-transitions in **NP**-complete problems from *yes* to *no* are accompanied by a peak in the hardness, i.e. run time needed, to solve it. For this **NP**-hard problem, this phase-transition is followed by a hard region.

C. Program Flow Diagram

Here a program flow diagram to clarify the separation of the SEC is shown. This is just an extension of Section 4.2.

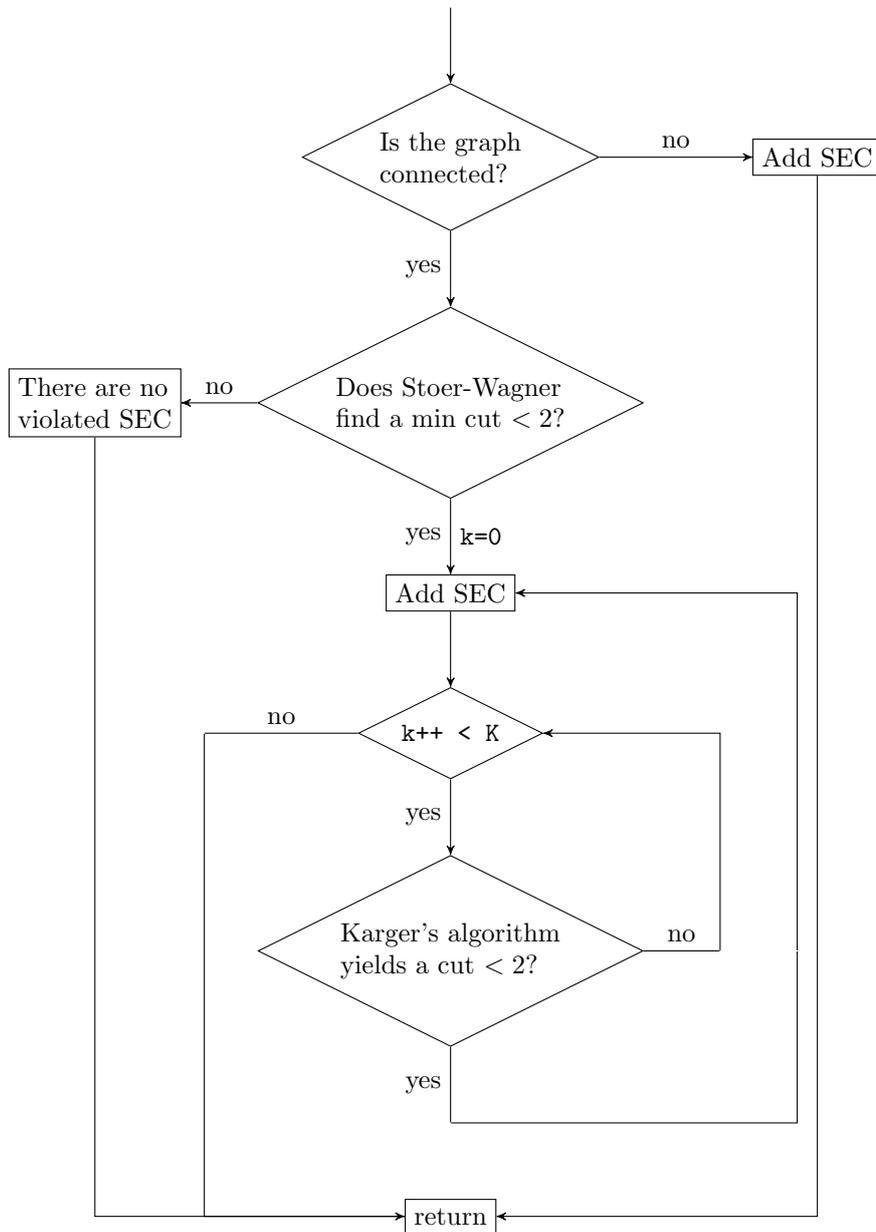


Figure C.1: Program flow diagram of the SEC separation used in this study. See also Section 4.2.

References

- [1] Alexander Schrijver. “On the history of combinatorial optimization (till 1960)”. In: *Handbooks in Operations Research and Management Science: Discrete Optimization* 12 (2005), p. 1.
- [2] Karl Menger. “Bericht über ein mathematisches Kolloquium 1929/30”. German. In: *Monatshefte für Mathematik und Physik* 38.1 (1931), pp. 17–38. ISSN: 0026-9255. DOI: 10.1007/BF01700678.
- [3] William Cook. *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation*. Princeton University Press, 2012. ISBN: 9780691152707.
- [4] Scott Kirkpatrick, M.P. Vecchi, et al. “Optimization by simulated annealing”. In: *science* 220.4598 (1983), pp. 671–680.
- [5] Fred Glover. “Future paths for integer programming and links to artificial intelligence”. In: *Computers and Operations Research* 13.5 (1986). Applications of Integer Programming, pp. 533–549. ISSN: 0305-0548. DOI: 10.1016/0305-0548(86)90048-1.
- [6] Marco Dorigo and Luca Maria Gambardella. “Ant colony system: a cooperative learning approach to the traveling salesman problem”. In: *Evolutionary Computation, IEEE Transactions on* 1.1 (1997). <http://www.papercore.org/Dorigo1997>, pp. 53–66.
- [7] Gerhard Reinelt. “TSPLIB—A traveling salesman problem library”. In: *ORSA journal on computing* 3.4 (1991). <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>, pp. 376–384.
- [8] Shen Lin and Brian W. Kernighan. “An effective heuristic algorithm for the traveling-salesman problem”. In: *Operations research* 21.2 (1973), pp. 498–516.
- [9] Stefan Hougardy and Rasmus T. Schroeder. “Edge Elimination in TSP Instances”. In: *Graph-Theoretic Concepts in Computer Science*. Ed. by Dieter Kratsch and Ioan Todinca. Vol. 8747. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 275–286. ISBN: 978-3-319-12339-4. DOI: 10.1007/978-3-319-12340-0_23.
- [10] Christos H. Papadimitriou. “The Euclidean travelling salesman problem is NP-complete”. In: *Theoretical Computer Science* 4.3 (1977), pp. 237–244. ISSN: 0304-3975. DOI: 10.1016/0304-3975(77)90012-3.
- [11] Sanjeev Arora. “Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems”. In: *Journal of the ACM (JACM)* 45.5 (1998), pp. 753–782.

- [12] Rainer E. Burkard, Vladimir G. Deineko, René van Dal, Jack A. A. van der Veen, and Gerhard J. Woeginger. *Well-Solvable Special Cases of the TSP: A Survey*. 1995.
- [13] Francisco Barahona. “On the computational complexity of Ising spin glass models”. In: *Journal of Physics A: Mathematical and General* 15.10 (1982), p. 3241.
- [14] Richard M. Karp. *Reducibility among combinatorial problems*. Springer, 1972.
- [15] Olivier C. Martin, Rémi Monasson, and Riccardo Zecchina. “Statistical mechanics methods and phase transitions in optimization problems”. In: *Theoretical computer science* 265.1 (2001), pp. 3–67.
- [16] Scott Kirkpatrick and Bart Selman. “Critical Behavior in the Satisfiability of Random Boolean Expressions”. In: *Science* 264.5163 (1994), pp. 1297–1301. DOI: 10.1126/science.264.5163.1297.
- [17] Giulio Biroli, Simona Cocco, and Rémi Monasson. “Phase transitions and complexity in computer science: an overview of the statistical physics approach to the random satisfiability problem”. In: *Physica A: Statistical Mechanics and its Applications* 306 (2002). Invited Papers from the 21th IUPAP International Conference on Statistical Physics, pp. 381–394. ISSN: 0378-4371. DOI: 10.1016/S0378-4371(02)00516-2.
- [18] Martin Weigt and Alexander K. Hartmann. “Number of Guards Needed by a Museum: A Phase Transition in Vertex Covering of Random Graphs”. In: *Phys. Rev. Lett.* 84 (26 June 2000), pp. 6118–6121. DOI: 10.1103/PhysRevLett.84.6118.
- [19] Alexander K. Hartmann and Martin Weigt. “Statistical mechanics of the vertex-cover problem”. In: *Journal of Physics A: Mathematical and General* 36.43 (2003), p. 11069.
- [20] Florent Krzakala, Andrea Montanari, Federico Ricci-Tersenghi, Guilhem Semerjian, and Lenka Zdeborová. “Gibbs states and the set of solutions of random constraint satisfaction problems”. In: *Proceedings of the National Academy of Sciences* 104.25 (2007), pp. 10318–10323.
- [21] Ian P. Gent and Toby Walsh. “The TSP phase transition”. In: *Artificial Intelligence* 88.1–2 (1996), pp. 349–358. ISSN: 0004-3702. DOI: 10.1016/S0004-3702(96)00030-6.
- [22] Timo Dewenter and Alexander K. Hartmann. “Phase transition for cutting-plane approach to vertex-cover problem”. In: *Physical Review E* 86.4 (2012), p. 041128.

- [23] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.
- [24] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. 2nd. McGraw-Hill Higher Education, 2001. ISBN: 0070131511.
- [25] Donald E. Knuth. *The Art of Computer Programming: Sorting and searching*. Addison-Wesley series in computer science and information processing. Addison-Wesley Publishing Company, 1968. ISBN: 9780201038033.
- [26] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. “PRIMES Is in P”. English. In: *Annals of Mathematics*. Second Series 160.2 (2004), ISSN: 0003486X.
- [27] James A. Carlson, Arthur Jaffe, and Andrew Wiles. *The millennium prize problems*. <http://www.claymath.org/millennium-problems>. American Mathematical Soc., 2006.
- [28] Herbert Edelsbrunner, Günter Rote, and Ermo Welzl. “Testing the necklace condition for shortest tours and optimal factors in the plane”. In: *Theoretical Computer Science* 66.2 (1989), pp. 157–180. ISSN: 0304-3975. DOI: 10.1016/0304-3975(89)90133-3.
- [29] Merrill M. Flood. “The Traveling-Salesman Problem”. In: *Operations Research* 4.1 (1956), pp. 61–75. DOI: 10.1287/opre.4.1.61.
- [30] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Books on Computer Science Series. Dover Publications, 1998. ISBN: 9780486402581.
- [31] William H. Press. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.
- [32] George Dantzig, Ray Fulkerson, and Selmer Johnson. “Solution of a large-scale traveling-salesman problem”. In: *Journal of the operations research society of America* 2.4 (1954). <http://www.papercore.org/Dantzig1954>, pp. 393–410.
- [33] C. E. Miller, A. W. Tucker, and R. A. Zemlin. “Integer Programming Formulation of Traveling Salesman Problems”. In: *J. ACM* 7.4 (Oct. 1960), pp. 326–329. ISSN: 0004-5411. DOI: 10.1145/321043.321046.
- [34] Václav Chvátal. “Edmonds polytopes and weakly Hamiltonian graphs”. In: *Mathematical Programming* 5.1 (1973), pp. 29–40.
- [35] Martin Grötschel and Manfred W. Padberg. “On the symmetric travelling salesman problem I: inequalities”. In: *Mathematical Programming* 16.1 (1979), pp. 265–280.

- [36] Martin Grötschel and Manfred W. Padberg. “On the symmetric travelling salesman problem II: Lifting theorems and facets”. In: *Mathematical Programming* 16.1 (1979), pp. 281–302.
- [37] Adam N. Letchford and Andrea Lodi. “Polynomial-time separation of simple comb inequalities”. In: *Integer Programming and Combinatorial Optimization*. Springer, 2002, pp. 93–108.
- [38] Lisa K. Fleischer, Adam N. Letchford, and Andrea Lodi. “Polynomial-time separation of a superclass of simple comb inequalities”. In: *Mathematics of Operations Research* 31.4 (2006), pp. 696–713.
- [39] David Applegate, Robert Bixby, Vašek Chvátal, and William Cook. “Implementing the Dantzig-Fulkerson-Johnson algorithm for large traveling salesman problems”. In: *Mathematical programming* 97.1-2 (2003), pp. 91–153.
- [40] Mechthild Stör and Frank Wagner. “A simple min-cut algorithm”. In: *Journal of the ACM (JACM)* 44.4 (1997). <http://www.papercore.org/Stoer1997>, pp. 585–591.
- [41] Robert Tarjan. “Depth-first search and linear graph algorithms”. In: *SIAM journal on computing* 1.2 (1972), pp. 146–160.
- [42] David R. Karger. “Global Min-cuts in RNC, and Other Ramifications of a Simple Min-out Algorithm”. In: *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '93. Austin, Texas, USA: Society for Industrial and Applied Mathematics, 1993, pp. 21–30. ISBN: 0-89871-313-7.
- [43] R. E. Gomory and T. C. Hu. “Multi-Terminal Network Flows”. English. In: *Journal of the Society for Industrial and Applied Mathematics* 9.4 (1961), ISSN: 03684245.
- [44] Martin Grötschel, László Lovász, and Alexander Schrijver. “The ellipsoid method and its consequences in combinatorial optimization”. In: *Combinatorica* 1.2 (1981), pp. 169–197.
- [45] Martin Grötschel, László Lovász, and Alexander Schrijver. “Geometric algorithms and combinatorial optimization”. In: *Algorithms and Combinatorics* 2 (1993), pp. 1–362.
- [46] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. “Valid Inequalities for Structured Integer Programs”. In: *Integer Programming*. Vol. 271. Graduate Texts in Mathematics. Springer International Publishing, 2014, pp. 281–319. ISBN: 978-3-319-11007-3. DOI: 10.1007/978-3-319-11008-0_7.

- [47] M. E. J. Newman and R. M. Ziff. “Fast Monte Carlo algorithm for site or bond percolation”. In: *Phys. Rev. E* 64 (1 June 2001), p. 016706. DOI: 10.1103/PhysRevE.64.016706.
- [48] Robert Endre Tarjan. “A class of algorithms which require nonlinear time to maintain disjoint sets”. In: *Journal of Computer and System Sciences* 18.2 (1979), pp. 110–127. ISSN: 0022-0000. DOI: 10.1016/0022-0000(79)90042-4.
- [49] Jack Edmonds. “Maximum matching and a polyhedron with 0, 1-vertices”. In: *J. Res. Nat. Bur. Standards B* 69.1965 (1965), pp. 125–130.
- [50] <https://en.wikipedia.org/wiki/Flower>, accessed 18.09.2015, 12:19.
- [51] Manfred W. Padberg and M. Ram Rao. “Odd minimum cut-sets and b-matchings”. In: *Mathematics of Operations Research* 7.1 (1982), pp. 67–80.
- [52] Manfred Padberg and Giovanni Rinaldi. “A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems”. English. In: *SIAM Review* 33.1 (1991), ISSN: 00361445. DOI: 10.1137/1033004.
- [53] M. Padberg and G. Rinaldi. “Optimization of a 532-city symmetric traveling salesman problem by branch and cut”. In: *Operations Research Letters* 6.1 (1987), pp. 1–7. ISSN: 0167-6377. DOI: 10.1016/0167-6377(87)90002-2.
- [54] D. Stauffer and A. Aharony. *Introduction To Percolation Theory*. Taylor & Francis, 1994. ISBN: 9781420074796.
- [55] N. Goldenfeld. *Lectures on Phase Transitions and the Renormalization Group*. Frontiers in physics. Addison-Wesley, Advanced Book Program, 1992. ISBN: 9780201554090.
- [56] H. Eugene Stanley. “Scaling, universality, and renormalization: Three pillars of modern critical phenomena”. In: *Reviews of modern physics* 71.2 (1999), S358.
- [57] Kurt Binder and Dieter W. Heermann. *Introduction: Purpose and Scope of This Volume, and Some General Comments*. Graduate Texts in Physics. Springer Berlin Heidelberg, 2010. ISBN: 978-3-642-03162-5. DOI: 10.1007/978-3-642-03163-2_1.
- [58] B. Efron. “Bootstrap Methods: Another Look at the Jackknife”. In: *Ann. Statist.* 7.1 (Jan. 1979), pp. 1–26. DOI: 10.1214/aos/1176344552.
- [59] A. Peter Young. *Everything You Wanted to Know About Data Analysis and Fitting but Were Afraid to Ask*. SpringerBriefs in Physics. Springer International Publishing, 2015. ISBN: 978-3-319-19050-1. DOI: 10.1007/978-3-319-19051-8_2.

- [60] Alexander K. Hartmann. *Big Practical Guide to Computer Simulations*. World Scientific, 2015. DOI: 10.1142/9019.
- [61] Richard W. Hamming. “Error detecting and error correcting codes”. In: *Bell System technical journal* 29.2 (1950), pp. 147–160.
- [62] E. Grisan, M. Foracchia, and A. Ruggeri. “A novel method for the automatic evaluation of retinal vessel tortuosity”. In: *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*. Vol. 1. Sept. 2003, 866–869 Vol.1. DOI: 10.1109/IEMBS.2003.1279902.
- [63] Personal Communication. Timo Dewenter.
- [64] M. Zumsande and A.K. Hartmann. “Low-energy excitations in the three-dimensional random-field Ising model”. In: *The European Physical Journal B* 72.4 (2009), pp. 619–627.
- [65] Michel X. Goemans and Dimitris J. Bertsimas. “Probabilistic analysis of the Held and Karp lower bound for the Euclidean traveling salesman problem”. In: *Mathematics of operations research* 16.1 (1991), pp. 72–89.
- [66] D.S. Johnson, L.A. McGeoch, and E.E. Rothberg. “Asymptotic Experimental Analysis for the Held-Karp Traveling Salesman Bound”. In: *Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms*. Vol. 81. SIAM. 1996, p. 341.
- [67] Allon G. Percus and Olivier C. Martin. “Finite size and dimensional dependence in the Euclidean traveling salesman problem”. In: *Physical Review Letters* 76.8 (1996), p. 1188.
- [68] Allon G. Percus and Olivier C. Martin. “The stochastic traveling salesman problem: Finite size scaling and the cavity prediction”. In: *Journal of Statistical Physics* 94.5-6 (1999), pp. 739–758.
- [69] Oliver Melchert. *autoScale.py - A program for automatic finite-size scaling analyses: A user’s guide*. arXiv:0910.5403 [physics.comp-ph], https://github.com/omelchert/SCALING_ANALYSIS. 2009.
- [70] Personal Communication. Satoshi Takabe.
- [71] Joseph B. Kruskal. “On the shortest spanning subtree of a graph and the traveling salesman problem”. In: *Proceedings of the American Mathematical Society* 7 (1956), pp. 48–50. DOI: 10.1090/S0002-9939-1956-0078686-7.
- [72] Olivier C. Martin and Steve W. Otto. “Combining simulated annealing with local search heuristics”. English. In: *Annals of Operations Research* 63.1 (1996), pp. 57–75. ISSN: 0254-5330. DOI: 10.1007/BF02601639.

- [73] NicoL.J. Ulder, EmileH.L. Aarts, Hans-Jürgen Bandelt, PeterJ.M. van Laarhoven, and Erwin Pesch. “Genetic local search algorithms for the traveling salesman problem”. English. In: *Parallel Problem Solving from Nature*. Ed. by Hans-Paul Schwefel and Reinhard Männer. Vol. 496. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1991, pp. 109–116. ISBN: 978-3-540-54148-6. DOI: 10.1007/BFb0029740.
- [74] Li-Pei Wong, Malcolm Yoke Hean Low, and Chin Soon Chong. “Bee colony optimization with local search for traveling salesman problem”. In: *International Journal on Artificial Intelligence Tools* 19.03 (2010), pp. 305–334.
- [75] Shen Lin. “Computer solutions of the traveling salesman problem”. In: *Bell System Technical Journal, The* 44.10 (1965), pp. 2245–2269.

Glossary

Boost Graph Library

Boost is an open source library collection <http://www.boost.org/>.. 19

Concorde

Concorde is a state of the art solver for the TSP, free for academic use and open source. <http://www.math.uwaterloo.ca/tsp/concorde.html>. 11, 18, 30, 33

CPLEX

CPLEX is a commercial LP and MIP solver from IBM, which can be used as a C++ library and is free for academic use. <http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud/>. 18, 30, 57

degree LP relaxation

The degree LP relaxation is the LP defined by the by Equations (4.1), (4.3) and (4.7).. 29–33, 36, 37, 44, 51, 60

fast Blossom LP relaxation

The fast Blossom LP relaxation is the LP defined by Equations (4.1), (4.3), (4.4) and (4.7) and some constraints of type Equation (4.13).. 30, 33, 35, 36, 44

Gnuplot

Gnuplot is a free and open source graphing utility. <http://gnuplot.info/>.
29

High-End Computing Resource Oldenburg

HERO is the HPC cluster of the Universität Oldenburg. It is funded by the DFG through its Major Research Instrumentation Programme (INST 184/108-1 FUGG) and the Ministry of Science and Culture (MWK) of the Lower Saxony State.. 30

Library for Efficient Modeling and Optimization in Networks

The LEMON Graph Library is an open source graph library of the COIN-OR project <https://lemon.cs.elte.hu/trac/lemon>. 19

LP relaxation

A LP relaxation is a LP corresponding to some IP, i.e. it does not include any integer constraints and might yield solutions, which are infeasible for the corresponding IP.. 17, 19, 22, 24, 25, 30, 51

NP

See Section 2.. 8, 9, 14, 46

NP-complete

See Section 2.. 6, 8, 9, 26, 60

NP-hard

See Section 2.. 5, 6, 9, 14, 23, 46, 60

P

See Section 2.. 8, 14

SEC LP relaxation

The SEC LP relaxation is the LP defined by Equations (4.1), (4.3), (4.4) and (4.7).. 17, 29, 30, 33, 34, 36, 40, 41, 43, 44, 51, 53, 59, 60

TSPLIB

TSPLIB is a collection of real world, typically difficult TSP problems and their optimal solutions [7]. 5, 21

Acronyms

BGL

Boost Graph Library. 19, *Glossary*: Boost Graph Library

CDF

cumulative distribution function. 46, 47, 50

DFS

depth first search. 19, 21, 31, 33

FCE

Fuzzy Circle Ensemble. 10, 39, 40, 44, 45, 49, 51

FSS

finite-size scaling. 27, 28, 40, 46, 48

HERO

High-End Computing Resource Oldenburg. 30, *Glossary*: High-End Computing Resource Oldenburg

IP

integer programming. 9, 17, 18, 23, 24, 33, 44

LEMON

Library for Efficient Modeling and Optimization in Networks. 19, *Glossary*: Library for Efficient Modeling and Optimization in Networks

LP

linear programming. 6, 13, 14, 17, 18, 22, 23, 30, 40

MIP

mixed integer programming. 18

PTAS

polynomial-time approximation scheme. 5

RFIM

random-field Ising model. 44, 45

SEC

subtour elimination constraint. 14–19, 21, 22, 30, 57, 59, 60, 62

TSP

Travelling Salesperson problem. 5, 6, 9, 10, 14, 15, 17, 18, 21–23, 33, 39, 40, 42, 44–48, 51, 53, 54, 56

VC

Vertex Cover. 6, 40

Acknowledgments

I want to acknowledge Alexander K. Hartmann for all the advises which prevented this project from being stuck at some points and resulted in different branches of this thesis.

A major factor for the success of this thesis were the nice colleagues, Pascal Fieth and Timo Dewenter, Sebastian von Ohr and Roman Bleim, with whom I discussed countless (mainly number-theoretic and combinatoric) problems and Christoph Norrenbrock and Mitchell Mkrtchian for the steady supply of coffee and in depth Star Trek discussions.

I also acknowledge the guests of this working group to whom I talked about this thesis, Dr. Giulio Biroli, Professor Ofer Biham and Professor Eytan Katzav, Professor Peter Young, Satoshi Takabe, Professor Martin Oettel and Professor Reimer Kühn. Every time I explained this work, I understood it a bit better. Unfortunately some of this conversations took place in a phase of this thesis where it was too late to follow the ideas developed during the discussions.

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Außerdem versichere ich, dass ich die allgemeinen Prinzipien wissenschaftlicher Arbeit und Veröffentlichung, wie sie in den Leitlinien guter wissenschaftlicher Praxis der Carl von Ossietzky Universität Oldenburg festgelegt sind, befolgt habe.

Oldenburg den 6. November 2015,
(Hendrik Schawe)