13. Polynomial Time Approximation Schemes for Geometric Optimization Problems in Euclidean Metric Spaces

Richard Mayr, Annette Schelten

13.1 Introduction

Many geometric optimization problems which are interesting and important in practice have been shown to be $\mathcal{NP}$-hard. However, for some cases it is possible to compute approximate solutions in polynomial time.

This chapter describes approaches to find polynomial time algorithms that compute approximate solutions to various geometric optimization problems in Euclidean metric spaces. It is mainly based on the recent work of Arora [Aro97], which significantly improved previous works by Arora [Aro96] and Mitchell [Mit96]. First let us define the problems we will consider:

TRAVELING SALESMAN PROBLEM (TSP)

Instance: A graph $G = (V, E)$ on n nodes with edge weights defined by $c : E \longrightarrow \mathbb{R}$.

Problem: Find a cycle in G with $|V| = n$ edges that contains all nodes and has minimal cost.

Such a cycle is called the *salesman tour* for G. A variant of this problem is k-TSP where for a given $k < n$ the aim is to find the k vertices with the shortest traveling salesman tour, together with this tour. Both problems are $\mathcal{NP}$-hard.

In practice the graph is often embedded in the Euclidean space $\mathbb{R}^d$ with dimension $d \in \mathbb{N}$. If additionally the distances are given by the Euclidean norm (l_2-norm) we speak of *Euclidean* TSP:

EUCLIDEAN TSP

Instance: A graph $G = (V, E)$ in $\mathbb{R}^d$ on n nodes with edge weights defined by the Euclidean norm.

Problem: Find a cycle in G with $|V| = n$ edges that contains all nodes and has minimal cost.

Since the distances between the nodes are given by a norm, we are in particular dealing with complete graphs. The more general case that the graph lies in an arbitrary metric space is called *metric* TSP.

It has long been known that TSP is $\mathcal{NP}$-hard [Kar72]. Later it was shown that this even holds for the special case of Euclidean TSP [Pap77, GGJ76]. While – under the assumption that $\mathcal{P} \neq \mathcal{NP}$ – metric TSP and many other problems do not have a polynomial time approximation scheme (proved by Arora, Lund, Motwani, Sudan and Szegedy [ALM+92]), there is a polynomial time approximation scheme (PTAS) for Euclidean TSP.

In the following we will describe such a PTAS where the graph lies in the Euclidean space $\mathbb{R}^2$. It can be generalized to the d-dimensional Euclidean space $\mathbb{R}^d$, as well as to other related problems (see Section 13.5).

Arora's [Aro97] new algorithm is a randomized PTAS, which finds a $(1+\frac{1}{c})$-approximation to Euclidean TSP in time $\mathcal{O}(n(\log n)^{\mathcal{O}(c)})$ where $c > 0$ and n denotes the number of given nodes. In earlier papers Arora [Aro96] and Mitchell [Mit96] independently obtained PTAS that need $\mathcal{O}(n^{\mathcal{O}(c)})$ time.

Quoting the author, the main idea of the PTAS relies "on the fact that the plane can be recursively partitioned such that some $(1+\frac{1}{c})$-approximate salesman tour crosses each line of the partition at most $t = \mathcal{O}(c \log n)$ times. Such a tour can be found by dynamic programming." [Aro97].

13.2 Definitions and Notations

In the following we use the convention that rectangles are axis-aligned. The *size* of a rectangle is the length of its longer edge. For a given set of n nodes the *bounding box* is the smallest square enclosing them. We will always denote the size of any bounding box by L and without loss of generality let $L \in \mathbb{N}$.

In the following we assume that there is a grid of granularity 1, such that all nodes lie on grid points. Moreover, we assume that the bounding box has size $L = nc$. This is no restriction, since any instance of the problem can be transformed into one that satisfies these conditions with small extra cost: Place a grid of granularity $L/(nc)$ over the instance and move every node to the nearest grid point. (It is possible that two nodes now lie on the same grid point.) If we find a solution for this new instance, then we also have a solution for the original instance. Let p be a salesman path for the new instance. We can construct a salesman path p' for the original instance as follows: For every node add two edges, from the node to the nearest grid point and back. The total length of these additional edges is $\leqslant 2n \cdot \sqrt{2}/2 \cdot L/(nc)$. As the length **Opt** of the optimal tour is at least L, this is $\leqslant \sqrt{2}\,\mathbf{Opt}/c = \mathcal{O}(\mathbf{Opt}/c)$. This additional cost is acceptable, since we are only interested in $(1+1/c)$-approximations. So far the minimal internode distance in the new instance is $L/(nc)$. Now we scale all distances by $L/(nc)$ and obtain a minimal internode distance of 1. The new size of the bounding box is now nc. The transformation only requires $\mathcal{O}(n \log n)$ time.

The definitions of a *dissection* and a *quadtree* are essential to this PTAS. Both are recursive partitionings of the bounding box into smaller squares. To get a dissection of the bounding box we partition any square into four squares of equal size until each square has length at most 1. These four squares are called the children of the bigger one. Similarly we define a quadtree. Also this time each square is divided into four equal squares, but we already stop when a square contains at most one node.

Instead of "normal" dissections or quadtrees we will consider shifted ones. For $a, b \in \mathbb{N}$ the (a,b)*-shift* of the dissection is obtained by shifting all x- and y-coordinates of all lines by a and b respectively, and then reducing them modulo L. This can lead to cases where previous rectangles are now wrapped around. Consider a simple dissection of depth one. There is only the bounding box and its four children. If it is shifted by shifts $a, b \leqslant L/2$, then three of the original children will be wrapped around. The only child that is not wrapped around is the child-square that used to reside in the lower left corner; it is now approximately in the center of the bounding box.

Let there be a shifted dissection. To construct the corresponding *quadtree with shift* (a,b) we cut off the partitioning at each square that contains at most one node.

In a dissection or in a quadtree a child-square has exactly half the size of its parent-square (the size is defined as the length of the longer edge). The size of the bounding box is nc and c is a constant, thus at a depth of $\mathcal{O}(\log n)$ the rectangles in the quadtree have size 1. As the minimal internode distance is 1, the construction stops there, because there is at most one node in the square. Thus any quadtree – shifted or not – has depth $\mathcal{O}(\log n)$ and consists of $\mathcal{O}(\text{number of nodes} \cdot \text{depth}) = \mathcal{O}(n \log n)$ squares. Hence the quadtree can be obtained in time $\mathcal{O}(n \log^2 n)$.

Finally we define the concept of a *light salesman tour*, which is crucial for the following.

Definition 13.1. *Let* $m, t \in \mathbb{N}$. *For each square we introduce an additional node at each corner and* m *further equally-spaced nodes on each of its four edges. These points are called portals.*

We say that a salesman tour is (m,t)*-light with respect to the shifted dissection if it crosses each edge of each square in the dissection at most* t *times and all crossings go through one of the portals. A tour can use a portal more than once.*

To find a nearly optimal salesman tour by dynamic programming means to find an approximate tour in each subsquare and to put these tours together. Therefore we have to consider all possibilities where a segment of a tour can enter a square. In a (m,t)-light salesman tour these points are limited by the discrete set of portals. Since the distances are given by the Euclidean norm, the triangle-inequality holds, which means that the direct way between two nodes is always at most as long as the way through an additional point. Hence removing these additional points (portals) never increases the cost of the tour.

13.3 The Structure Theorem and Its Proof

The algorithm, which we will describe in Section 13.4, mainly relies on the following Structure Theorem:

Theorem 13.2. *(Structure Theorem) Consider an Euclidean* TSP *instance with bounding box of size L and the corresponding dissection with shift (a, b) where $a, b \in \mathbb{N}$ with $0 \leqslant a, b \leqslant L$ are picked randomly (every choice of a, b has the same probability).*

Then with probability at least $1/2$ this dissection has a salesman tour of cost at most $(1 + 1/c)$ **Opt** *that is (m, t)-light where $m = \mathcal{O}(c \ \log L)$, $t = \mathcal{O}(c)$ and $m \geqslant t \log L$.*

In the proof of this theorem we assume that we have an optimal tour p and randomly chosen shifts $a, b \in \mathbb{N}$. Then we modify this tour until we get a new approximate salesman tour – probably through additional nodes – in the dissection with shift (a, b) which is (m, t)-light. The following lemma helps us to reduce the number of crossings between the salesman tour and the edges of each square.

Lemma 13.3. *(Patching Lemma) Let p be a closed path. Consider a line segment S of length l and assume that p crosses S at least three times. Then there exist line segments on S of total length less than or equal to $6l$ such that the addition of these segments to p changes p into a closed path p' of length $|p| + 6l$ that crosses S at most twice.*

Proof. We assume that p (of length $|p|$) is a closed path that crosses S $2k + 2$ ($2k + 1$, respectively) times. Let $x_1, \ldots, x_{2k+2}$ ($x_1, \ldots, x_{2k+1}$, resp.) be the corresponding crossing points. For $i = 1, \ldots, 2k$ break the path p at the points x_i. This implies that p breaks into $2k$ subpaths $P_1, \ldots, P_{2k}$. Also we double all these points x_i, $i = 1, \ldots, 2k$ and call these copies x'_i. Now we are adding the line segments of a minimal cost salesman tour through $x_1, \ldots, x_{2k}$, a minimal cost salesman tour through $x'_1, \ldots, x'_{2k}$ and also the line segments of minimal cost perfect matchings among $x_1, \ldots, x_{2k}$ and among $x'_1, \ldots, x'_{2k}$. All added line segments are lying on S which yields an Eulerian graph and hence there is a tour p' through $x_1, \ldots, x_{2k+2}, x'_1, \ldots, x'_{2k+2}$ using these line segments and the paths $P_1, \ldots, P_{2k}$. (An Eulerian graph is a connected graph which has a tour through all edges. These graphs are characterized by the fact that they are connected and that each point has even degree.) The new path p' is now using line-segments of S, nevertheless there are at most two crossing points. Also the added segments are of length at most $6l$ and hence $|p'| \leqslant |p| + 6l$. ■

There is a longer proof [Aro97] that shows that one can also find such a tour p' where the length is only increased by $3l$, and probably even this can be improved. In the proof of the Structure Theorem we just need the fact that p only increases by a constant factor of the length of S.

The next lemma shows an upper bound on how often a salesman tour crosses grid lines. It also will be used in the proof of the Structure Theorem. We defined that all nodes lie on grid points of a grid of granularity 1 (see Section 13.2). So the minimal internode distance is at least 1 and thus all line segments in the salesman path have a length of at least 1.

Lemma 13.4. *Let p be a path consisting of segments of length at least 1 and $t(p,l)$ the number of times that p crosses a horizontal or vertical line l of the grid, then*

$$\sum_{l:vertical} t(p,l) \; + \sum_{l:horizontal} t(p,l) \;\leqslant\; \left(\sqrt{2}+2\right) length(p).$$

Proof. Regard an edge of p of length s and let x and y be the lengths of the horizontal and vertical projections of this edge, thus $x^2 + y^2 = s^2$. The edge contributes at most $(x+1)+(y+1)$ to the left hand side. As $s \geqslant 1$ we have

$$x + y + 2 \;\leqslant\; \sqrt{2(x^2+y^2)} + 2 \;=\; \sqrt{2}s + 2 \;\leqslant\; \left(\sqrt{2} + \frac{2}{s}\right) s \leqslant (\sqrt{2}+2)s$$

Taking the sum over all edges of p yields the result. ■

Definition 13.5. *The squares in a (shifted) dissection can be assigned a level. The bounding box is at level* 0*, its four children at level* 1*, and so on. Consider a grid line through the bounding box that contains line-segments of the dissection. If such a line l contains an edge of a square of level i in the dissection we say that l also has level i. For each line l there is a minimal $i \in \mathbb{N}$ such that l has level i. We call i the* minimal level *of l. (Note that l also has all levels $j \geqslant i$.)*

Proof of the Structure Theorem. Assume that there is an optimal tour p and shifts $a, b \in \mathbb{N}$ which are choosen randomly. Now we consider the dissection with shift (a,b) and show how the tour is modified into an approximate salesman tour which is (m,t)-light. In particular we determine how much this modification increases the cost.

As defined in Section 13.2 all nodes lie on grid points of a grid of granularity 1. Without loss of generality we assume that L is a power of 2. This ensures that all lines used in the dissection are grid lines. And since the shift values a and b are integers, the lines of the shifted dissection are still grid lines.

In the following we only consider the vertical lines of the dissection, for horizontal ones the modification works analogously. Let i be the minimal level of such a vertical line l. By definition the property to be (m,t)-light consists of two components.

(1) The salesman tour crosses each edge of a square with level i at most t times.

To meet this first condition we use the Patching Lemma. We charge the cost increases to the line l to which it is applied.

The edges on the vertical line l correspond to the line segments which lie between the y-coordinates $b+r\cdot\frac{L}{2^i} \bmod L$ and $b+(r+1)\cdot\frac{L}{2^i} \bmod L$ $(r=0,1,2,\ldots,2^i-1)$. For each $j \geqslant i$ let $l'_{j,r}$ denote the line segment between the y-coordinates $b+r\cdot\frac{L}{2^j}$ mod L and $b+(r+1)\cdot\frac{L}{2^j}$ mod L. In order to minimize the cost of the reducing of crossings we do not apply the lemma directly, but do the patching bottom-up for every j with $\log L \geqslant j \geqslant i$. Therefore we use the following procedure $CHANGE(l,i,b)$.

<u>$CHANGE(l,i,b)$</u>
for $j=\log L$ **down to** i **do**
  for $r=0,1,\ldots,2^j-1$ **do**
    if $l'_{j,r}$ is crossed by the current tour more often than t times, then reduce the number of crossings by applying the Patching Lemma
  end
end

Obviously, for every u, all modifications for $j=\log L,\ldots,u+1$ influence the modification for $j=u$. In each application of the Patching Lemma to a vertical line l of the dissection we add some vertical line segments of l to the tour p which implies that patching a vertical line l possibly increases the number of times the tour crosses a horizontal line h of the dissection. Fortunately, we can assume that the increase in the number of crossings at h due to the patching on l is at most two. Otherwise we would again apply the Patching Lemma to reduce them to two. This additional application does not increase the cost, because all added segments lie on l and have no horizontal separation. These two extra crossings have to be taken into account for both horizontal and vertical lines. So altogether we need 4 extra crossings.

It may happen that a line segment is wrapped around and the Patching Lemma has to be applied separately for each part. In this case we can just ensure that the number of crossings after this procedure is four, not two as in the normal case.

Now we are considering how much this procedure increases the cost of the tour. The additional cost is influenced by the length of the bounding box L, the length of the new line segments introduced by the Patching Lemma and by the number of segments to which the Patching Lemma has to be applied. For $j \geqslant i$ let $c_{l,j}(b)$ denote the number of segments to which we apply the Patching Lemma in the j-th iteration. This implies that the additional cost due to $CHANGE(l, i, b)$ is less than or equal to

$$\sum_{j \geqslant i} c_{l,j}(b) \cdot 6 \cdot \frac{L}{2^j}.$$

Whenever the Patching Lemma is applied at least $t+1$ crossings are replaced by at most 4. Thus every application of the Patching Lemma decreases the number of crossings by at least $t-3$.

Since the original tour p crosses l only $t(p,l)$ times, we conclude $\sum_{j \geqslant 1} c_{l,j}(b) \leqslant \frac{t(p,l)}{t-3}$ for each $b \in \mathbb{N}$.

The shifts $a, b \in \mathbb{N}$ are chosen randomly. Hence the probability that a line l has level i is $2^i/L$ for all $i \leqslant \log L$. Thus for every vertical line l the expected cost which is charged to the line l is

$$\begin{aligned}
&= \sum_{i \geqslant 1} \frac{2^i}{L} \cdot \text{cost increase due to } CHANGE(l, i, b) \\
&\leqslant \sum_{i \geqslant 1} \frac{2^i}{L} \cdot \sum_{j \geqslant i} c_{l,j}(b) \cdot 6 \cdot \frac{L}{2^j} \\
&= 6 \cdot \sum_{j \geqslant 1} \frac{c_{l,j}(b)}{2^j} \cdot \sum_{i \leqslant j} 2^i \\
&\leqslant 6 \cdot \sum_{j \geqslant 1} 2 \cdot c_{l,j}(b) \\
&\leqslant \frac{12\, t(p,l)}{t-3}.
\end{aligned}$$

Since l is vertical this inequality holds for all $b \in \{0, 1, \ldots, L-1\}$.

(2) All these crossings have to be at portals.

This time we simply move each crossing to the nearest portal which implies the following cost:

Again consider a line l of the dissection with minimal level i. There are at most $t(p,l)$ crossings which have to be moved and for each of it the cost to move it to the nearest portal is at most $\frac{L}{2^i m}$. As we assumed $m \geqslant t \log L$, the cost for this step is upperbounded by

$$\sum_{i=1}^{\log L} \frac{2^i}{L} \cdot t(p,l) \cdot \frac{L}{2^i m} = \frac{t(p,l)\log L}{m} \leqslant t(p,l)/t.$$

Adding the cost in (1) and (2) we find that the expected cost for modifying an optimal salesman tour p into a salesman tour which is (m,t)-light at line l is at most:

$$\frac{12 \cdot t(p,l)}{t-3} + \frac{t(p,l)}{t} = \frac{t(p,l)}{t}(13 + \frac{36}{t-3}).$$

For sufficiently large t ($t \geqslant 12$) this is upperbounded by

$$\frac{17\, t(p,l)}{t}.$$

By taking the sum over all lines of the dissection and using Lemma 13.4 (remember that all lines of the dissection are grid lines) we get

$$\sum_{l:vertical} \frac{17t(p,l)}{t} + \sum_{l:horizontal} \frac{17t(p,l)}{t} \leqslant \left(\sqrt{2}+2\right) \frac{17\,\mathbf{Opt}}{t}.$$

Thus when $t \geqslant 34c\left(\sqrt{2}+2\right)$ the tour cost increases by at most $\mathbf{Opt}/(2c)$. It follows from the Markov inequality that with probability $\geqslant 1/2$ the increase is not more than $\mathbf{Opt}/c$ and hence the cost of the best (m,t)-light salesman tour corresponding to the shifted dissection is at most $(1+1/c)\,\mathbf{Opt}$.

$t \geqslant 34c\left(\sqrt{2}+2\right) \approx 116.08 \cdot c$ implies that $t = 117c$ suffices. However, in step (1) we saw that we sometimes need 4 extra crossings. Since $c \geqslant 1$, a value of $t = 121c$ suffices. This satisfies our previous assumption that $t \geqslant 12$.

Arora [Aro97] showed that the Patching Lemma also holds for a constant 3 instead of 6. By using this lower constant and several other optimizations he achieves smaller values for t. So far the best known is about $t = 10c$. ∎

13.4 The Algorithm

Our aim is to find an approximate salesman tour through n given nodes in $\mathcal{O}(n \log^{\mathcal{O}(c)} n)$-time. In the following we will simply speak of a solution when we mean an approximate solution in contrast to an optimal one.

Like in Section 13.2 we assume that all nodes are on grid points of a grid of granularity 1 and that the size L of the bounding box is $L = nc$.

Let $0 \leqslant a, b \leqslant L$ with $a, b \in \mathbb{N}$ be two randomly chosen shifts. Using dynamic programming the algorithm computes the best (m, t)-light salesman tour with respect to the corresponding quadtree with shift (a, b). (Remember that constructing this shifted quadtree needs at most $\mathcal{O}(n \log^2 n)$ time.) The Structure Theorem guarantees that with probability at least $1/2$ this salesman tour has a cost of at most $(1 + 1/c)\,\mathbf{Opt}$.

The algorithm has to find a solution for all nontrivial squares. As the depth of the quadtree is $\mathcal{O}(\log n)$ these are $\mathcal{O}(\text{number of squares with a node} \cdot \textit{depth}) = \mathcal{O}(n \log n)$ squares. For each side of each square there are at most $\mathcal{O}(m^t)$ possibilities where the salesman tour could cross the boundary of the square (these are $\mathcal{O}(m^{4t})$ per square). Each considered tour enters and leaves the square at one of these portals, altogether there are at most $(4t)!$ different pairs of crossings.

Each of the resulting subproblems is specified by the following parameters:

(X) again $\mathcal{O}(m^{4t})$ possibilities to choose at most t portals on the edges of the subsquare.

(Y) $(4t)^{4t} \cdot (4t)!$ possible orderings of these portals.

Recursively this procedure solves the subproblems and keeps the choice that leads to the lowest cost in (X) and (Y). The algorithm uses a lookup table to store these solutions, thus it never solves the same subproblem twice. Hence the running time is bounded by $\mathcal{O}(T \cdot (m^{4t})^2 \cdot (4t)^{4t} \cdot (4t)!^2)$ where T denotes the number of these squares. Since $T = \mathcal{O}(n \log n)$, $m = \mathcal{O}(c \log L)$ and $L = nc$ we obtain a running time of $\mathcal{O}(n \log n \cdot (c\ \log(nc))^{\mathcal{O}(c)}) = \mathcal{O}(n (\log n)^{\mathcal{O}(c)})$.

13.5 Applications to Some Related Problems

In this section we want to give a short overview to which optimization problems this algorithm or a similar one can be applied.

A possible generalization is to regard spaces with a higher dimension. As there is a R^d version of the Patching Lemma [Aro97], the Structure Theorem and the algorithm can be generalized to higher dimensions. For any fixed dimension d this modified algorithm runs in polynomial time. In time $\mathcal{O}(n (\log n)^{(\mathcal{O}(dc))^{d-1}})$ it computes an (m, t)-light approximate salesman tour where $m = \mathcal{O}((\sqrt{d}\, c \log n)^{d-1})$ and $t = \mathcal{O}((\sqrt{d}\, c)^{d-1})$.

The Patching Lemma and Lemma 13.4 hold independently of the considered problem if these are similar problems like k-TSP, k-MST and so on. And so does the basic structure of the Structure Theorem. For Euclidean k-TSP – the more general version of Euclidean TSP – there are two main changes. First $L < \mathbf{Opt}$ can happen and hence it is more complicated to fulfill the assumption that there is a grid of granularity one such that all nodes lie on grid points (section 13.2). To construct an approximation P to **Opt** with $P \leqslant \mathcal{O}(\mathbf{Opt} \cdot n)$ helps [Aro97]. Additionally we are confronted with the problem which k nodes out of the given n the salesman tour should use. Nevertheless the algorithm works similar and the running time only grows by a factor of $\mathcal{O}(k)$. The same happens for the Euclidean case of k-MST with $k < n$ where k-MST is defined as follows:

k-MINIMAL SPANNING TREE PROBLEM (k-MST)

Instance: A graph $G = (V, E)$ with edge weights defined by $c : E \longrightarrow R$ and an integer $k < n$.

Problem: Find a spanning tree on k nodes that has a minimal sum of weights.

While this problem is $\mathcal{NP}$-hard, there are polynomial time solutions in the special case of $k = n$.

If we are not searching for a minimal tour but for a minimal spanning tree through all nodes of a graph, then we can possibly minimize the lengths of the tree by introducing additional intermediate points. These points are called *Steiner points* and the connected tree is a *Steiner tree*.

The algorithm also has to be modified a little bit to find such a minimal Steiner tree. This time there possibly occur Steiner points in trivial squares. (Trivial squares are squares that do not contain any node of the input.) But since we are interested in computing an (m, t)-light Steiner tree the number of these points in any trivial square is bounded by $\mathcal{O}(t)$. If we use a regular grid of $\mathcal{O}(t \log n)$ vertical and $\mathcal{O}(t \log n)$ horizontal lines in such a square then we get $\mathcal{O}((t \log n)^2)$ grid points. Choosing the additional Steiner points out of this set neither increases the cost of the Steiner tree too much, nor are there too many possibilities ($\mathcal{O}((t \log n)^{\mathcal{O}(t)})$). Hence the running time does not change.

13.6 The Earlier PTAS

In Arora's PTAS the Euclidean space is partitioned recursively into smaller parts such that all have a common simple structure. The important point is that there is always a near optimal solution to the optimization problem that "agrees closely" with the partitioning. This means that as the depth of the tiling increases there is still a solution that "agrees" with this tiling and whose quality differs from the previous best approximation only by a small constant factor.

Both earlier PTAS – the one due to Arora [Aro96] and the one due to Mitchell [Mit96] – are based on the same concept, except that these algorithms are not randomized. Arora used an extremely simple partitioning technique called 1/3 : 2/3-tiling, while [MBCV97] Mitchell, Blum, Chalasani and Vempala introduced a new partitioning technique (guillotine subdivisions) which leads to an algorithm that achieves a constant factor approximation for the Euclidean k-MINIMAL SPANNING TREE PROBLEM. By generalizing this idea to m-guillotine subdivisions Mitchell [Mit96] obtained PTAS for k-MST, $k \leqslant n$ and TSP.

An m-guillotine subdivision is a recursively defined partition of the bounding box into smaller instances. In these algorithms there are, at each step, $\mathcal{O}(n)$ different possibilities to partition a problem into two subproblems. Also the number of so called boundary information is limited. All these possibilities are considered one by one which implies a running time of $\mathcal{O}(n^{\mathcal{O}(c)})$.

In contrast to this Arora now considers exactly one partition and looks at all levels at once. There are $\log n$ levels, hence this gives a running time of $\log^{\mathcal{O}(c)} n$ for each subproblem. Since the number of these subproblems is polynomial in n, the whole running time is bounded by $\mathcal{O}(n \log^{\mathcal{O}(c)} n)$.

Finally, the question occurs if these algorithms are applicable in practice. Unfortunately this is not yet the case. Until Arora and Mitchell showed that the Euclidean TSP belongs to the $\mathcal{NP}$-hard problems which have a PTAS, Christofides' approximation algorithm was the best known. Nevertheless, these PTAS are mainly of theoretical interest. In polynomial time the algorithm of Christofides [Chr76] computes a tour that has at most $3/2$ times the optimal cost. And still in this special case this algorithm is more practical. While the Christofides approximation runs in time $\mathcal{O}(n^{2.5})$, Arora's current version is much slower. So far in the best implementation the $\text{poly}(\log n)$ term still grows like $(\log n)^{40c}$ [Aro97].

Bibliography

[ABI86] N. Alon, L. Babai, and A. Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of Algorithms*, 7:567–583, 1986.

[AFK96] S. Arora, A. Frieze, and H. Kaplan. A new rounding procedure for the assignment problem with applications to dense graph arrangement problems. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 21–30, 1996.

[AHOH96] T. Asano, K. Hori, T. Ono, and T. Hirata. Approximation algorithms for MaxSat: Semidefinite programming and network flows approach. Technical report, 1996.

[AKK95a] S. Arora, D. Karger, and M. Karpinski. Polynomial time approximation schemes for dense instances of $\mathcal{NP}$-hard problems. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, pages 284–293, 1995.

[AKK95b] S. Arora, D. Karger, and M. Karpinski. Polynomial time approximation schemes for dense instances of $\mathcal{NP}$-hard problems. Unpublished manuscript, November 1995.

[AL96] S. Arora and C. Lund. Hardness of approximations. In D.S. Hochbaum, editor, *Approximation Algorithms for $\mathcal{NP}$-hard Problems*, pages 399–446. PWS Publishing Company, 1996.

[Ali95] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1):13–51, 1995.

[ALM$^+$92] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science*, pages 14–23, 1992.

[AOH96] T. Asano, T. Ono, and T. Hirata. Approximation algorithms for the maximum satisfiability problem. *Nordic Journal of Computing*, 3:388–404, 1996.

[Aro94] S. Arora. *Probabilistic checking of proofs and hardness of approximation problems.* PhD thesis, Department of Computer Science, Berkeley, 1994.

[Aro96] S. Arora. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 2–11, 1996.

[Aro97] S. Arora. Nearly linear time approximation schemes for Euclidean TSP and other geometric problems. In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science*, 1997. To appear.

[AS92] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of $\mathcal{NP}$. In *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science*, pages 2–13, 1992.

[Asa97] T. Asano. Approximation algorithms for MaxSat: Yannakakis vs. Goemans-Williamson. In *Proceedings of the 5th Israel Symposium on the Theory of Computing and Systems*, 1997. To appear.

[Bak94] B.S. Baker. Approximation algorithms for $\mathcal{NP}$-complete problems on planar graphs. *Journal of the ACM*, 41(1):153–180, 1994.

[BC93] D.P. Bovet and P. Crescenzi. *Introduction to the Theory of Complexity.* Prentice-Hall, 1993.

[BCW80] M. Blum, A.K. Chandra, and M.W. Wegman. Equivalence of free Boolean graphs can be decided probabilistically in polynomial time. *Information Processing Letters*, 10:80–82, 1980.

[Bel96] M. Bellare. Proof checking and approximation: Towards tight results. *Complexity Theory Column of Sigact News*, 27, 1996.

[BFL91] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991.

[BFLS91] L. Babai, L. Fortnow, L. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 21–31, 1991.

[BGG93] M. Bellare, O. Goldreich, and S. Goldwasser. Randomness in interactive proofs. *Computational Complexity*, 3:319–354, 1993.

[BGKW88] M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 113–131, 1988.

[BGLR93] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient probabilistically checkable proofs and applications to approximation. In

Proceedings of the 25th Annual ACM Symposium on Theory of Computing, 1993.

[BGS95] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs and non-approximability. Technical Report TR95-024, Electronic Colloquium on Computational Complexity, 1995.

[BH92] R. Boppana and M.M. Halldórsson. Approximating maximum independent set by excluding subgraphs. *BIT*, 32:180–196, 1992.

[BM95] R. Bacik and S. Mahajan. Semidefinite programming and its applications to $\mathcal{NP}$ problems. In *Proceedings of the 1st Computing and Combinatorics Conference (COCOON)*. Lecture Notes in Computer Science 959, Springer Verlag, 1995.

[BR94] M. Bellare and J. Rompel. Randomness-efficient oblivious sampling. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 276–287, 1994.

[BS94] M. Bellare and M. Sudan. Improved non-approximability results. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 184–193, 1994.

[BW86] E. Berlekamp and L. Welch. Error correction of algebraic block codes. US Patent Number 4,633,470, 1986.

[Chr76] N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, 1976.

[CK81] I. Csiszar and J. Korner. *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Academic Press, 1981.

[CKST95] P. Crescenzi, V. Kann, R. Silvestri, and L. Trevisan. Structure in approximation classes. In *Proceedings of the 1st Computing and Combinatorics Conference (COCOON)*, pages 539–548. Lecture Notes in Computer Science 959, Springer Verlag, 1995.

[Coo71] S.A. Cook. The complexity of theorem proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[CST96] P. Crescenzi, R. Silvestri, and L. Trevisan. To weight or not to weight: Where is the question? In *Proceedings of the 4th Israel Symposium on the Theory of Computing and Systems*, pages 68–77, 1996.

[Dan63] G.B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

[DF85] M.E. Dyer and A.M. Frieze. A simple heuristic for the p-center problem. *Operations Research Letters*, 3:285–288, 1985.

[Fag74] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In *SIAM-AMS Proceedings*, pages 43–73, 1974.

[Fei91] U. Feige. On the success probability of the two provers in one-round proof systems. In *Proceedings of the 6th IEEE Symposium on Structure in Complexity Theory*, pages 116–123, 1991.

[Fei95] U. Feige. Error reduction by parallel repetition – the state of the art. Technical Report CS95-32, WISDOM Technical reports in Computer Science, 1995.

[Fei96] U. Feige. A threshold of $\ln n$ for approximating set cover. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 314–318, 1996.

[Fer96] W. Fernandez de la Vega. MaxCut has a randomized approximation scheme in dense graphs. *Random Structures and Algorithms*, 8(3):187–198, 1996.

[FG95] U. Feige and M. Goemans. Approximating the value of two prover proof systems, with applications to Max2Sat and MaxDiCut. In *Proceedings of the 3rd Israel Symposium on the Theory of Computing and Systems*, pages 182–189, 1995.

[FGL+91] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost $\mathcal{NP}$-complete. In *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science*, pages 2–12, 1991.

[FK91] H. Furstenberg and Y. Katznelson. A density version of the Hales-Jewett theorem. *Journal d'Analyse Mathematique*, 57:64–119, 1991.

[FK94] U. Feige and J. Kilian. Two-prover protocols – low error at affordable rates. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 172–183, 1994.

[FL92] U. Feige and L. Lovász. Two-prover one-round proof systems: Their power and their problems. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 733–744, 1992.

[Fri91] J. Friedman. On the second eigenvalue and random walks in random d-regular graphs. *Combinatorica*, 11(4):331–362, 1991.

[FRS88] L. Fortnow, J. Rompel, and M. Sipser. On the power of multi-prover interactive protocols. In *Proceedings of the 3rd IEEE Symposium on Structure in Complexity Theory*, pages 156–161, 1988.

[FRS90] L. Fortnow, J. Rompel, and M. Sipser. Errata for on the power of multi-prover interactive protocols. In *Proceedings of the 5th IEEE Symposium on Structure in Complexity Theory*, pages 318–319, 1990.

[FV96] U. Feige and O. Verbitsky. Error reduction by parallel repetition – A negative result. In *Proceedings of the 11th Annual IEEE Conference on Computational Complexity Theory*, 1996.

[FW91] M. Formann and F. Wagner. A packing problem with applications to lettering of maps. In *Proceedings of the 7th Annual ACM Symposium on Computational Geometry*, pages 281–288, 1991.

[GG81] O. Gabber and Z. Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22:407–420, 1981.

[GGJ76] M.R. Garey, R.L. Graham, and D.S. Johnson. Some $\mathcal{NP}$-complete geometric problems. In *Proceedings of the 8th Annual ACM Symposium on Theory of Computing*, pages 10–22, 1976.

[Gil93] D. Gillman. A Chernoff bound for random walks on expanders. In *Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science*, pages 680–691, 1993.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability. A Guide to the Theory of $\mathcal{NP}$-Completeness.* W.H. Freeman and Company, 1979.

[Goe97] M.X. Goemans. Semidefinite programming in combinatorial optimization. In *Proceedings of the 16th International Symposium on Mathematical Programming*, 1997.

[Gol95] O. Goldreich. Probabilistic proof systems (survey). Technical report, Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, 1995.

[Gra90] R.M. Gray. *Entropy and Information Theory.* Springer-Verlag, 1990.

[GS92] P. Gemmell and M. Sudan. Highly resilient correctors for polynomials. *Information Processing Letters*, 43:169–174, 1992.

[GvL86] G.H. Golub and C.F. van Loan. *Matrix Computations.* North Oxford Academic, 1986.

[GW94a] M.X. Goemans and D.P. Williamson. .878 approximation algorithms for MaxCut and Max2Sat. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 422–431, 1994.

[GW94b] M.X. Goemans and D.P. Williamson. New 3/4-approximation algorithms for the maximum satisfiability problem. *SIAM Journal on Discrete Mathematics*, 7:656–666, 1994.

[GW95] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[Hås96a] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 627–636, 1996.

[Hås96b] J. Håstad. Testing of the long code and hardness for clique. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 11–19, 1996.

[Hås97a] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. Technical Report TR97-038, Electronic Colloquium on Computational Complexity, 1997. An earlier version was presented in *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science* 1996, pp. 627-636.

[Hås97b] J. Håstad. Some optimal inapproximability results. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 1–10, 1997. Also appeared as Technical Report TR97-037, Electronic Colloquium on Computational Complexity.

[HL96] T. Hofmeister and H. Lefmann. A combinatorial design approach to MaxCut. *Random Structures and Algorithms*, 9(1-2):163–175, 1996.

[HN79] W. Hsu and G. Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1:209–216, 1979.

[Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[HPS94] S. Hougardy, H.J. Prömel, and A. Steger. Probabilistically checkable proofs and their consequences for approximation algorithms. *Discrete Mathematics*, 136:175–223, 1994.

[HS85] D.S. Hochbaum and D. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10:180–184, 1985.

[IK75] O.H. Ibarra and C.E. Kim. Fast approximation algorithms for the knapsack and sum of subsets problems. *Journal of the ACM*, 22:463–468, 1975.

[Joh74] D.S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

[Kar72] R.M. Karp. Reducibility among combinatorial problems. In R.E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[Kar84] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

[Kar91] R.M. Karp. An introduction to randomized algorithms. *Discrete Applied Mathematics*, 34:165–201, 1991.

[Kar96] H.J. Karloff. How good is the Goemans-Williamson MAXCUT algorithm? In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 427–434, 1996.

[Kha79] L.G. Khachian. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979.

[KK82] N. Karmarkar and R.M. Karp. An efficient approximation scheme for the one-dimensional bin packing problem. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 312–320, 1982.

[KL96] P. Klein and H.-I. Lu. Efficient approximation algorithms for semidefinite programs arising from MAXCUT and COLORING. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 338–347, 1996.

[KMS94] D. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semidefinite programming. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 2–13, 1994.

[KMSV94] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani. On syntactic versus computational views of approximability. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 819–830, 1994.

[Knu81] D.E. Knuth. *The Art of Computer Programming, Vol. 2, Seminumerical Algorithms.* Addison-Wesley, 2nd edition, 1981.

[KR90] R.M. Karp and V. Ramachandran. Parallel algorithms for shared-memory machines. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Vol. A, Algorithms and Complexity*, pages 869–941. Elsevier, 1990.

[KZ97] M. Karpinski and A. Zelikovsky. Approximating dense cases of covering problems. Technical Report TR97-004, Electronic Colloquium on Computational Complexity, 1997.

[LFKN92] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992.

[LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

[LN87] R. Lidl and H. Niederreiter. *Finite Fields.* Cambridge University Press, 1987.

[Lov75] L. Lovász. On the ratio of the optimal integral and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.

[LPS86] A. Lubotzky, R. Phillips, and P. Sarnak. Explicit expanders and the Ramanujan conjectures. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, 1986.

[Lub85] M. Luby. A simple parallel algorithm for the maximal independent set problem. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 1–10, 1985.

[Lub86] M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM Journal on Computing*, 15(4):1036–1053, 1986.

[LY93] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 286–293, 1993. Also appears in: *Journal of the ACM* 41: 960-981, 1994.

[Mar75] G.A. Margulis. Explicit construction of concentrators. *Problems of Information Transmission*, 10:325–332, 1975.

[MBCV97] J.S.B. Mitchell, A. Blum, P. Chalasani, and S. Vempala. A constant-factor approximation algorithm for the geometric k-MST problem in the plane. *SIAM Journal on Computing*, pages 402–408, 1997. To appear. Journal version of: J.S.B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple new method for the geometric k-MST problem. In *Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms.* 1996.

[MBG+93] A. Menezes, I. Blake, X. Gao, R. Mullin, S. Vanstone, and T. Yaghoobian. *Applications of Finite Fields.* Kluwer Academic Publishers, 1993.

[Mit96] J.S.B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: Part II – a simple polynomial-time approximation scheme for geometric k-MST, TSP, and related problems. Technical report, Department of Applied Mathematics and Statistics, Stony Brook, 1996.

[MNR96] R. Motwani, J. Naor, and P. Raghavan. Randomized approximation algorithms in combinatorial optimization. In D.S. Hochbaum, editor, *Approximation Algorithms for $\mathcal{NP}$-hard Problems*, pages 447–481. PWS Publishing Company, 1996.

[MR95a] S. Mahajan and H. Ramesh. Derandomizing semidefinite programming based approximation algorithms. In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*, pages 162–169, 1995.

[MR95b] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

[MS77] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes.* North-Holland, 1977.

[Nis96] N. Nisan. Extracting randomness: How and why – A survey. In *Proceedings of the 11th IEEE Symposium on Structure in Complexity Theory*, 1996.

[NSS95] M. Naor, L. Schulman, and A. Srinivasan. Splitters and near-optimal derandomization. In *Proceedings of the 36th Annual IEEE Symposium of Foundations of Computer Science*, pages 182–191, 1995.

[Pap77] C.H. Papadimitriou. The Euclidean traveling salesman problem is $\mathcal{NP}$-complete. *Theoretical Computer Science*, 4:237–244, 1977.

[Pap94] C.H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[PM81] A. Paz and S. Moran. Non deterministic polynomial optimization problems and their approximation. *Theoretical Computer Science*, 15:251–277, 1981.

[PS82] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, 1982.

[PS94] A. Polishchuk and D.A. Spielman. Nearly linear-size holographic proofs. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 194–203, 1994.

[PT95] S. Poljak and Z. Tuza. Maximum cuts and large bipartite subgraphs. In *Combinatorial Optimization*, volume 20 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 181–244, 1995.

[PY91] C.H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.

[Rag88] P. Raghavan. Probabilistic construction of deterministic algorithms: approximating packing integer programs. *Journal of Computer and System Sciences*, 37:130–143, 1988.

[Raz95] R. Raz. A parallel repetition theorem. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, pages 447–456, 1995.

[Raz97] R. Raz. A parallel repetition theorem (full version). Unpublished manuscript, 1997.

[RS92] R. Rubinfeld and M. Sudan. Self-testing polynomial functions efficiently and over rational domains. In *Proceedings of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 23–32, 1992.

[RT87] P. Raghavan and C.D. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7:365–374, 1987.

[Spe94] J. Spencer. Randomization, derandomization and antirandomization: three games. *Theoretical Computer Science*, 131:415–429, 1994.

[SS77] R. Solovay and V. Strassen. A fast Monte-Carlo test for primality. *SIAM Journal on Computing*, 6(1):84–85, 1977. An erratum appeared in *SIAM Journal on Computing* 7(1): 118, 1978.

[Sud92] M. Sudan. *Efficient checking of polynomials and proofs and the hardness of approximation problems.* PhD thesis, Department of Computer Science, Berkeley, 1992.

[SW97] M. Sudan and D.P. Williamson. Some notes on Håstad's proof. Unpublished manuscript, 1997.

[Tre97] L. Trevisan. *Reductions and (Non)-Approximability.* PhD thesis, Computer Science Department, University of Rome"La Sapienza", 1997.

[TSSW96] L. Trevisan, G.B. Sorkin, M. Sudan, and D.P. Williamson. Gadgets, approximation, and linear programming. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 617–626, 1996.

[VB96] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[Ver94] O. Verbitsky. Towards the parallel repetition conjecture. In *Proceedings of the 9th IEEE Symposium on Structure in Complexity Theory*, pages 304–307, 1994.

[Ver96] O. Verbitsky. Towards the parallel repetition conjecture. *Theoretical Computer Science*, 157(2):277–282, 1996.

[Weg93] I. Wegener. *Theoretische Informatik: eine algorithmenorientierte Einführung.* B.G. Teubner, 1993.

[Yan92] M. Yannakakis. On the approximation of maximum satisfiability. In *Proceedings of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1–9, 1992.

[Ye90] Y. Ye. An $\mathcal{O}(n^3L)$ potential reduction algorithm for linear programming. In *Contemporary Mathematics*, volume 114, pages 91–107, 1990.

[Zuc93] D. Zuckerman. $\mathcal{NP}$-complete problems have a version that's hard to approximate. In *Proceedings of the 8th IEEE Symposium on Structure in Complexity Theory*, pages 305–312, 1993. Journal version, entitled: On unapproximable versions of $\mathcal{NP}$-complete problems. In *SIAM Journal on Computing,* 25: 1293-1304, 1996.