37th Annual Symposium on Foundations of Computer Science

Proceedings

37th Annual Symposium on Foundations of Computer Science

October 14 – 16, 1996

Burlington, Vermont

Sponsored by

IEEE Computer Society

IEEE Computer Society Technical Committee on Mathematical Foundations of Computing

IEEE Computer Society Press
Los Alamitos, California

Washington • Brussels • Tokyo

IEEE Computer Society Press
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1264

IEEE Catalog Number 96CH35973
ISBN 0-8186-[illegible] (softbound)
ISBN 0-7803-3762-X (casebound)
ISBN 0-8186-7596-9 (microfiche)
Library of Congress: 80-646634
ISSN: 0272-5428

Additional copies may be ordered from:

IEEE Computer Society Press
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1264
Tel: +1-714-821-8380
Fax: +1-714-821-4641
Email: cs.books@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: +1-908-981-1393
Fax: +1-908-981-9667

IEEE Computer Society
13, Avenue de l'Aquilon
B-1200 Brussels
BELGIUM
Tel: +32-2-770-2198
Fax: +32-2-770-8505

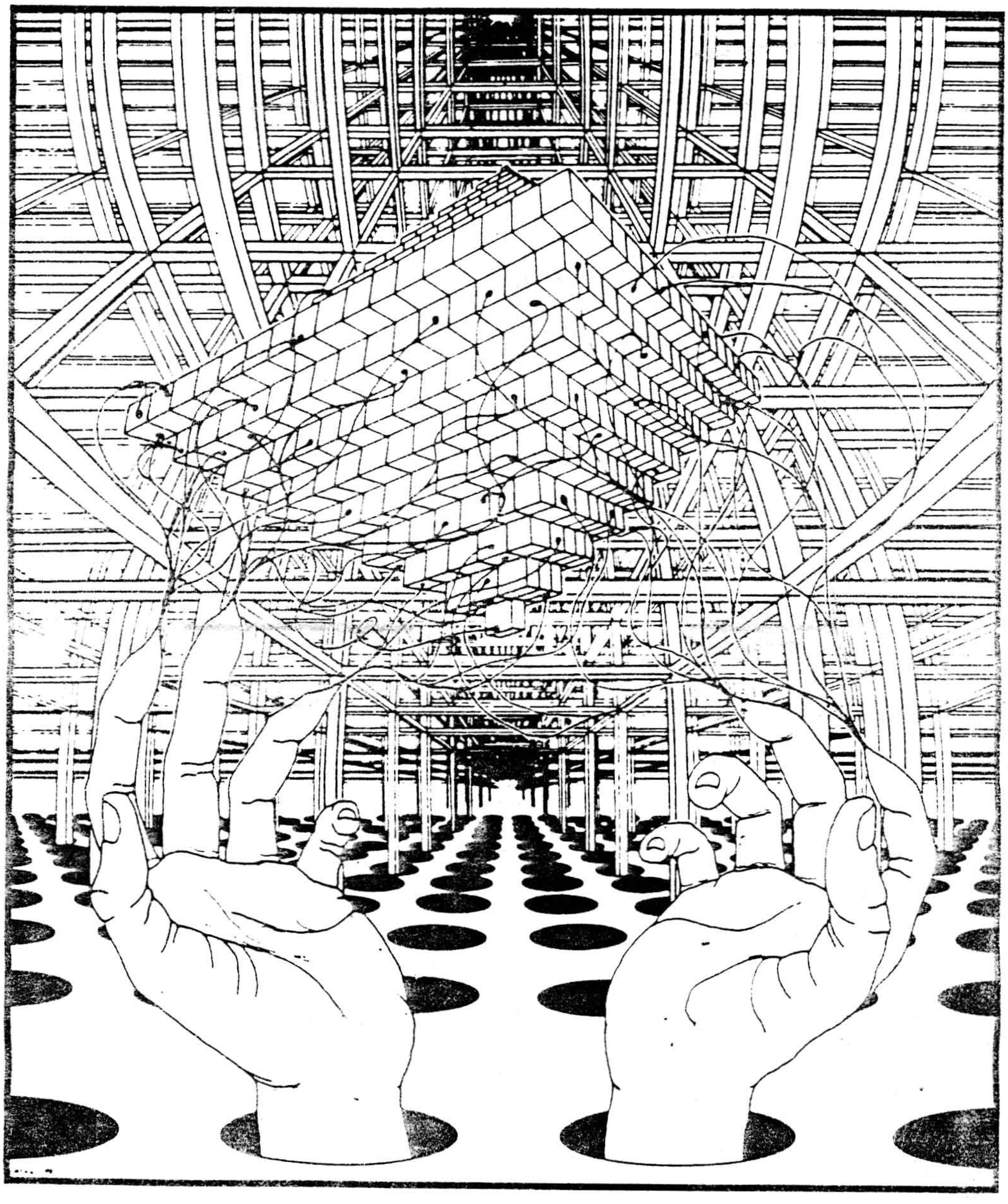
IEEE Computer Society
Ooshima Building
2-19-1 Minami-Aoyama
Minato-ku, Tokyo 107
JAPAN
Tel: +81-3-3408-3118
Fax: +81-3-3408-3553

Editorial production by Regina Spencer Sipple
Cover design by Alvy Ray Smith
Printed in the United States of America by KNI, Inc.

The Institute of Electrical and Electronics Engineers, Inc.

Proceedings

37th Annual Symposium on Foundations of Computer Science



Foreword

The papers in these proceedings were presented at the 37th Annual Symposium on Foundations of Computer Science (FOCS '96), sponsored by the IEEE Computer Society Technical Committee on Mathematical Foundations of Computing. The conference was held in Burlington, Vermont, October 14-16, 1996.

The program committee consisted of Anne Condon, Russell Impagliazzo, Sandy Irani, David Karger, Dexter Kozen, Rao Kosaraju, Michael Luby, Carsten Lund, Yishay Mansour, Rajeev Motwani, Michael Paterson, Baruch Schieber, Martin Tompa, Tandy Warnow, and Chee Yap. We met on June 2-3, 1996, and selected 63 papers from the 174 detailed abstracts submitted. In addition, Thomas Cover and Michael Rabin were invited to give plenary lectures, reprinted in these proceedings.

The submissions were not refereed, and many of them represent reports of continuing research. It is expected that most of these papers will appear in a more complete and polished form in scientific journals in the future.

The committee selected the paper "Single-Source Unsplittable Flow," by Jon Kleinberg, to receive the Machtey Award, given to the best student-authored paper. There were many excellent candidates for this award, each one deserving.

The committee wishes to thank all of those who submitted papers for consideration, as well as those who helped with the process of evaluating the submissions. A list of the latter individuals appears in these proceedings under the heading "Reviewers." The committee also wishes to thank Joe Kilian for the long hours he spent working on the electronic submission process, Danny Sleator and Judy Watson for expert and cheerful technical assistance, and Alok Aggarwal, Allan Borodin, and Prabhakar Raghavan for invaluable advice and assistance.

Martin Tompa
Program Committee Chair

Committees

General Conference and Technical Committee Chair

Alok Aggarwal, IBM Research

Outgoing General Conference and Technical Committee Chair

Allan Borodin, University of Toronto

Local Arrangements and Finance Chair

Carl Smith, University of Maryland

Program Committee Chair

Martin Tompa, University of Washington

Publicity Chair

Sorin Istrail, Sandia National Laboratories

Site Coordinator

Alok Aggarwal, IBM Research

Program Committee

Anne Condon
Russell Impagliazzo
Sandy Irani
David Karger
S. Rao Kosaraju
Dexter Kozen
Michael Luby
Carsten Lund
Yishay Mansour
Rajeev Motwani
Michael Paterson
Baruch Schieber
Martin Tompa
Tandy Warnow
Chee Yap

Reviewers

Eric Allender
Noga Alon
Kevin Atteson
Hagit Attiya
Yonatan Aumann
Baruch Awerbuch
Yossi Azar
Eric Bach
Amotz Bar-Noy
David Mix Barrington
Paul Beame
Michael Ben-Or
Andras Benczur
Charles H. Bennett
Avrim Blum
Ryan Borgstrom
Sam Buss
Paul Callahan
Ran Canetti
Pei Cao
Moses Charikar
Chandra Chekuri
Joseph Cheriyan
Ken Clarkson
Richard Cole
Don Coppersmith
David Eppstein
Martin Farach
Joan Feigenbaum
Anja Feldmann
Faith Fich
Lance Fortnow
Matt Franklin
Alan Frieze
Eli Gafni
William Gasarch
Michel Goemans
Leslie Goldberg
Oded Goldreich
Michael Goodrich
Mor Harchol
Bruce Hendrickson
Diane Hernek
Dorit Hochbaum
Neil Immerman
Piotr Indyk
Tao Jiang
David Johnson
Nabil Kahale
Sampath Kannan
Sanjeev Khanna
Samir Khuller
Joe Kilian
Valerie King
Jon Kleinberg
Phokion Kolaitis
Elias Koutsoupias
Hugo M. Krawczyk
Eyal Kushilevitz
Richard Ladner
Vitus Leung
Jack Lutz
Gary Miller
John Mitchell
S. Muthukrishnan
Joseph (Seffi) Naor
Moni Naor
Andrew Odylzko
Frank J. Oles
Steven Phillips
Nick Pippenger
Tal Rabin
Alexander Rabinovich
Prabhakar Raghavan
Ran Raz
Nick Reingold
Phil Rogaway
Ronitt Rubinfeld
Steven Rudich
Larry Ruzzo
Katherine St. John
Jeanette Schmidt
Steve Seiden
David Shmoys
Peter Shor
Meera Sitharam
Cliff Stein
Craig Silverstein
Alistair Sinclair
Greg Sorkin
Aravind Srinivasan
Madhu Sudan
Elizabeth Sweedyk
Eva Tardos
Jacobo Toran
Shang-Hua Teng
Manfred Warmuth
John Watrous
Joel Wein
Scott Weinstein
Avi Wigderson
David P. Williamson
David Bruce Wilson
Rebecca Wright
Shibu Yooseph
Neal Young

Table of Contents

SESSION 1A:

CHAIR: RAJEEV MOTWANI

Polynomial Time Approximation Schemes for Euclidean TSP and other Geometric Problems

Sanjeev Arora*
Princeton University

Abstract

We present a polynomial time approximation scheme for Euclidean TSP in $\Re^2$. Given any n nodes in the plane and $\epsilon > 0$, the scheme finds a $(1+\epsilon)$-approximation to the optimum traveling salesman tour in time $n^{O(1/\epsilon)}$. When the nodes are in $\Re^d$, the running time increases to $n^{\tilde{O}(\log^{d-2} n)/\epsilon^{d-1}}$. The previous best approximation algorithm for the problem (due to Christofides) achieves a 3/2-approximation in polynomial time.

We also give similar approximation schemes for a host of other Euclidean problems, including Steiner Tree, k-TSP, Minimum degree-k spanning tree, k-MST, etc. (This list may get longer; our techniques are fairly general.) The previous best approximation algorithms for all these problems achieved a constant-factor approximation.

All our algorithms also work, with almost no modification, when distance is measured using any geometric norm (such as ℓ_p for $p \geq 1$ or other Minkowski norms).

1 Introduction

In the *Traveling Salesman Problem* ("TSP"), we are given n nodes and for each pair $\{i, j\}$ of distinct nodes a distance $d_{i,j}$. We desire a closed path that visits each node exactly once (i.e., is a *salesman tour*) and incurs the least *cost* (which is the sum of the distances along the path). This classic problem has proved a rich testing ground for most important algorithmic ideas during the past few decades, and influenced the emergence of fields such as operations research, polyhedral theory and complexity theory. For a fascinating history, see Lawler et al. [29].

Since the 1970s, mounting evidence from complexity theory suggests that the problem is computationally difficult. Exact optimization is NP-hard (Karp [23]). So is approximating the optimum within *any* constant factor (Sahni and Gonzalez [40]). There are also other reasons to believe in the TSP's nastiness (cf. D^P completeness [37] and PLS-completeness [20]).

But TSP instances arising in practice are usually quite special, so the hardness results may not necessarily apply to them. In *metric TSP* the nodes lie in a metric space (i.e., the distances satisfy the triangle inequality). In *Euclidean TSP* the nodes lie in $\Re^2$ (or more generally, in $\Re^d$ for some d) and distance is defined using the ℓ_2 norm. Note that Euclidean TSP is a subcase of metric TSP.

Unfortunately, even Euclidean TSP is NP-hard (Papadimitriou [35], Garey, Graham, and Johnson [12]). Therefore algorithm designers were left with no choice but to consider more modest notions of a "good" solution. Karp [24], in a seminal work on *probabilistic analysis of algorithms*, showed that when the n nodes are picked uniformly and independently from unit square, then the *fixed dissection heuristic* with high probability finds tours whose cost is within $1+\epsilon$ of optimal (where $\epsilon > 0$ is arbitrarily small). Christofides [9] designed an *approximation algorithm* that on every instance of metric TSP computes a tour of cost at most 1.5 times the optimum.

Two decades of research failed to improve upon Christofides' algorithm for metric TSP. But some researchers continued to hope that even a PTAS might exist. A PTAS or *Polynomial-Time Approximation Scheme* is a polynomial-time algorithm — or a family of such algorithms— that, for each fixed $\epsilon > 0$, can approximate the problem within a factor $1+\epsilon$. (The running time could depend upon ϵ, but for each fixed ϵ has to be polynomial in the input size.) PTAS's are known for very few problems; two important ones are in [17, 22]. Recently Arora, Lund, Motwani, Sudan, and Szegedy [3] showed that if P $\neq$ NP, then metric TSP and many other problems do not have a PTAS. Their work relied upon the theory of *MAX-SNP-completeness* (Papadimitriou and Yannakakis [38]), the notion of *probabilistically checkable proofs* or *PCPs* (Feige, Goldwasser, Lovász, Safra and Szegedy [11], Arora and Safra [4]), and the connection between PCPs and hardness of approximation [11].

The status of Euclidean TSP remained open, however. In this paper, we show that Euclidean TSP has a PTAS. For

*Supported by NSF CAREER award NSF CCR-9502747 and an Alfred Sloan Fellowship. Email: arora@cs.princeton.edu

every $\epsilon > 0$, the PTAS computes a $(1+\epsilon)$-approximation to the optimal tour in $n^{O(1/\epsilon)}$ time. When the nodes are in $\Re^d$, the running time rises to $n^{O(\log^{d-2} n \log\log n)/\epsilon^{d-1}}$. Our techniques apply to many other geometric problems, which are described in Section 1.1.

We design the PTAS by showing that every TSP instance in $\Re^2$ (also in $\Re^d$ for every fixed d) has a $(1+\epsilon)$-approximate tour with the following very simple structure: there is a way to recursively partition the plane so that "very few" edges of the tour cross each line of the partition (see Theorem 4). A tour with such simple structure can be found using dynamic programming. We remark that the idea of partitioning a TSP instance into smaller instances has been used before, most famously in [24]. Dynamic programming has also been used before, most recently in an approximation scheme for planar graph TSP [28].

Our Structure Theorem about near-optimal tours also seems to shed some light — at least at an intuitive level— on one mysterious aspect of the TSP: the remarkable performance of simple heuristics. The most well-known of them, such as K-OPT or Lin-Kernighan [31], date to the 1960s and 70s. Using simple local-exchange rules, they quickly come up with very good salesman tours on "real-life" TSP instances [19, 5]. But many of these "real-life" instances are either Euclidean or derived from Euclidean instances! (See for example the TSPLIB library [39].) Since our structure theorem shows that such instances have near-optimal salesman tours with a very simple structure, the fact that simple heuristics can find such tours should be no mystery.

We cannot show, however, that any known heuristic is a PTAS[1]. But maybe our techniques will motivate further research on this topic. For example, even our current dynamic programming algorithm can be viewed (after some twists in the definition of "local search") as a local search algorithm that performs upto $O(\log n/\epsilon)$ edge exchanges per step (see Section 2.4).

Finally, the inevitable question: Is our PTAS practical? A straightforward implementation (for even moderate values of ϵ) is very slow, but we see no reason why a speedier, more subtle, implementation may not exist (see Section 4.1 in the appendix). At the very least, the Theorem gives a way of decomposing TSP instances into a large number of "independent" and smaller instances, and this may prove helpful in parallelizing existing TSP routines.

[1] In fact, thus far there is no evidence that any of the known heuristics is a PTAS for Euclidean TSP. The few published results in fact suggest quite the opposite. With an adversarially-chosen starting tour, K-OPT (for any constant K) may produce a tour whose cost is $\Omega(\log n/\log\log n)$ times the cost of the optimum tour, even when the n nodes lie in $\Re^2$ [8]. In case of metric TSP, finding a locally-optimum tour for K-OPT (for $K \geq 8$) is PLS-complete [26]. (This strongly suggests that no polynomial-time algorithm can find such a local optimum; see [20].) Many variants of Lin-Kernighan are also PLS-complete [36].

1.1 Steiner Tree and other geometric problems.

Many network problems identified in the past few decades are very similar to the TSP. Below, we define some of them. We will restrict attention to the Euclidean (or geometric) versions of these problems. The best approximation algorithms for all of them achieve a constant factor approximation in polynomial time (see the survey by Bern and Eppstein [6]). We are able to design PTAS's for the planar versions, and $n^{O(\log^{d-2} n \log\log n)/\epsilon^{d-1}}$ time approximation schemes for the $\Re^d$ versions.

Minimum Steiner Tree: Given n nodes in $\Re^d$, find the minimum cost tree connecting them[2]. In general, the minimum spanning tree is not an optimal solution (as observed many decades ago). In $\Re^2$ the cost of the MST can be as far as a factor $2/\sqrt{3}$ from the optimum. (Furthermore, the famous Gilbert-Pollak [13] conjecture said it can't be any further from the optimum; the conjecture was proved by Du and Hwang [10]). A spate of research activity in recent years (starting with the work of Zelikovsky[42]) has provided better algorithms; with an approximation ratio approaching 1.10. The metric case is MAX-SNP-hard.

Degree-restricted-MST. Given n nodes in $\Re^d$ and an integer $k \geq 2$, find the minimum cost spanning tree in which every node has degree at most k. When $k = 2$, the problem is polynomial-time equivalent to the TSP and hence NP-hard. The case $k = 3$ is NP-hard; $k = 4$ is open, and when $k \geq 5$ the problem can be solved optimally in polynomial time. For the cases $k = 3, 4$ in $\Re^2$, a constant-factor approximation algorithm is given by Khuller, Raghavachari, and Young [27].

k-TSP: Given n nodes in $\Re^d$ and an integer $k > 1$, find the smallest tour that visits at least k nodes. An approximation algorithm due to Mata and Mitchell [32] achieves a constant factor approximation in $\Re^2$.

k-MST: Given n nodes in $\Re^d$ and an integer $k \geq 2$, find k nodes with the shortest MST. Blum, Chalasani, and Vempala [7] gave the first $O(1)$-factor approximation algorithm for points in $\Re^2$; there has been much other work before and since.

We remark that until recently the approximation algorithms for the last three problems heavily used the geometry of the plane and broke down even in $\Re^3$. But recent algorithms — discovered independently of our paper — work for any metric space.

[2] It appears that this problem was first posed by Gauss in a letter to Schumacher [15].

2 The TSP Algorithm

As mentioned in the introduction, we design our PTAS for Euclidean TSP in $\Re^2$ by showing that there is a way to recursively partition the plane so that there exists a $(1+\epsilon)$-approximate tour that crosses each line of this partition "very few" times. We state this formally in Theorem 4 and then describe the PTAS. Theorem 4 is proved in Section 2.1, and the algorithm for $\Re^d$ is described in Section 2.2.

In this paper, whenever we say "rectangle", we mean an axis-aligned rectangle. The *size* of the rectangle is the length of its longer edge. The *bounding box* of a set of nodes is the smallest rectangle enclosing them.

To simplify the exposition, we first modify the TSP instances a little so that internode distances are not too different from one another.

Proposition 1 *Let $n, \epsilon > 0$ be such that $n > 10/\epsilon$. Then the problem of computing a $(1+\epsilon)$-approximation to the optimum tour length in an n-node instance can be reduced in poly(n) time to the problem of computing a $(1+9\epsilon/10)$-approximation in an instance in which the the smallest internode distance is 1 unit and the size of the bounding box is at most $1.5n^2$.*

Proof: The reduction involves perturbing the n-node instance a little. Let T be the cost of the minimum spanning tree. Note that the optimum salesman tour has cost at least T and at most $1.5T$ [9]. So the size of the bounding box is at most $0.75T$. Construct a new instance by placing a grid of granularity $T/2n^2$ in the plane and moving each node to its nearest gridpoint (this may cause some nodes to merge). Because each node moved by at most $T/2n^2$ and a salesman tour has n edges, the tour cost changed by at most $2n \cdot T/2n^2 = T/n$. Now since $\epsilon/10 > 1/n$, it suffices to compute a $(1+9\epsilon/10)$-approximation in the new instance.

Finally, divide all distances in the new instance by $T/2n^2$, so that the smallest internode distance is at least 1 and the bounding box has size at most $0.75T/(T/2n^2) < 1.5n^2$.
□

Now we define a recursive partition of a rectangle. A *line separator* of a rectangle is a straight line segment parallel to its shorter edge that partitions it into two rectangles of at least 1/3rd the area. For example, if the rectangle's width W is greater than its height, then a line separator is any vertical line in the middle $W/3$ of the rectangle.

Definition 1 (1/3:2/3 tiling) *A* $1/3:2/3$-tiling *of a rectangle R is a binary tree (i.e., a hierarchy) of sub-rectangles of R. The rectangle R is at the* root*. If the size of R is ≤ 1, then the hierarchy contains nothing else. Otherwise the root contains a line separator for R, and has two subtrees that are $1/3:2/3$-tilings of the two rectangles into which the line separator divides R.*

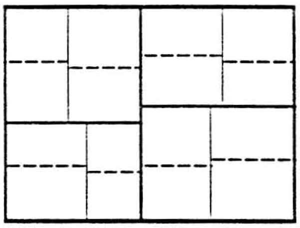


Figure 1. A $1/3:2/3$ tiling

The depth *of the tiling is the maximum depth of this tree.*

Note that rectangles at depth d in the tiling form a partition of the root rectangle. The set of all rectangles at depth $d+1$ is a refinement of this partition obtained by putting a line separator through each depth d rectangle of size > 1. The area of any depth d rectangle is at most $(2/3)^d$ times the total area. The following proposition is therefore immediate.

Proposition 2 *If a rectangle has width W and height H, then its every $1/3:2/3$ tiling has depth at most $\log_{1.5} W + \log_{1.5} H + 2$.*

We need one more definition about tilings.

Definition 2 (portals) *A* portal *in a $1/3:2/3$-tiling is any point that lies on the edge of some rectangle in the tiling. If m is any positive integer then a set of portals P is called* m-regular *for the tiling if there are exactly m equidistant portals on the line separator of each rectangle of the tiling. (We assume that the the end-points of the line separator are also portals. In other words the line separator is partitioned into exactly $m-1$ equal parts by the portals on it.)*

Now we indicate how the above ideas are used. We first note that an optimum salesman tour is always a simple polygon. To simplify the exposition, we will allow tours with "bent" edges. These bent edges arise as follows. We introduce additional ("Steiner") nodes in the plane (these will be portals in some $1/3:2/3$ tiling) and ask that the tour visit these nodes in addition to the input nodes. We call such a tour a *salesman path*. Of course, at the end of the algorithm we can change a salesman path into a polygonal tour by straightening the bent edges (i.e., removing the additional points).

To avoid disturbing the tour too much, we wish to limit the number of additional nodes. This motivates the following definition.

Definition 3 (m-light) *Let $m \in Z^+$ and π be a salesman path on some set of nodes. Let S be a $1/3:2/3$ tiling of the bounding box and P be an m-regular set of portals on this tiling. Then π is* m-light *with respect to S if the following*