

*Bang Ye Wu and Kun-Mao Chao*

---

# *Spanning Trees and Optimization Problems*

*CRC PRESS*

*Boca Raton London New York Washington, D.C.*

---

## *Preface*

The research on spanning trees has been one of the most important areas in algorithm design. People who are interested in algorithms will find this book informing and inspiring. The new results are still accumulating, and we try to make it clear the whole picture of the current status and future developments.

This book is written for graduate or advanced undergraduate students in computer science, electrical engineering, industrial engineering, and mathematics. It is also a good reference for professionals.

Our motivations for writing up this book:

1. To the best of our knowledge, there is no book totally dedicated to the topics of spanning trees.
2. Our recent progress in spanning trees reveals a new line of investigation.
3. Designing approximation algorithms for spanning tree problems have become an exciting and important field in theoretical computer science.
4. Besides numerous network design applications, spanning trees have also been playing important roles in newly established research areas, such as biological sequence alignments, and evolutionary tree construction.

This book is a general, and rigorous text on algorithms for spanning trees. It covers the full spectrum of spanning tree algorithms from classical computer science to modern applications. The selected topics in this book makes it an excellent handbook on algorithms for spanning trees. At the end of every chapter, we report related work and recent progress.

We first explain general properties of spanning trees. We then focus on three categories of spanning trees, namely, minimum spanning trees, shortest-paths trees, and optimum routing cost spanning trees. We also show how to balance the tree costs. Besides the theoretical description of the methods, many examples are used to illustrate the ideas behind. Moreover, we demonstrate some applications of these spanning trees. We explore in details some other interesting spanning trees, including maximum leaf spanning trees and minimum diameter spanning trees. In addition, Steiner trees and evolutionary trees are also discussed. We close this book by summarizing other important problems related to spanning trees.

Writing a book is not as easy as we thought at the very beginning of this project. We have tried our best to make it consistent and correct. However, it's a mission impossible for imperfect authors to produce a perfect book.

Should you find any mathematical, historical, or typographical errors, please kindly let us know.

We are extremely grateful to our mentor Chuan Yi Tang, who always makes the subject of algorithms exciting and beautiful in his superb lectures. His guidance and suggestions throughout this study were indispensable. We also thank our colleagues in the weekly theory-group seminar hosted by Richard Chia-Tung Lee and Chuan Yi Tang.

We thank Vineet Bafna, Giuseppe Lancia, and R. Ravi for their collaborations on the minimum routing cost spanning tree problem. We thank Tao Jiang and Howard Karloff for suggesting the merger of two different works at the early stage of our investigation. We thank the anonymous reviewers for their valuable comments.

It has been a pleasure working with CRC Press in the development of this book. We are very proud to have this book included in the CRC series on Discrete Mathematics and its Applications edited by Kenneth H. Rosen. Ken also provided critical reviews and invaluable information for which we are grateful. We thank Sunil Nair for his final approval of our proposal. Richard O'Hanley was the first to inform us to think about the possibility of publishing a book at CRC Press. Robert B. Stern then handled the proposal review and contract arrangements in a perfect and efficient way. Bob also proposed many constructive suggestions throughout the project. Jamie B. Sigal helped us with both production and permissions issues, and his gentle reminders kept us moving on in a right pace.

Finally, we thank our families for their love, patience, and encouragement. We thank our wives — Mei-Ling Cheng and Pei-Ju Tsai, and our sons — Ming-Hsuan Wu and Leo Liang Chao for tolerating our absent-mindedness during the writing of this book. We promise to work less than 168 hours a week by not taking on a new grand project immediately.

Bang Ye Wu  
bangye@mail.stu.edu.tw  
<http://www.personal.stu.edu.tw/bangye>

Kun-Mao Chao  
kmchao@csie.ntu.edu.tw  
<http://www.csie.ntu.edu.tw/~kmchao>

September, 2003

---

# *Contents*

<b>Preface</b>	<b>i</b>
<b>1 Spanning Trees</b>	<b>1</b>
1.1 Counting Spanning Trees . . . . .	1
<b>2 Minimum Spanning Trees</b>	<b>9</b>
2.1 Introduction . . . . .	9
2.2 Borůvka's Algorithm . . . . .	11
2.3 Prim's Algorithm . . . . .	13
2.4 Kruskal's Algorithm . . . . .	15
2.5 Applications . . . . .	17
2.5.1 Cable TV . . . . .	17
2.5.2 Circuit design . . . . .	17
2.5.3 Islands connection . . . . .	17
2.5.4 Clustering gene expression data . . . . .	17
2.5.5 MST-based approximations . . . . .	18
2.6 Summary . . . . .	18
Bibliographic Notes and Further Reading . . . . .	19
Exercies . . . . .	20
<b>3 Shortest-Paths Trees</b>	<b>23</b>
3.1 Introduction . . . . .	23
3.2 Dijkstra's Algorithm . . . . .	25
3.3 The Bellman-Ford Algorithm . . . . .	32
3.4 Applications . . . . .	36
3.4.1 Multicast . . . . .	36
3.4.2 SPT-based approximations . . . . .	37
3.5 Summary . . . . .	37
Bibliographic Notes and Further Reading . . . . .	37
Exercies . . . . .	38
<b>4 Minimum Routing Cost Spanning Trees</b>	<b>41</b>
4.1 Introduction . . . . .	41
4.2 Approximating by a Shortest-Paths Tree . . . . .	44
4.2.1 A simple analysis . . . . .	44
4.2.2 Solution decomposition . . . . .	46
4.3 Approximating by a General Star . . . . .	47

4.3.1	Separators and general stars . . . . .	47
4.3.2	A $15/8$ -approximation algorithm . . . . .	52
4.3.3	A $3/2$ -approximation algorithm . . . . .	55
4.3.4	Further improvement . . . . .	56
4.4	A Reduction to the Metric Case . . . . .	58
4.5	A Polynomial Time Approximation Scheme . . . . .	61
4.5.1	Overview . . . . .	61
4.5.2	The $\delta$ -spine of a tree . . . . .	66
4.5.3	Lower bound . . . . .	69
4.5.4	From trees to stars . . . . .	70
4.5.5	Finding an optimal $k$ -star . . . . .	74
4.6	Applications . . . . .	79
4.6.1	Network design . . . . .	79
4.6.2	Computational biology . . . . .	79
4.7	Summary . . . . .	82
	Bibliographic Notes and Further Reading . . . . .	82
	Exercises . . . . .	83
<b>5</b>	<b>Optimum Communication Spanning Trees</b>	<b>85</b>
5.1	Introduction . . . . .	85
5.2	Product-Requirement . . . . .	87
5.2.1	Overview . . . . .	87
5.2.2	Preliminaries . . . . .	88
5.2.3	Approximating by 2-stars . . . . .	91
5.2.4	A polynomial time approximation scheme . . . . .	98
5.3	Sum-Requirement . . . . .	104
5.4	Multiple Sources . . . . .	109
5.4.1	Computational complexity for fixed $p$ . . . . .	111
5.4.2	A PTAS for the 2-MRCT . . . . .	115
5.5	Applications . . . . .	124
5.6	Summary . . . . .	125
	Bibliographic Notes and Further Reading . . . . .	125
	Exercises . . . . .	127
<b>6</b>	<b>Balancing the Tree Costs</b>	<b>129</b>
6.1	Introduction . . . . .	129
6.2	Light Approximate Shortest Path Trees . . . . .	130
6.2.1	Overview . . . . .	130
6.2.2	The algorithm . . . . .	131
6.2.3	The analysis of the algorithm . . . . .	134
6.3	Light Approximate Routing Cost Spanning Trees . . . . .	136
6.3.1	Overview . . . . .	136
6.3.2	The algorithm . . . . .	137
6.3.3	The performance analysis . . . . .	140
6.3.4	On general graphs . . . . .	143

6.4 Applications . . . . .	143
6.5 Summary . . . . .	144
Bibliographic Notes and Further Reading . . . . .	144
Exercises . . . . .	145
<b>7 Steiner Trees and Some Other Problems</b>	<b>147</b>
7.1 Steiner Minimal Trees . . . . .	147
7.1.1 Approximation by MST . . . . .	148
7.1.2 Improved approximation algorithms . . . . .	151
7.2 Trees and Diameters . . . . .	154
7.2.1 Eccentricities, diameters, and radii . . . . .	154
7.2.2 The minimum diameter spanning trees . . . . .	158
7.3 Maximum Leaf Spanning Trees . . . . .	162
7.3.1 Leafy trees and leafy forests . . . . .	162
7.3.2 The algorithm . . . . .	165
7.3.3 Performance ratio . . . . .	166
7.4 Some Other Problems . . . . .	168
7.4.1 Network design . . . . .	168
7.4.2 Computational biology . . . . .	170
Bibliographic Notes and Further Reading . . . . .	173
Exercises . . . . .	174
<b>References</b>	<b>175</b>
<b>Index</b>	<b>181</b>



---

## *List of Tables*

5.1	OCT problems and their approximation ratios . . . . .	126
-----	---	-----



---

## List of Figures

1.1	A four-vertex complete graph $K_4$ . . . . .	2
1.2	All sixteen spanning trees . . . . .	2
1.3	An eight-vertex spanning tree . . . . .	4
1.4	Generating a Prüfer sequence from a spanning tree . . . . .	5
1.5	Recovering a spanning tree from a Prüfer sequence . . . . .	6
2.1	A weighted graph . . . . .	9
2.2	Some minimum spanning trees . . . . .	10
2.3	The execution of the BORŮVKA algorithm . . . . .	12
2.4	The execution of the PRIM algorithm . . . . .	14
2.5	The execution of the KRUSKAL algorithm . . . . .	16
3.1	A shortest-paths tree . . . . .	24
3.2	An undirected graph with a negative-weight edge . . . . .	24
3.3	A weighted, directed graph . . . . .	26
3.4	The execution of Dijkstra's algorithm (1) . . . . .	26
3.5	The execution of Dijkstra's algorithm (2) . . . . .	27
3.6	The execution of Dijkstra's algorithm (3) . . . . .	27
3.7	The execution of Dijkstra's algorithm (4) . . . . .	28
3.8	The execution of Dijkstra's algorithm (5) . . . . .	28
3.9	The execution of Dijkstra's algorithm (6) . . . . .	29
3.10	The execution of Dijkstra's algorithm (7) . . . . .	29
3.11	The execution of Dijkstra's algorithm (8) . . . . .	30
3.12	A shortest-paths tree . . . . .	30
3.13	A weighted, directed graph with a negative-weight edge . . . . .	31
3.14	A failure of Dijkstra's algorithm . . . . .	31
3.15	A wrong shortest-paths tree . . . . .	32
3.16	A correct shortest-paths tree . . . . .	32
3.17	The execution of Bellman-Ford algorithm (1) . . . . .	34
3.18	The execution of Bellman-Ford algorithm (2) . . . . .	34
3.19	The execution of Bellman-Ford algorithm (3) . . . . .	35
3.20	The execution of Bellman-Ford algorithm (4) . . . . .	35
3.21	The execution of Bellman-Ford algorithm (5) . . . . .	36
3.22	A directed graph with a negative-weight cycle . . . . .	39
4.1	An example illustrating the routing loads . . . . .	43
4.2	Two extreme trees . . . . .	44

4.3	A centroid of a tree . . . . .	46
4.4	An example of a minimal separator a tree . . . . .	48
4.5	A $\delta$ -separator . . . . .	49
4.6	Replacing bad edges . . . . .	59
4.7	Transforming to 3-stars . . . . .	64
4.8	Branches of a $\delta$ -path . . . . .	66
4.9	Separator, spine and <i>CAL</i> . . . . .	68
4.10	From trees to stars . . . . .	70
4.11	The configuration of a 3-star . . . . .	75
4.12	The auxiliary graphs for the faster method . . . . .	78
5.1	The p.r.c. and s.r.c. costs . . . . .	86
5.2	The relationship of the OCT problems . . . . .	87
5.3	A counterexample of the incremental method . . . . .	93
5.4	The approximation ratio of a 2-star . . . . .	95
5.5	A pseudo-polynomial time method for PROCT . . . . .	99
5.6	Bad edges and s.r.c. cost . . . . .	106
5.7	Transformation from SAT to 2-MRCT( $\alpha$ ) . . . . .	112
5.8	Partition for 2-MRCT . . . . .	119
5.9	Breaking cycle for 2-MRCT . . . . .	122
6.1	An LAST of vertices in Euclidean space . . . . .	130
6.2	The Eulerian cycle on a doubling tree . . . . .	132
6.3	A sample execution of algorithm FIND-LAST . . . . .	133
6.4	A MST with large routing cost . . . . .	136
6.5	A shortest-paths tree and an LASF . . . . .	138
6.6	Execution of Algorithm FIND-LART . . . . .	139
7.1	Steiner minimal trees . . . . .	148
7.2	A sample execution of Algorithm MST-STEINER . . . . .	150
7.3	Steiner minimal trees and minimum spanning trees . . . . .	151
7.4	A sample execution of the ZELIKOVSKY-STEINER algorithm . . . . .	152
7.5	Diameters, centers and radii . . . . .	154
7.6	Diameters and centers of a tree . . . . .	157
7.7	Leafy trees . . . . .	163
7.8	A maximal leafy forest . . . . .	164

## Symbol Description

$Z^+$	the set of all positive integers
$Z_0^+$	the set of all nonnegative integers
$ S $	the cardinality of a set $S$
$G = (V, E, w)$	a graph $G$ with vertex set $V$ , edge set $E$ , and edge weight function $w : E \rightarrow Z_0^+$
$V(G)$	the vertex set of graph $G$
$E(G)$	the edge set of graph $G$
$w(G)$	the total length of edges in $E(G)$ , i.e., $\sum_{e \in E(G)} w(e)$
$SP_G(u, v)$	a shortest path between vertices $u$ and $v$ on graph $G$
$d_G(u, v)$	the distance (shortest path length) between $u$ and $v$ on $G$ , i.e., $w(SP_G(u, v))$
$d_G(v, U)$	the minimum distance from vertex $v$ to any vertex in $U$ , i.e., $\min_{u \in U} \{d_G(v, u)\}$
$\bar{G}$	the metric closure of a graph $G$
$D_G(v, U)$	the maximum distance from vertex $v$ to any vertex in $U$ , i.e., $\max_{u \in U} \{d_G(v, u)\}$
$G - e$	the graph obtained by removing edge $e$ from $G$
$G + e$	the graph obtained by inserting edge $e$ into $G$
$G \setminus e$	the resulting graph after contracting $e$ in $G$
$parent(v)$	the parent of a vertex $v$ in a rooted tree
$n$	the number of vertices of the input graph
$m$	the number of edges of the input graph