LNCS 2204

Andreas Brandstädt
Van Bang Le (Eds.)

Graph-Theoretic Concepts in Computer Science

27th International Workshop, WG 2001
Boltenhagen, Germany, June 2001
Proceedings

Springer

Andreas Brandstädt   Van Bang Le (Eds.)

Graph-Theoretic Concepts in Computer Science

27th International Workshop, WG 2001
Boltenhagen, Germany, June 14-16, 2001
Proceedings

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Andreas Brandstädt
Van Bang Le
University of Rostock, Department of Computer Science
18051 Rostock, Germany
E-mail: {ab/le}@informatik.uni-rostock.de

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Graph theoretic concepts in computer science : 27th international workshop ; proceedings / WG 2001, Boltenhagen, Germany, June 14 - 16, 2001. Andreas Brandstädt ; Van Bang Le (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2001
(Lecture notes in computer science ; Vol. 2204)
ISBN 3-540-42707-4

CR Subject Classification (1998): F.2, G.1.2, G.1.6, G.2, G.3, E.1, I.3.5

ISSN 0302-9743
ISBN 3-540-42707-4 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by Steingräber Satztechnik GmbH, Heidelberg
Printed on acid-free paper SPIN: 10840737 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 2204
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Lecture Notes in Computer Science

For information about Vols. 1–2126
please contact your bookseller or Springer-Verlag

Vol. 2127: V. Malyshkin (Ed.), Parallel Computing Technologies. Proceedings, 2001. XII, 516 pages. 2001.

Vol. 2129: M. Goemans, K. Jansen, J.D.P. Rolim, L. Trevisan (Eds.), Approximation, Randomization, and Combinatorial Optimization. Proceedings, 2001. IX, 297 pages. 2001.

Vol. 2130: G. Dorffner, H. Bischof, K. Hornik (Eds.), Artificial Neural Networks – ICANN 2001. Proceedings, 2001. XXII, 1259 pages. 2001.

Vol. 2131: Y. Cotronis, J. Dongarra (Eds.), Recent Advances in Parallel Virtual Machine and Message Passing Interface. Proceedings, 2001. XV, 438 pages. 2001.

Vol. 2132: S.-T. Yuan, M. Yokoo (Eds.), Intelligent Agents. Specification. Modeling, and Application. Proceedings, 2001. X, 237 pages. 2001. (Subseries LNAI).

Vol. 2133: B. Christianson, B. Crispo, J.A. Malcolm, M. Roe (Eds.), Security Protocols. Proceedings, 2001. VIII, 257 pages. 2001.

Vol. 2134: M. Figueiredo, J. Zerubia, A.K. Jain (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition. Proceedings, 2001. X, 652 pages. 2001.

Vol. 2135: G.N.C. Kirby, A. Dearle, D.I.K. Sjøberg (Eds.), Persistent Object Systems. Proceedings, 2000. VIII, 321 pages. 2001.

Vol. 2136: J. Sgall, A. Pultr, P. Kolman (Eds.), Mathematical Foundations of Computer Science 2001. Proceedings, 2001. XII, 716 pages. 2001.

Vol. 2138: R. Freivalds (Ed.), Fundamentals of Computation Theory. Proceedings, 2001. XIII, 542 pages. 2001.

Vol. 2139: J. Kilian (Ed.), Advances in Cryptology – CRYPTO 2001. Proceedings, 2001. XI, 599 pages. 2001.

Vol. 2140: I. Attali, T. Jensen (Eds.), Java on Smart Cards: Programming and Security. Proceedings, 2001. VIII, 255 pages. 2001.

Vol. 2141: G.S. Brodal, D. Frigioni, A. Marchetti-Spaccamela (Eds.), Algorithm Engineering. Proceedings, 2001. X, 199 pages. 2001.

Vol. 2142: L. Fribourg (Ed.), Computer Science Logic. Proceedings, 2001. XII, 615 pages. 2001.

Vol. 2143: S. Benferhat, P. Besnard (Eds.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty. Proceedings, 2001. XIV, 818 pages. 2001. (Subseries LNAI).

Vol. 2144: T. Margaria, T. Melham (Eds.), Correct Hardware Design and Verification Methods. Proceedings, 2001. XII, 482 pages. 2001.

Vol. 2145: M. Leyton, A Generative Theory of Shape. XVI, 554 pages. 2001.

Vol. 2146: J.H. Silverman (Eds.), Cryptography and Lattices. Proceedings, 2001. VII, 219 pages. 2001.

Vol. 2147: G. Brebner, R. Woods (Eds.), Field-Programmable Logic and Applications. Proceedings, 2001. XV, 665 pages. 2001.

Vol. 2149: O. Gascuel, B.M.E. Moret (Eds.), Algorithms in Bioinformatics. Proceedings, 2001. X, 307 pages. 2001.

Vol. 2150: R. Sakellariou, J. Keane, J. Gurd, L. Freeman (Eds.), Euro-Par 2001 Parallel Processing. Proceedings, 2001. XXX, 943 pages. 2001.

Vol. 2151: A. Caplinskas, J. Eder (Eds.), Advances in Databases and Information Systems. Proceedings, 2001. XIII, 381 pages. 2001.

Vol. 2152: R.J. Boulton, P.B. Jackson (Eds.), Theorem Proving in Higher Order Logics. Proceedings, 2001. X, 395 pages. 2001.

Vol. 2153: A.L. Buchsbaum, J. Snoeyink (Eds.), Algorithm Engineering and Experimentation. Proceedings, 2001. VIII, 231 pages. 2001.

Vol. 2154: K.G. Larsen, M. Nielsen (Eds.), CONCUR 2001 – Concurrency Theory. Proceedings, 2001. XI, 583 pages. 2001.

Vol. 2156: M.I. Smirnov, J. Crowcroft, J. Roberts, F. Boavida (Eds.), Quality of Future Internet Services. Proceedings, 2001. XI, 333 pages. 2001.

Vol. 2157: C. Rouveirol, M. Sebag (Eds.), Inductive Logic Programming. Proceedings, 2001. X, 261 pages. 2001. (Subseries LNAI).

Vol. 2158: D. Shepherd, J. Finney, L. Mathy, N. Race (Eds.), Interactive Distributed Multimedia Systems. Proceedings, 2001. XIII, 258 pages. 2001.

Vol. 2159: J. Kelemen, P. Sosík (Eds.), Advances in Artificial Life. Proceedings, 2001. XIX, 724 pages. 2001. (Subseries LNAI).

Vol. 2161: F. Meyer auf der Heide (Ed.), Algorithms – ESA 2001. Proceedings, 2001. XII, 538 pages. 2001.

Vol. 2162: Ç. K. Koç, D. Naccache, C. Paar (Eds.), Cryptographic Hardware and Embedded Systems – CHES 2001. Proceedings, 2001. XIV, 411 pages. 2001.

Vol. 2163: P. Constantopoulos, I.T. Sølvberg (Eds.), Research and Advanced Technology for Digital Libraries. Proceedings, 2001. XII, 462 pages. 2001.

Vol. 2164: S. Pierre, R. Glitho (Eds.), Mobile Agents for Telecommunication Applications. Proceedings, 2001. XI, 292 pages. 2001.

Vol. 2165: L. de Alfaro, S. Gilmore (Eds.), Process Algebra and Probabilistic Methods. Proceedings, 2001. XII, 217 pages. 2001.

Vol. 2166: V. Matoušek, P. Mautner, R. Mouček, K. Taušer (Eds.), Text, Speech and Dialogue. Proceedings, 2001. XIII, 452 pages. 2001. (Subseries LNAI).

Vol. 2167: L. De Raedt, P. Flach (Eds.), Machine Learning: ECML 2001. Proceedings, 2001. XVII, 618 pages. 2001. (Subseries LNAI).

Vol. 2168: L. De Raedt, A. Siebes (Eds.), Principles of Data Mining and Knowledge Discovery. Proceedings, 2001. XVII, 510 pages. 2001. (Subseries LNAI).

Vol. 2170: S. Palazzo (Ed.), Evolutionary Trends of the Internet. Proceedings, 2001. XIII, 722 pages. 2001.

Vol. 2172: C. Batini, F. Giunchiglia, P. Giorgini, M. Mecella (Eds.), Cooperative Information Systems. Proceedings, 2001. XI, 450 pages. 2001.

Vol. 2173: T. Eiter, W. Faber, M. Truszczynski (Eds.), Logic Programming and Nonmonotonic Reasoning. Proceedings, 2001. XI, 444 pages. 2001. (Subseries LNAI).

Vol. 2174: F. Baader, G. Brewka, T. Eiter (Eds.), KI 2001: Advances in Artificial Intelligence. Proceedings, 2001. XIII, 471 pages. 2001. (Subseries LNAI).

Vol. 2175: F. Esposito (Ed.), AI*IA 2001: Advances in Artificial Intelligence. Proceedings, 2001. XII, 396 pages. 2001. (Subseries LNAI).

Vol. 2176: K.-D. Althoff, R.L. Feldmann, W. Müller (Eds.), Advances in Learning Software Organizations. Proceedings, 2001. XI, 241 pages. 2001.

Vol. 2177: G. Butler, S. Jarzabek (Eds.), Generative and Component-Based Software Engineering. Proceedings, 2001. X, 203 pages. 2001.

Vol. 2180: J. Welch (Ed.), Distributed Computing. Proceedings, 2001. X, 343 pages. 2001.

Vol. 2181: C. Y. Westort (Ed.), Digital Earth Moving. Proceedings, 2001. XII, 117 pages. 2001.

Vol. 2182: M. Klusch, F. Zambonelli (Eds.), Cooperative Information Agents V. Proceedings, 2001. XII, 288 pages. 2001. (Subseries LNAI).

Vol. 2183: R. Kahle, P. Schroeder-Heister, R. Stärk (Eds.), Proof Theory in Computer Science. Proceedings, 2001. IX, 239 pages. 2001.

Vol. 2184: M. Tucci (Ed.), Multimedia Databases and Image Communication. Proceedings, 2001. X, 225 pages. 2001.

Vol. 2185: M. Gogolla, C. Kobryn (Eds.), «UML» 2001 – The Unified Modeling Language. Proceedings, 2001. XIV, 510 pages. 2001.

Vol. 2186: J. Bosch (Ed.), Generative and Component-Based Software Engineering. Proceedings, 2001. VIII, 177 pages. 2001.

Vol. 2187: U. Voges (Ed.), Computer Safety, Reliability and Security. Proceedings, 2001. XVI, 261 pages. 2001.

Vol. 2188: F. Bomarius, S. Komi-Sirviö (Eds.), Product Focused Software Process Improvement. Proceedings, 2001. XI, 382 pages. 2001.

Vol. 2189: F. Hoffmann, D.J. Hand, N. Adams, D. Fisher, G. Guimaraes (Eds.), Advances in Intelligent Data Analysis. Proceedings, 2001. XII, 384 pages. 2001.

Vol. 2190: A. de Antonio, R. Aylett, D. Ballin (Eds.), Intelligent Virtual Agents. Proceedings, 2001. VIII, 245 pages. 2001. (Subseries LNAI).

Vol. 2191: B. Radig, S. Florczyk (Eds.), Pattern Recognition. Proceedings, 2001. XVI, 452 pages. 2001.

Vol. 2192: A. Yonezawa, S. Matsuoka (Eds.), Metalevel Architectures and Separation of Crosscutting Concerns. Proceedings, 2001. XI, 283 pages. 2001.

Vol. 2193: F. Casati, D. Georgakopoulos, M.-C. Shan (Eds.), Technologies for E-Services. Proceedings, 2001. X, 213 pages. 2001.

Vol. 2194: A.K. Datta, T. Herman (Eds.), Self-Stabilizing Systems. Proceedings, 2001. VII, 229 pages. 2001.

Vol. 2195: H.-Y. Shum, M. Liao, S.-F. Chang (Eds.), Advances in Multimedia Information Processing – PCM 2001. Proceedings, 2001. XX, 1149 pages. 2001.

Vol. 2196: W. Taha (Ed.), Semantics, Applications, and Implementation of Program Generation. Proceedings, 2001. X, 219 pages. 2001.

Vol. 2197: O. Balet, G. Subsol, P. Torguet (Eds.), Virtual Storytelling. Proceedings, 2001. XI, 213 pages. 2001.

Vol. 2198: N. Zhong, Y. Yao, J. Liu, S. Ohsuga (Eds.), Web Intelligence: Research and Development. Proceedings, 2001. XVI, 615 pages. 2001. (Subseries LNAI).

Vol. 2199: J. Crespo, V. Maojo, F. Martin (Eds.), Medical Data Analysis. Proceedings, 2001. X, 311 pages. 2001.

Vol. 2200: G.I. Davida, Y. Frankel (Eds.), Information Security. Proceedings, 2001. XIII, 554 pages. 2001.

Vol. 2201: G.D. Abowd, B. Brumitt, S. Shafer (Eds.), Ubicomp 2001: Ubiquitous Computing. Proceedings, 2001. XIII, 372 pages. 2001.

Vol. 2202: A. Restivo, S. Ronchi Della Rocca, L. Roversi (Eds.), Theoretical Computer Science. Proceedings, 2001. XI, 440 pages. 2001.

Vol. 2204: A. Brandstädt, V.B. Le (Eds.), Graph-Theoretic Concepts in Computer Science. Proceedings, 2001. X, 329 pages. 2001.

Vol. 2205: D.R. Montello (Ed.), Spatial Information Theory. Proceedings, 2001. XIV, 503 pages. 2001.

Vol. 2206: B. Reusch (Ed.), Computational Intelligence. Proceedings, 2001. XVII, 1003 pages. 2001.

Vol. 2207: I.W. Marshall, S. Nettles, N. Wakamiya (Eds.), Active Networks. Proceedings, 2001. IX, 165 pages. 2001.

Vol. 2208: W.J. Niessen, M.A. Viergever (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2001. Proceedings, 2001. XXXV, 1446 pages. 2001.

Vol. 2209: W. Jonker (Ed.), Databases in Telecommunications II. Proceedings, 2001. VII, 179 pages. 2001.

Vol. 2210: Y. Liu, K. Tanaka, M. Iwata, T. Higuchi, M. Yasunaga (Eds.), Evolvable Systems: From Biology to Hardware. Proceedings, 2001. XI, 341 pages. 2001.

Vol. 2211: T.A. Henzinger, C.M. Kirsch (Eds.), Embedded Software. Proceedings, 2001. IX, 504 pages. 2001.

Vol. 2212: W. Lee, L. Mé, A. Wespi (Eds.), Recent Advances in Intrusion Detection. Proceedings, 2001. X, 205 pages. 2001.

Vol. 2213: M.J. van Sinderen, L.J.M. Nieuwenhuis (Eds.), Protocols for Multimedia Systems. Proceedings, 2001. XII, 239 pages. 2001.

Vol. 2215: N. Kobayashi, B.C. Pierce (Eds.), Theoretical Aspects of Computer Software. Proceedings, 2001. XV, 561 pages. 2001.

Vol. 2217: T. Gomi (Ed.), Evolutionary Robotics. Proceedings, 2001. XI, 139 pages. 2001.

Preface

The 27th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2001) was held in Boltenhagen (Mecklenburg-Vorpommern) June 14–16, 2001. It was organized by the Theoretical Computer Science group of the Department of Computer Science at the University of Rostock. The organizers gratefully acknowledge the support by the German Research Community (Deutsche Forschungsgemeinschaft - DFG) and the state Mecklenburg-Vorpommern.

As was the case for previous WG workshops, this workshop was devoted to the theoretical and practical aspects of graph concepts in computer science, and its contributed talks showed how recent research results from algorithmic graph theory can be used in computer science and which graph-theoretic questions arise from new developments in computer science. Moreover, the workshop gave an impression of the impact of computer science on the efficiency of graph algorithms and on structural aspects of graphs and graph decomposition.

The workshop looks back to a remarkable tradition of 27 years; all previous WG conferences took place in the middle of Europe (Germany, The Netherlands, Italy, Switzerland, Slowakia). The participants of WG 2001 came from various countries such as Canada, China, Czech Republic, France, Great Britain, Greece, Italy, Norway, Russia, Sweden, Syria, Taiwan, The Netherlands, U.S.A., and, of course, Germany.

The program committee represented the wide scientific spectrum and the aims of the conference. In a careful reviewing process with at least four reports per submission, the program committee selected 27 papers from the submissions. All accepted papers were presented at the conference, and the referees' comments as well as the numerous fruitful discussions during the workshop have been taken into account by the authors of these conference proceedings. Moreover, there were two fascinating invited talks by

> H.-J. Bandelt (University of Hamburg) on some connections between graph concepts such as median hulls and Steiner trees on one hand and molecular biology on the other hand,

and by

> F. Meyer auf der Heide (University of Paderborn) on data management in networks, giving a survey on strategies for distributing and accessing shared objects in large parallel and distributed systems.

It is our pleasure to thank all those who contributed to the scientific success of WG 2001:
- all authors of submitted and of presented papers, and in particular the speakers
- the referees and subreferees
- the invited speakers

- the DFG
- the state of Mecklenburg-Vorpommern, and last but not least
- the organizers Katrin Erdmann, Thomas Szymczak, Ernst de Ridder, supported by Roswitha Fengler before and Tilo Klembt, Suhail Mahfud, and Hans-Jörg Schulz during the conference.

August 2001 Andreas Brandstädt and Van Bang Le

The Tradition of WG

	Hosts – Location
1975	U. Pape – Berlin
1976	H. Noltemeier – Göttingen
1977	J. Mühlbacher – Linz
1978	M. Nagl, H.J. Schneider – Schloß Feuerstein, near Erlangen
1979	U. Pape – Berlin
1980	H. Noltemeier – Bad Honnef
1981	J. Mühlbacher – Linz
1982	H.J. Schneider, H. Göttler – Neunkirchen, near Erlangen
1983	M. Nagl, J. Perl – Haus Ohrbeck, near Osnabrück
1984	U. Pape – Berlin
1985	H. Noltemeier – Schloß Schwanenberg, near Würzburg
1986	G. Tinhofer, G. Schmidt – Stift Bernried, near München
1987	H. Göttler, H.J. Schneider – Schloß Banz, near Bamberg
1988	J. van Leeuwen – Amsterdam
1989	M. Nagl – Schloß Rolduc, near Aachen
1990	R.H. Möhring – Johannesstift Berlin
1991	G. Schmidt, R. Berghammer – Richterheim Fischbachau, München
1992	E.W. Mayr – Wilhelm-Kempf-Haus, Wiesbaden-Naurod
1993	J. van Leeuwen – Sports Center Papendal, near Utrecht
1994	G. Tinhofer, E.W. Mayr, G. Schmidt – Herrsching, near München
1995	M. Nagl – Haus Eich, Aachen
1996	G. Ausiello, A. Marchetti-Spaccamela – Cadenabbia
1997	R.H. Möhring – Bildungszentrum am Müggelsee, Berlin
1998	J. Hromkovič, O. Sýkora – Smolenice-Castle, near Bratislava
1999	P. Widmayer – Centro Stefano Franscini, Monte Verità, Ascona
2000	D. Wagner – Waldhaus Jakob, Konstanz
2001	A. Brandstädt – Boltenhagen, near Rostock

Program Committee

Hans Bodlaender	Utrecht University, The Netherlands
Andreas Brandstädt	University of Rostock, Germany (chair)
Victor Chepoi	Université de la Mediterranée, Marseille, France
Michel Habib	LIRMM Montpellier, France
Klaus Jansen	University of Kiel, Germany
Dieter Kratsch	Université de Metz, France
Luděk Kučera	Charles University, Prague, Czech Republic
Alberto Marchetti-Spaccamela	Università di Roma “La Sapienzia”, Italy
Ernst W. Mayr	TU München, Germany
Rolf H. Möhring	TU Berlin, Germany
Manfred Nagl	RWTH Aachen, Germany
Hartmut Noltemeier	University of Würzburg, Germany
Jeremy P. Spinrad	Vanderbilt University, Nashville, USA
Ondrej Sýkora	Loughborough University, United Kingdom
Gottfried Tinhofer	TU München, Germany
Peter Widmayer	ETH Zürich, Switzerland

Additional Reviewers

Gerardo Bandera
Thomas Bayer
Hans-J. Boeckenhauer
Peter Damaschke
Feodor F. Dragan
Stefan Dobrev
Thomas Erlebach
Jens Ernst
Jiří Fiala
Stefanie Gerke
Yubao Guo
Alexander Hall
Volker Heun
Klaus Holzapfel
Juraj Hromkovič
Sandi Klavžar
Ekkehard Köhler
Christian Laforest
Jean-Marc Lanlignel
Van Bang Le
Zsuzsanna Lipták
Sonia P. Mansilla
Ross McConnell
Michal Mnuk
Haiko Müller
Takao Nishizeki
Conrad Pomm
Pascal Prea
Martin Raab
Ingo Rohloff
Udi Rotics
Mark Scharbrodt
Konrad Schlude
Heiko Schröder
Martin Skutella
Ladislav Stacho
Robert Szelepcsenyi
Thomas Szymczak
Ioan Todinca
Yann Vaxes
Ulrich Voll
Imrich Vrt'o
Karsten Weihe

Sponsoring Institutions

Deutsche Forschungsgemeinschaft (DFG)
Land Mecklenburg-Vorpommern

Table of Contents

Median Hulls as Steiner Hulls in Rectilinear and Molecular Sequence Spaces (Invited Presentation)

Hans-J. Bandelt

Departement of Mathematics, University of Hamburg, Germany
bandelt@math.uni-hamburg.de

1 Introduction

The *Steiner problem* is to find a shortest connection of a finite subset X in a given metric space (M, d). If the space meets some local compactness and connectivity conditions, then a solution, a *Steiner minimal tree* for X, exists [7]. More generally, considering any graph-theoretic tree T with all nodes of degree < 3 labeled by elements of X (that is, an X*-labeled tree*), we may ask for a minimal length realization of T in (M, d), that is, for an embedding of the node set of T in M which extends the identity map of X and yields the smallest possible length of the image Steiner tree (relative to the labeled tree T). The embedded unlabeled nodes of T are called *Steiner points.* A Steiner minimal tree is then that realization which attains minimum length among the (finite) collection of X-labeled trees. Given the subset X, one is interested in restricting the search for Steiner points in the metric space (M, d). Any proper subset of M that is guaranteed to harbor at least one Steiner minimal tree for X is called a *Steiner hull* for X [7, 12]. Particular interest attaches to Steiner hulls that are finitely generated in that they are determined by a finite subset S of M containing X and all Steiner points of some Steiner minimal tree for X; this finite set S can trivially be turned into a Steiner hull by attaching one geodesic from M for each pair of its points. For the Steiner problem in rectilinear space, that is, a d-dimensional real space equipped with the metric associated with the 1-norm, finitely generated Steiner hulls have been described [7, 12].

One of the motivations to study the Steiner problem in rectilinear space comes from biology, where one is interested in reconstructing the phylogeny or genealogy of species or populations based on molecular or morphological data. The Steiner problem is there referred to as the parsimony problem and Steiner minimal trees are then called most parsimonious reconstructions of most parsimonious trees. Actually, Farris [8] was the first to describe an algorithm for finding a minimal length realization of any X-labeled tree within a rectilinear space (an "ordered character" in the biological context refers to one coordinate of the space). This was accordingly modified for molecular sequence spaces by Fitch [9]; cf. [11]. Such a space (equipped with the Hamming distance) is, for some $d > 0$, the d-th Cartesian power of an equidistant alphabet (typically, the nucleotide alphabet). In the binary case, that is, say, for the alphabet $\{0, 1\}$, it

A. Brandstädt and V.B. Le (Eds.): WG 2001, LNCS 2204, pp. 1–7, 2001.

is knowm that median hulls generate Steiner hulls [2, 14]. In fact, an algebraic analysis of the algorithm of Farris and Fitch (the FFH algorithm, see below) reveals that any X-labeled tree T has a minimal length realization as a Steiner tree relative to T using only Steiner points from the median hull of X. In practical applications where this hull would still be too large, one could further reduce this set in a heuristic fashion, of course at the expense that the Steiner hull property may no longer provably be maintained [1, 4].

2 Reduction to Hypercubes

Consider a tree T interconnecting a family of sequences from a finite subset X of the d-dimensional rectilinear space. Then one can realize T as a Steiner tree in the rectilinear space attaining minimal length relative to T [7]. Adjoin to X all Steiner points of this Steiner tree and extend the resulting set to a finite grid Y (of dimension at most d). This grid can be recoded by 0-1 sequences and thus be embedded into a finite-dimensional hypercube H in which every coordinate receives a certain weight. We can think of X being embedded in H such that a Steiner tree realization of T in H has the same minimal length as in the original rectilinear space. We will see below that the adjunction of a set of Steiner points was an unnecessary proviso since the median hull of X (necessarily contained in Y) turns out to harbor an ample collection of potential Steiner points to guarantee a minimal length realization relative to any prescribed tree T.

3 Farris-Fitch-Hartigan (FFH) Labeling

First, in order to describe and analyze the FFH algorithm, it is convenient to encode any given tree T in terms of its leaves $t_0, t_1, \ldots t_{n-1}$ $(n > 2)$. One leaf, say t_0, is selected as a root. In dealing with the thus rooted tree T we employ the metaphor of a maternal genealogy and think of T as being visualized in the plane such that t_0 is at the top and the other leaves are at the bottom. The rooted subtree $T' := T - t_0$ obtained by deleting t_0 (and the edge incident with it) has the (unique) daughter node u_0 of t_0 as its root. The so-called New Hampshire encoding [10] of T' is a string comprising brackets and the leaf names $t_1, \ldots, t_{n-1}$. The code for T' is obtained bottom-up: if in the first step, say, $t_1, \ldots, t_i$ $(i > 1)$ comprise all the daughter nodes of their common mother node u, then we record this by $u = (t_1, \ldots, t_i)$ where the order within the bracket is arbitrary. Then delete $t_1, \ldots, t_i$ and turn u into a leaf, which represents the subtree with original leaves $t_1, \ldots, t_i$. Continue until the bracketing stops with a single node, the root u_0 of T'. To arrive at the final code, successively replace every interior node by the associated bracket.

We now turn the New Hampshire encodings of the subtrees into operations on the hypercube. First we deal with the case of a single coordinate which may take values 0, 1. In order to handle ties , we introduce a third letter N, which stands for "not determined". For each $k > 1$ we define the k-ary majority operation on $\{0, 1, \mathrm{N}\}$ as follows. The operation returns 0 if 0 occurs more often than 1 among the k entries, it reurns 1 if 1 is more frequent than 0, and returns N otherwise, that is , in the case of a tie between 0 and 1. The outcome is then simply written as a bracket with the entries listed as a string. By definition, this operation is commutative. For $k = 2$ we have (00) = (0N) = 0, (11) = (1N) = 1, and (01) = (NN) = N, and for $k = 3$ we have (000) = (001) = (00N) = (0NN) = 0, (111) = (110) = (11N) = (1NN) = 1, and (01N) = (NNN) = N. Restricted to $\{0, 1\}$ the latter entails the median operation on $\{0, 1\}$ [3]. The k-ary bracket operation on the set $\{0, 1, \mathrm{N}\}^d$ of sequences of length d is then defined coordinate-wise.

The FFH algorithm makes two passes through the tree T rooted at t_0, a bottom-up pass and then a top-down pass [10, 13]. Starting from the labeled leaves $t_1, \ldots, t_{n-1}$, any mother node with k children labeled by $x_1, \ldots, x_k$ receives the label $(x_1 \ldots x_k)$. Proceeding bottom-up this pass stops with the labeling of the root u_0 of T'. In the top-down pass, beginning with u_0, the undetermined ("N") coordinates are successively specified by 0 or 1: a node of T with current label x such that the mother node is already labeled by a 0-1 sequence y receives the 0-1 sequence (xxy) as its new label. This, in brief, is how the algorithm works. The (well-known) proof that this labeling yields a minimal length realization relative to T is straightforward by induction: interpret each label of the bottom-up pass as a set of 0-1 sequences obtained by all combinations when inserting 0 and 1 for N; this set describes exactly the 0-1 sequences at the interior node under consideration each leading to a minimal length realization of the subtree rooted at this node; the final top-down pass then executes the minimization link-wise by using the specified 0-1 sequence at the mother node in question.

4 Examples

First consider the set $X = \{v, w, x, y\}$ of length 7 sequences $v = 1000111$, $w = 0100100$, $x = 0010010$, $y = 0001001$. Let T be the tree with four leaves t_0, t_1, t_2, t_3 and interior edge separating t_0, t_1 from t_2, t_3. Then $T' := T - t_0 = (t_1, (t_2, t_3))$. When we label t_0, t_1, t_2, t_3 by v, w, x, y, respectively, then the root of T' receives the label $(vw(wxy)) = 0000100$ under FFH and the second interior node of T is then labeled by $(wxy) = 0000000$. When we assign to the leaves t_0, t_1, t_2, t_3 either w, v, x, y or x, y, v, w, or y, x, v, w instead, we obtain six further sequences, so that eventually all 0-1 sequences of length 7 with prefix 0000 are generated. The resulting set of 12 sequences (including X) is then closed under the median operation and constitutes the graph $\lambda(4)$ described in [5].

Next we consider the set $X = \{v, w, x, y, z\}$ of length 15 sequences where the coordinates correspond to the partitions of X into two parts such that 0 is always assigned to the majority part (for convenience). Assume that T is a tree with five leaves t_0, t_1, t_2, t_3, t_4 which are labeled by v, w, x, y, z, respectively, unless stated otherwise. Let t_0 serve as the root of T, and put $T' := T - t_0$. Since there are two binary rooted trees with four leaves, we distinguish two cases.

Case 1: $T' = ((t_1,t_2),(t_3,t_4))$. Then the root of T' receives the label

$$(vwxyz) = (v((vwx)yz)((vyz)wx))$$

by the FFH algorithm. Its daughter node that is the mother node of t_3 and t_4 then gets the label

$$(vwxyyzz) = ((vwx)yz).$$

When we permute the labels, the root label always remains the same, but the mother node of t_3 and t_4 can receive nine further possible labels.

Case 2: $T' = (((t_1,t_2),t_3),t_4)$. Then via FFH the root of T' carries the label

$$(vvwxyyzzz) = (((wxy)yz)vz),$$

its interior daughter node

$$(wxyyz) = ((wxy)yz),$$

and the interior granddaughter node

$$(wxy).$$

Reshuffling the labels will produce altogether 30 label candidates at the root, 20 at its interior daughter node, and 10 at its interior granddaughter node.

In either case and under each permutation of labels, we obtain a Steiner minimal tree (of lenghth 23), where each time the FFH algorithm selects three Steiner points from a pool of 71 sequences.

The FFH algorithm can in fact contruct further sequences from X when one prescribes a tree with more than five leaves (and thus multiple occurrences of labels), which would no longer lead to a Steiner minimal tree for X. Specifically, pick a tree with six leaves t_i $(i = 0,\dots,5)$ such that $T - t_0 = ((((t_1,t_2),t_3),t_4),t_5)$ and t_0,t_1,t_2,t_3,t_4,t_5 are labeled by z,w,x,v,y,v, respectively. Then the root of $T - t_0$ receives

$$(vvvwxyz) = (((vwx)vy)vz).$$

Permutation of the labels gives four additional sequences. The original five sequences of X together with the former 71 Steiner points and the latter five sequences form a set closed under the median operation and constitute the nodes of the graph $\lambda(5)$, see [5]. We will see below that this set exhausts all possible interior labels obtainable via FFH from any tree with leaves labeled by sequences of X.

5 Median Hull Includes FFG Hull

The median hull of a finite subset X of a rectilinear space or a hypercube is the smallest (necessarily finite) set containing X and closed under the median operation which assigns to any triplet u,v,w its median (uvw). This hull can be constructed by successively adding medians until no further new points (sequences) arise.

Now, regarding X as a subset of the k-dimensional hypercube H. One can easily check whether a particular sequence $a_1 \dots a_k$ from H belongs to the median hull of X: a necessary and sufficient condition is that each truncation (alias projection) $a_i a_j$ of the sequence to two coordinates belongs to the corresponding truncation of X, that is, agrees at coordinates i and j with some sequence of X [6]. To see this, apply induction on the sequence length. Assume that $a_1 a_2 \dots a_{k-1} a_k$ meets this criterion on coordinate pairs. Then by the induction hypothesis the two truncated sequences $a_1 a_2 \dots a_{k-1}$ and $a_2 \dots a_{k-1} a_k$ belong to the median hull of the corresponding truncation of X, and the length 2 sequence $a_1 a_k$ belongs to the truncation of X to coordinates $1, k$. Hence there exist $b_1, \dots, b_k$ from $\{0, 1\}$ such that $a_1 a_2 \dots a_{k-1} b_k$, $b_1 a_2 \dots a_{k-1} a_k$, $a_1 b_2 \dots b_{k-1} a_k$ belong to the median hull of X. Since $(a_i a_i b_i) = a_i$ for all i, we obtain $a_1 \dots a_k$ as the median of these three sequences, as required.

In order to show that the FFH algorithm produces only labels from the median hull of X, we may therefore assume that the leaves of the leaf-rooted tree T under consideration are labeled by 0-1 sequences of length 2. If all four sequences occur at the leaves, then there is nothing left to show. So, assume that no leaf receives label 11, say. We claim that after the bottom-up phase 1N, N1 and 11 never label any interior node. Argue by induction from the bottom-upwards. Assume that the interior node under consideration has 1 as the first label coordinate. This means that more daughter nodes have 1 than 0 at the first coordinate. Then by the induction hypothesis the former daughter nodes must have 0 at the second coordinate, whereas only the latter daughter nodes may have 1 as the second coordinate. Hence the mother node necessarily receives label 10, which is fine. Finally, in the top-down phase the feasible labels 0N and N0 are specified as either 00 or 01 and 10, respectively, while NN is replaced by the label at the mother node, which is different from 11 (by employing a top-down induction argument).

We conclude that all labels erected by the FFH algorithm lie in the median hull of X.

6 FFH Hull Includes Median Hull

Now that we know that the FFH labeling stays within the median hull of the given set X of binary sequences we may ask whether it actually exhausts the latter: can every sequence in the median hull of X be obtained as a FFH label of some interior node of a suitable tree T rooted at some leaf t_0? Yes, of course. To see this, use induction on the number of median operations employed in generating a particular sequence v of the median hull of X. Assume that $v = (u_1 u_2 u_3)$ where each u_i is the FFH label of the daughter node of a leaf-root t_i of some tree T_i, independent of the label from X attached to t_i. In other words, the FFH algorithm already fixes the labels (as a 0-1 sequence) at the root of the tree $T_i' := T_i - t_i$ in the bottom-up phase. Stipulating that all rooted trees