LNCS 3122

Klaus Jansen · Sanjeev Khanna
José D. P. Rolim · Dana Ron (Eds.)

Approximation, Randomization, and Combinatorial Optimization

Algorithms and Techniques

7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2004 and 8th International Workshop on Randomization and Computation, RANDOM 2004
Cambridge, MA, USA, August 2004, Proceedings

Springer

Klaus Jansen Sanjeev Khanna
José D.P. Rolim Dana Ron (Eds.)

Approximation, Randomization, and Combinatorial Optimization

Algorithms and Techniques

7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2004
and 8th International Workshop on Randomization and Computation, RANDOM 2004
Cambridge, MA, USA, August 22-24, 2004
Proceedings

Springer

Volume Editors

Klaus Jansen
Christian-Albrechts-University of Kiel
Institute of Computer Science and Applied Mathematics
Olshausenstr. 40, 24098 Kiel, Germany
E-mail: kj@informatik.uni-kiel.de

Sanjeev Khanna
University of Pennsylvania, Dept. of Computer and Information Science
3330 Walnut Street, Philadelphia, PA 19104, USA
E-mail: sanjeev@cis.upenn.edu

José D.P. Rolim
University of Geneva, Computer Science Department
24 rue Général Dufour, 1211 Geneva 4, Switzerland
E-mail: jose.rolim@cui.unige.ch

Dana Ron
Harvard University, Radcliffe Institute for Advanced Study
10 Garden St, Cambridge, MA 02138, USA
E-mail: danar@eecs.harvard.edu

Library of Congress Control Number: 2004110448

CR Subject Classification (1998): F.2, G.2, G.1

ISSN 0302-9743
ISBN 3-540-22894-2 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 11311911 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3122

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Lecture Notes in Computer Science

For information about Vols. 1–3062

please contact your bookseller or Springer

Vol. 3194: R. Camacho, R. King, A. Srinivasan (Eds.), Inductive Logic Programming. XI, 361 pages. 2004. (Subseries LNAI).

Vol. 3177: Z.R. Yang, H. Yin, R. Everson (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2004. VXIII, 852 pages. 2004.

Vol. 3174: F. Yin, J. Wang, C. Guo (Eds.), Advances in Neural Networks - ISNN 2004. XXXV, 1021 pages. 2004.

Vol. 3172: M. Dorigo, M. Birattari, C. Blum, L. M.Gambardella, F. Mondada, T. Stützle (Eds.), Ant Colony, Optimization and Swarm Intelligence. XII, 434 pages. 2004.

Vol. 3158: I. Nikolaidis, M. Barbeau, E. Kranakis (Eds.), Ad-Hoc, Mobile, and Wireless Networks. IX, 344 pages. 2004.

Vol. 3157: C. Zhang, H. W. Guesgen, W.K. Yeap (Eds.), PRICAI 2004: Trends in Artificial Intelligence. XX, 1023 pages. 2004. (Subseries LNAI).

Vol. 3156: M. Joye, J.-J. Quisquater (Eds.), Cryptographic Hardware and Embedded Systems - CHES 2004. XIII, 455 pages. 2004.

Vol. 3155: P. Funk, P.A. González Calero (Eds.), Advanced in Case-Based Reasoning. XIII, 822 pages. 2004. (Subseries LNAI).

Vol. 3153: J. Fiala, V. Koubek, J. Kratochvíl (Eds.), Mathematical Foundations of Computer Science 2004. XIV, 902 pages. 2004.

Vol. 3152: M. Franklin (Ed.), Advances in Cryptology – CRYPTO 2004. XI, 579 pages. 2004.

Vol. 3150: G.-Z. Yang, T. Jiang (Eds.), Medical Imaging and Augmented Reality. XII, 378 pages. 2004.

Vol. 3148: R. Giacobazzi (Ed.), Static Analysis. XI, 393 pages. 2004.

Vol. 3146: P. Érdi, A. Esposito, M. Marinaro, S. Scarpetta (Eds.), Computational Neuroscience: Cortical Dynamics. XI, 161 pages. 2004.

Vol. 3144: M. Papatriantafilou, P. Hunel (Eds.), Principles of Distributed Systems. XI, 246 pages. 2004.

Vol. 3143: W. Liu, Y. Shi, Q. Li (Eds.), Advances in Web-Based Learning – ICWL 2004. XIV, 459 pages. 2004.

Vol. 3142: J. Diaz, J. Karhumäki, A. Lepistö, D. Sannella (Eds.), Automata, Languages and Programming. XIX, 1253 pages. 2004.

Vol. 3140: N. Koch, P. Fraternali, M. Wirsing (Eds.), Web Engineering. XXI, 623 pages. 2004.

Vol. 3139: F. Iida, R. Pfeifer, L. Steels, Y. Kuniyoshi (Eds.), Embodied Artificial Intelligence. IX, 331 pages. 2004. (Subseries LNAI).

Vol. 3138: A. Fred, T. Caelli, R.P.W. Duin, A. Campilho, D.d. Ridder (Eds.), Structural, Syntactic, and Statistical Pattern Recognition. XXII, 1168 pages. 2004.

Vol. 3136: F. Meziane, E. Métais (Eds.), Natural Language Processing and Information Systems. XII, 436 pages. 2004.

Vol. 3134: C. Zannier, H. Erdogmus, L. Lindstrom (Eds.), Extreme Programming and Agile Methods - XP/Agile Universe 2004. XIV, 233 pages. 2004.

Vol. 3133: A.D. Pimentel, S. Vassiliadis (Eds.), Computer Systems: Architectures, Modeling, and Simulation. XIII, 562 pages. 2004.

Vol. 3131: V. Torra, Y. Narukawa (Eds.), Modeling Decisions for Artificial Intelligence. XI, 327 pages. 2004. (Subseries LNAI).

Vol. 3130: A. Syropoulos, K. Berry, Y. Haralambous, B. Hughes, S. Peter, J. Plaice (Eds.), TeX, XML, and Digital Typography. VIII, 265 pages. 2004.

Vol. 3129: Q. Li, G. Wang, L. Feng (Eds.), Advances in Web-Age Information Management. XVII, 753 pages. 2004.

Vol. 3128: D. Asonov (Ed.), Querying Databases Privately. IX, 115 pages. 2004.

Vol. 3127: K.E. Wolff, H.D. Pfeiffer, H.S. Delugach (Eds.), Conceptual Structures at Work. XI, 403 pages. 2004. (Subseries LNAI).

Vol. 3126: P. Dini, P. Lorenz, J.N.d. Souza (Eds.), Service Assurance with Partial and Intermittent Resources. XI, 312 pages. 2004.

Vol. 3125: D. Kozen (Ed.), Mathematics of Program Construction. X, 401 pages. 2004.

Vol. 3124: J.N. de Souza, P. Dini, P. Lorenz (Eds.), Telecommunications and Networking - ICT 2004. XXVI, 1390 pages. 2004.

Vol. 3123: A. Belz, R. Evans, P. Piwek (Eds.), Natural Language Generation. X, 219 pages. 2004. (Subseries LNAI).

Vol. 3122: K. Jansen, S. Khanna, J.D.P. Rolim, D. Ron (Eds.), Approximation, Randomization, and Combinatorial Optimization. IX, 428 pages. 2004.

Vol. 3121: S. Nikoletseas, J.D.P. Rolim (Eds.), Algorithmic Aspects of Wireless Sensor Networks. X, 201 pages. 2004.

Vol. 3120: J. Shawe-Taylor, Y. Singer (Eds.), Learning Theory. X, 648 pages. 2004. (Subseries LNAI).

Vol. 3118: K. Miesenberger, J. Klaus, W. Zagler, D. Burger (Eds.), Computer Helping People with Special Needs. XXIII, 1191 pages. 2004.

Vol. 3116: C. Rattray, S. Maharaj, C. Shankland (Eds.), Algebraic Methodology and Software Technology. XI, 569 pages. 2004.

Vol. 3114: R. Alur, D.A. Peled (Eds.), Computer Aided Verification. XII, 536 pages. 2004.

Vol. 3113: J. Karhumäki, H. Maurer, G. Paun, G. Rozenberg (Eds.), Theory Is Forever. X, 283 pages. 2004.

Vol. 3112: H. Williams, L. MacKinnon (Eds.), Key Technologies for Data Management. XII, 265 pages. 2004.

Vol. 3111: T. Hagerup, J. Katajainen (Eds.), Algorithm Theory - SWAT 2004. XI, 506 pages. 2004.

Vol. 3110: A. Juels (Ed.), Financial Cryptography. XI, 281 pages. 2004.

Vol. 3109: S.C. Sahinalp, S. Muthukrishnan, U. Dogrusoz (Eds.), Combinatorial Pattern Matching. XII, 486 pages. 2004.

Vol. 3108: H. Wang, J. Pieprzyk, V. Varadharajan (Eds.), Information Security and Privacy. XII, 494 pages. 2004.

Vol. 3107: J. Bosch, C. Krueger (Eds.), Software Reuse: Methods, Techniques and Tools. XI, 339 pages. 2004.

Vol. 3106: K.-Y. Chwa, J.I. Munro (Eds.), Computing and Combinatorics. XIII, 474 pages. 2004.

Vol. 3105: S. Göbel, U. Spierling, A. Hoffmann, I. Iurgel, O. Schneider, J. Dechau, A. Feix (Eds.), Technologies for Interactive Digital Storytelling and Entertainment. XVI, 304 pages. 2004.

Vol. 3104: R. Kralovic, O. Sykora (Eds.), Structural Information and Communication Complexity. X, 303 pages. 2004.

Vol. 3103: K. Deb, e. al. (Eds.), Genetic and Evolutionary Computation – GECCO 2004. XLIX, 1439 pages. 2004.

Vol. 3102: K. Deb, e. al. (Eds.), Genetic and Evolutionary Computation – GECCO 2004. L, 1445 pages. 2004.

Vol. 3101: M. Masoodian, S. Jones, B. Rogers (Eds.), Computer Human Interaction. XIV, 694 pages. 2004.

Vol. 3100: J.F. Peters, A. Skowron, J.W. Grzymała-Busse, B. Kostek, R.W. Świniarski, M.S. Szczuka (Eds.), Transactions on Rough Sets I. X, 405 pages. 2004.

Vol. 3099: J. Cortadella, W. Reisig (Eds.), Applications and Theory of Petri Nets 2004. XI, 505 pages. 2004.

Vol. 3098: J. Desel, W. Reisig, G. Rozenberg (Eds.), Lectures on Concurrency and Petri Nets. VIII, 849 pages. 2004.

Vol. 3097: D. Basin, M. Rusinowitch (Eds.), Automated Reasoning. XII, 493 pages. 2004. (Subseries LNAI).

Vol. 3096: G. Melnik, H. Holz (Eds.), Advances in Learning Software Organizations. X, 173 pages. 2004.

Vol. 3095: C. Bussler, D. Fensel, M.E. Orlowska, J. Yang (Eds.), Web Services, E-Business, and the Semantic Web. X, 147 pages. 2004.

Vol. 3094: A. Nürnberger, M. Detyniecki (Eds.), Adaptive Multimedia Retrieval. VIII, 229 pages. 2004.

Vol. 3093: S.K. Katsikas, S. Gritzalis, J. Lopez (Eds.), Public Key Infrastructure. XIII, 380 pages. 2004.

Vol. 3092: J. Eckstein, H. Baumeister (Eds.), Extreme Programming and Agile Processes in Software Engineering. XVI, 358 pages. 2004.

Vol. 3091: V. van Oostrom (Ed.), Rewriting Techniques and Applications. X, 313 pages. 2004.

Vol. 3089: M. Jakobsson, M. Yung, J. Zhou (Eds.), Applied Cryptography and Network Security. XIV, 510 pages. 2004.

Vol. 3087: D. Maltoni, A.K. Jain (Eds.), Biometric Authentication. XIII, 343 pages. 2004.

Vol. 3086: M. Odersky (Ed.), ECOOP 2004 – Object-Oriented Programming. XIII, 611 pages. 2004.

Vol. 3085: S. Berardi, M. Coppo, F. Damiani (Eds.), Types for Proofs and Programs. X, 409 pages. 2004.

Vol. 3084: A. Persson, J. Stirna (Eds.), Advanced Information Systems Engineering. XIV, 596 pages. 2004.

Vol. 3083: W. Emmerich, A.L. Wolf (Eds.), Component Deployment. X, 249 pages. 2004.

Vol. 3080: J. Desel, B. Pernici, M. Weske (Eds.), Business Process Management. X, 307 pages. 2004.

Vol. 3079: Z. Mammeri, P. Lorenz (Eds.), High Speed Networks and Multimedia Communications. XVIII, 1103 pages. 2004.

Vol. 3078: S. Cotin, D.N. Metaxas (Eds.), Medical Simulation. XVI, 296 pages. 2004.

Vol. 3077: F. Roli, J. Kittler, T. Windeatt (Eds.), Multiple Classifier Systems. XII, 386 pages. 2004.

Vol. 3076: D. Buell (Ed.), Algorithmic Number Theory. XI, 451 pages. 2004.

Vol. 3075: W. Lenski (Ed.), Logic versus Approximation. IX, 205 pages. 2004.

Vol. 3074: B. Kuijpers, P. Revesz (Eds.), Constraint Databases and Applications. XII, 181 pages. 2004.

Vol. 3073: H. Chen, R. Moore, D.D. Zeng, J. Leavitt (Eds.), Intelligence and Security Informatics. XV, 536 pages. 2004.

Vol. 3072: D. Zhang, A.K. Jain (Eds.), Biometric Authentication. XVII, 800 pages. 2004.

Vol. 3071: A. Omicini, P. Petta, J. Pitt (Eds.), Engineering Societies in the Agents World. XIII, 409 pages. 2004. (Subseries LNAI).

Vol. 3070: L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L.A. Zadeh (Eds.), Artificial Intelligence and Soft Computing - ICAISC 2004. XXV, 1208 pages. 2004. (Subseries LNAI).

Vol. 3068: E. André, L. Dybkjær, W. Minker, P. Heisterkamp (Eds.), Affective Dialogue Systems. XII, 324 pages. 2004. (Subseries LNAI).

Vol. 3067: M. Dastani, J. Dix, A. El Fallah-Seghrouchni (Eds.), Programming Multi-Agent Systems. X, 221 pages. 2004. (Subseries LNAI).

Vol. 3066: S. Tsumoto, R. Słowiński, J. Komorowski, J.W. Grzymała-Busse (Eds.), Rough Sets and Current Trends in Computing. XX, 853 pages. 2004. (Subseries LNAI).

Vol. 3065: A. Lomuscio, D. Nute (Eds.), Deontic Logic in Computer Science. X, 275 pages. 2004. (Subseries LNAI).

Vol. 3064: D. Bienstock, G. Nemhauser (Eds.), Integer Programming and Combinatorial Optimization. XI, 445 pages. 2004.

Vol. 3063: A. Llamosí, A. Strohmeier (Eds.), Reliable Software Technologies - Ada-Europe 2004. XIII, 333 pages. 2004.

Foreword

This volume contains the papers presented at the *7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems* (APPROX 2004) and the *8th International Workshop on Randomization and Computation* (RANDOM 2004), which took place concurrently at Harvard University, Cambridge, on August 22–24, 2004. APPROX focuses on algorithmic and complexity issues surrounding the development of efficient approximate solutions to computationally hard problems, and this year's workshop was the seventh in the series after Aalborg (1998), Berkeley (1999), Saarbrücken (2000), Berkeley (2001), Rome (2002), and Princeton (2003). RANDOM is concerned with applications of randomness to computational and combinatorial problems, and this year's workshop was the eighth in the series following Bologna (1997), Barcelona (1998), Berkeley (1999), Geneva (2000), Berkeley (2001), Harvard (2002), and Princeton (2003).

Topics of interest for APPROX and RANDOM are: design and analysis of approximation algorithms, inapproximability results, approximation classes, online problems, small space and data streaming algorithms, sub-linear time algorithms, embeddings and metric space methods in approximation, math progamming in approximation algorithms, coloring and partitioning, cuts and connectivity, geometric problems, network design and routing, packing and covering, scheduling, game theory, design and analysis of randomized algorithms, randomized complexity theory, pseudorandomness and derandomization, random combinatorial structures, random walks/Markov chains, expander graphs and randomness extractors, probabilistic proof systems, random projections and embeddings, error-correcting codes, average-case analysis, property testing, computational learning theory, and other applications of approximation and randomness.

The volume contains 19+18 contributed papers, selected by the two program committees from 54+33 submissions received in response to the call for papers.

We would like to thank all of the authors who submitted papers, the members of the program committees

APPROX 2004

Chandra Chekuri, Bell Laboratories
Lisa Fleischer, Carnegie Mellon U. and IBM T.J. Watson
Sudipto Guha, U. of Pennsylvania
Sanjeev Khanna, U. of Pennsylvania (Chair)
Rajmohan Rajaraman, Northeastern U.
Tim Roughgarden, UC Berkeley
Baruch Schieber, IBM T.J. Watson
Martin Skutella, Max Planck Institute
Dan Spielman, MIT
Luca Trevisan, UC Berkeley
Mihalis Yannakakis, Columbia U.
Neal Young, UC Riverside

RANDOM 2004

Noga Alon, Tel Aviv U.
Amos Beimel, Ben Gurion U.
Peter Bro Miltersen, U. of Aarhus
Funda Ergun, Case Western Reserve U.
Uri Feige, Weizmann Institute
Leslie Ann Goldberg, U. of Warwick
Russell Impagliazzo, UC San Diego
Adam Kalai, Toyota Technological Institute
Satish Rao, UC Berkeley
Dana Ron, Tel Aviv U. and Radcliffe Institute, Harvard (Chair)
Rocco Servedio, Columbia U.
Neal Young, UC Riverside

and the external subreferees Udo Adamy, Dorit Aharonov, Ernst Althaus, Javed Aslam, Nikhil Bansal, Therese Beidl, Petra Berenbrink, Joan Boyar, Gruia Calinescu, Kamalika Chaudhuri, Bernard Chazelle, Jiangzhuo Chen, Marek Chrobak, Colin Cooper, Erik Demaine, Nikhil Devanur, Thomas Erlebach, Martin Farach-Colton, Sándor Fekete, Tom Friedetzky, Rajiv Gandhi, Mordecai Golin, Oded Goldreich, Alex Hall, Refael Hassin, Stefan Hougardy, Piotr Indyk, Wojtek Jawor, Mark Jerrum, Lujun Jia, Valentine Kabanets, Marek Karpinski, Tali Kaufman, Julia Kempe, Claire Kenyon, Tracy Kimbrel, Robert Kleinberg, Adam Klivans, Jochen Könemann, Stavros Kolliopoulos, Madhular Korupolu, Robert Krauthgamer, Kofi Laing, Matt Levine, Moshe Lewenstein, Guolong Lin, Yi-Kai Liu, Stefano Lonardi, Marco Lübbecke, Mohammad Mahdian, Russ Martin, Daniele Micciancio, Mike Molloy, Cris Moore, Elchanan Mossel, Ian Munro, Assaf Naor, Moni Naor, Kirk Pruhs, Harald Räcke, Balaji Raghavachari, Vijaya Ramachandran, Dana Randall, April Rasala, Mauricio Resende, Gaby Scalosub, Guido Schäfer, Amir Shpilka, Dana Shapira, Riccardo Silvestri, Dan Spielman, Nicolas Stier, Leen Stougie, Martin Strauss, Maxim Sviridenko, Hisao Tamaki, Prasad Tetali, Santosh Vempala, Eric Vigoda, Anil Vullikanti, Dan Wang, Enav Weinreb, Gerhard Woeginger, Nick Wormald, Alex Zelikovsky and Yan Zhang.

We gratefully acknowledge support from the Radcliffe Institute for Advanced Study and the Computer Science Department of Harvard University, Cambridge, MA, the Institute of Computer Science of the Christian-Albrechts-Universität zu Kiel and the Department of Computer Science of the University of Geneva. We also thank Ute Iaquinto and Parvaneh Karimi Massouleh for their help.

August 2004

Sanjeev Khanna and Dana Ron, Program Chairs
Klaus Jansen and José D.P. Rolim, Workshop Chairs

Table of Contents

Contributed Talks of APPROX

Contributed Talks of RANDOM

Designing Networks with Existing Traffic to Support Fast Restoration

Mansoor Alicherry[1], Randeep Bhatia[1], and Yung-Chun (Justin) Wan[2]

[1] Bell Labs, Lucent Technologies, Murray Hill, NJ 07974
{mansoor,randeep}@research.bell-labs.com
[2] Department of Computer Science, University of Maryland, College Park, MD 20742
ycwan@cs.umd.edu

Abstract. We study a problem motivated by a scheme for supporting fast restoration in MPLS and optical networks. In this local restoration scheme detour paths are set-up a priori and network resources are pre-reserved exclusively for carrying rerouted traffic under network failures. (i.e. they do not carry any traffic under normal working conditions). The detours are such that failed links can be bypassed locally from the first node that is upstream from the failures. This local bypass activation from the first detection point for failures along with the dedication of network resources for handling failures permits very fast recovery times, a critical requirement for these networks. By allowing sharing of the dedicated resources among different detours the local restoration scheme results in efficient utilization of the pre-reserved network capacity.

In this paper we are interested in the problem of dedicating the least amount of the currently available network capacity for protection, while guaranteeing fast restoration to the existing traffic along with any traffic that may be admitted in the future. We show that the problem is NP-hard, and give a 2-approximation algorithm for the problem. We also show that the integrality gap of a natural relaxation of our problem is $\Omega(n)$, thus establishing that any LP-based approach using this relaxation cannot yield a better approximation algorithm for our problem.

1 Introduction

Dynamic provisioning of bandwidth guaranteed paths with fast restoration capability is an important network service feature for the emerging Multi-Protocol Label Switched (MPLS) networks [7] and optical mesh networks [18]. The fast restoration capabilities are required in order to provide the needed reliability for services such as packetized voice, critical VPN traffic, etc. Traditionally ring based SONET [11] networks have offered 50ms restoration to bandwidth guaranteed services, using pre-reserved spare protection capacity and pre-planned protection paths. Pre-planning protection in rings has been especially attractive, because of the availability of exactly one backup path between any two nodes, leading to very simple and fast automatic protection switching mechanisms. However in ring based SONET networks these advantages come at the cost of reserving at least half the total capacity for protection.

A local restoration scheme [12], [16], [20] is proposed to provide fast restoration in mesh based MPLS and optical networks. In this scheme which is also referred to as

K. Jansen et al. (Eds.): APPROX and RANDOM 2004, LNCS 3122, pp. 1–12, 2004.

link restoration the traffic on each link e of the network is protected by a detour path that does not include link e. Upon failure of any link e, any traffic on e is switched to its detour path. Thus, link restoration provides a local mechanism to route around a failure. In this restoration scheme the restoration capacity of the pre-setup detours is not used under normal no-failure conditions (except possibly by low priority preemptible traffic). Local restoration when used in conjunction with advanced reservation of the restoration capacities and pre-setup detours results in low restoration latency. Pre-provisioned link restoration also results in operational simplicity since the detours have to be only provisioned once for a given network topology and since the online connection routing can now be done oblivious to the reliability requirements, using only the resources that are not reserved for restoration.

An important consideration for any fast restoration scheme is to minimize the network resources dedicated for restoration and hence to maximize the proportion of network resources available for carrying traffic under normal working conditions. In general, link restoration provides guaranteed protection against only single link failures hence the reserved restoration capacity may be shared among the different pre-setup detours (since at most one detour may carry restored traffic at any given time). Thus pre-provisioned link restoration scheme offers the promise of fast restoration recovery for just a small fraction of the total capacity reserved for restoration, due to the high degree of restoration capacity sharing that is possible in mesh networks.

In many situations one would like to support fast restoration on existing networks without disturbing the existing traffic, meaning that the restoration scheme can only use up to the current available network capacity (link capacity minus existing traffic capacity) for protecting the existing traffic and any new traffic. Note that existing traffic makes the problem harder. A simpler polynomial-time 2-approximation algorithm in the absence of existing traffic was presented in [1] (see Related Work).

In this paper we are interested in the optimization problem of dedicating the least amount of the currently available network capacity for protection, while guaranteeing fast restoration to the existing traffic along with any traffic that may be admitted in the future, for the pre-provisioned link restoration scheme. Specifically we are interested in partitioning the available link capacities into working and protection, such that the latter is dedicated for restoration and the former is available to carry any current or new traffic, with the objective of guaranteeing link restoration for minimal total protection capacity. Note that in a network with a static topology this problem may need be solved only once, since the solution remains feasible even as the admitted traffic pattern changes. However, the solution may not stay optimal over time as the admitted traffic pattern changes, and may be recomputed occasionally to ensure efficient utilization of network resources. Also changes in network topology (which are common but not frequent) may require recomputing the solution, since the old solution may not even guarantee link restoration.

1.1 Problem Definition

Given a undirected network $G = (V, E)$, with link capacities u_e and existing traffic W_e on link $e \in E$ the problem is to partition the capacities on link e into a working capacity w_e and a protection capacity p_e (s.t. $w_e + p_e = u_e$) such that

- The total protection capacity $\sum_{e \in E} p_e$ is minimized.
- For every link $e = (u, v)$, in the network $G - e$ obtained by removing link e from G, there exists a path P_e (detour for link e) between nodes u and v, such that every link e' on P_e satisfies $p_{e'} \geq w_e$.
- The working capacity w_e is at least the amount of the existing traffic W_e.
- In case such a partition is not feasible on G, output an empty solution.

In other words, on link e, p_e capacity is reserved for carrying restored traffic during failures and at most w_e ($w_e \geq W_e$) traffic is carried during normal working conditions. Thus on the failure of link e, at most w_e traffic on it is rerouted over the pre-setup detour path P_e using only the reserved capacities on the links on P_e.

Note that given the p_e and w_e values for all links e in any feasible solution, the detour paths P_e can be easily computed. Hence we do not include the computation of the detour paths P_e in the statement of the problem.

1.2 Our Contribution

We show that given an instance of the problem, it can be determined in polynomial time, using a fast and efficient algorithm, if the problem has a feasible solution. However, we show that computing an optimal solution for feasible instances is NP-hard. Moreover, we present a simple and efficient algorithm that computes a solution to the given feasible instance in which the total protection capacity reserved is guaranteed to be within two times the protection capacity reserved in any optimal solution. We also show that the integrality gap of a natural relaxation of our problem is $\Omega(n)$, thus establishing that any LP-based approach using this relaxation cannot yield a better approximation algorithm for our problem.

2 Related Work

The main approaches for supporting pre-provisioned link restoration scheme in mesh networks are based on identifying ring structures. Once the set of rings are identified then pre-planned restoration schemes as in SONET [11] are employed. In some of these approaches the network is designed in term of rings [17] or by partially using rings [9]. Thus, these schemes are only applicable to constrained topologies. In some other of these approaches each link is covered by a cycle leading to a cycle cover for the network [9]. Each of these cycles is then provisioned with enough protection capacity to cover the links that belong to it. On the failure of the link the working traffic is rerouted over the protection capacities in the surviving links of the covering cycle. There are two drawbacks of these approaches: first the amount of pre-provisioned protection capacity can be significant and second it is hard to find the smallest cycle cover of a given network [19]. An improvement to these schemes is those based on the notion of p-cycle [10]. Here the main idea is that a cycle can be used to protect not just the links on the cycle but also the chords (spokes) of the cycle, thus showing that far fewer cycles (than in a cycle cover) may be sufficient for providing full protection. An algorithm to minimize the total spare capacity, based on solving an integer program over all possible cycles is given in [10]. To the best of our knowledge no fast approximation algorithms for this problem are known. An alternative to cycle covers, intended to overcome the difficulty of finding good covers, is to cover every link in a network with exactly two cycles [8]. A set of cycles that meets this requirement is called a double cycle cover [13]. For planar graphs, double cycle covers can be found in polynomial-time. For non-planar graphs, it is conjectured that double cycle covers exist, and they are typically found quickly in practice. However, even for double cycle cover based protection schemes, the required pre-provisioned protection capacity can be significant.

Non-ring based approaches to link restoration on mesh networks include generalized loop-back [14], [15], where the main idea is to select a digraph, called the primary, such that the conjugate digraph, called the secondary, can be used to carry the switched traffic for any link failure in the primary. Chekuri *et al.* [5] consider the problem of adding protection capacity to the links of a given network (primary) carrying working traffic, at minimum cost, so that the resulting network is capable of supporting link protection for a given set of links, where the protection is provided to the working traffic on the primary network. In their model no limit is imposed on the total capacities of the links, and they provide a 4-approximation algorithm when all links in the original primary network have uniform bandwidth (carrying the same amount of working traffic) and they provide a 10.87-approximation algorithm for the general case. In addition they also provide a $O(\log n)$-approximation algorithm for the problem of jointly designing the primary and protected networks, given a demand matrix for the working traffic.

Our previous work [1] considers the problem under the assumption that there is no existing working traffic on any link ($W_e = 0$). We also allow the rerouted traffic to be split on at most two detours in the event of a failure. We show that the optimization problem is NP-hard and provide a 2-approximation algorithm. We also give a lower bound for the problem when there is no restriction on splitting of rerouted traffic.

All the schemes mentioned earlier assume that protection is provided for a single link failure. Choi *et al.* [6] present a heuristic for protecting against two link failures, based on link restoration. The problem of survivable network design has also been extensively studied [4, 3]. Most of the work here has focused on obtaining strong relaxations to be used in cutting plane methods.

3 Algorithm

Recall that u_e is the total capacity of link e, and W_e is the amount of existing traffic in link e. We define $u_e - W_e$ to be the *maximum protection capacity available* on the link e.

Our algorithm (Algorithm A) creates a solution by first computing a maximum spanning tree T_A based on the maximum protection capacity available on the links. It sets the protection capacity of any link e on the maximum spanning tree T_A to the maximum protection capacity available on link e. For any non-tree (T_A) link e (called *cross link*), it initially sets its working capacity equal to the minimum protection capacity assigned to the links on the unique path in T_A between the endpoints of link e. It then selects a few of the cross links and *boosts up* their protection capacities. The main idea is to protect the cross links by using only links in the tree T_A. Also for each tree link e its detour consists of all but one tree link, and the boosting up of some cross links is used to ensure that the cross link on this detour has enough protection to protect e.

The detailed algorithm is given in Algorithm A. We denote the working and protection capacity assigned by Algorithm A on link e by w_e^A and p_e^A respectively ($w_e^A + p_e^A = u_e$). Conceptually the algorithm (as well as the analysis) consists of two phases – finding basic protection capacity in all links, and then boosting the protection capacity for some cross links. These two steps are combined into one in the algorithm, by setting $w_{e_i}^A$ of cross link e_i to $\max(\min(p, u_{e_i} - w), W_{e_i})$ in line 18. p is the minimum protection capacity of the unique path in the tree, which can protect this cross link. w is the maximum working capacity of a link in the unique path, which is still unprotected. So we need to assign at least w protection, but cannot have a working capacity of more than p in this cross link. Also, since we have to support all the existing traffic, $w_{e_i}^A$ has to be at least W_{e_i}.

Algorithm 1 *Algorithm A*

Let $\{e_1, e_2, \ldots, e_m\}$ be the links sorted in decreasing order of maximum available protection capacities ($u_e - W_e$).
$T_A = \phi$
for $i = 1, \ldots, m$ {
  if ($T_A \cup \{e_i\}$ does not form a cycle) {
    $T_A = T_A \cup \{e_i\}$
    if ($W_{e_i} > u_{e_i}/2$)
      No solution exists.
    $w_{e_i}^A = W_{e_i}$
    $p_{e_i}^A = u_{e_i} - w_{e_i}^A$
    Mark e_i as unprotected
  } else {
    Let $P = \{e_{j1}, e_{j2}, \ldots, e_{j_k}\}$ be the unique path in T_A connecting the endpoints of e_i.
    Let M be the links in P which are marked as unprotected.
    $w = \max_{e \in M} w_e^A \quad (w = 0 \text{ if } M = \emptyset)$
    $p = \min_{e \in P} p_e^A$
    if ($p < W_{e_i}$)
      No solution exists.
    $w_{e_i}^A = \max(\min(p, u_{e_i} - w), W_{e_i})$
    $p_{e_i}^A = u_{e_i} - w_{e_i}^A$
    if ($w > p_{e_i}^A$)
      No solution exists.
    Unmark edges in M.
  }
}
if any link e_i is marked
  No solution exists.

3.1 Correctness

Lemma 1. *The solution (if any) returned by Algorithm A is a feasible solution.*

Proof. It is easy to see that the amount of working traffic of every link (w_e^A) in any solution returned by Algorithm A, is always at least the amount of existing traffic (W_e).

Now we show that for every link $e = (u, v)$, there is a backup path P_e in $G - e$ between nodes u and v, such that every link e' on P_e satisfies $p^A_{e'} \geq w^A_e$.
Case 1, for all $e \in T_A$: Note that $G-e$ is connected, because otherwise link e would stay marked at the end of the algorithm in which case no solution is returned by Algorithm A. Thus, there is at least one cross link $e_c = (u_c, v_c)$, which, together with the unique path in T_A between nodes u_c and v_c (excluding link e), forms a backup path for e. Without loss of generality, let e_c be the first cross link considered by the algorithm, such that adding it to T_A results in a cycle C containing link e. Consider the path $P_e = C \setminus \{e\}$. The link e_c has been assigned the least protection by Algorithm A among the links on P_e. Note that link e is marked (hence in M) at the time when e_c is considered by the algorithm. Thus when link e_c is considered by the algorithm we must have $p^A_{e_c} \geq w \geq W_e = w^A_e$ implying that P_e is a valid backup (detour) for link e.
Case 2, for all $e \notin T_A$: The backup path P_e for link e is the unique path in T_A connecting e, which always exists. The links on this path have enough protection because the algorithm sets the working traffic w^A_e of link e to at most $\min_{e' \in P_e} p^A_{e'}$.

Lemma 2. *If a feasible solution exists, the algorithm will return a solution.*

Proof. The algorithm will not return a solution in 4 cases.
Case 1, there exists a link $e \in T_A$, $W_e > u_e/2$: If a tree link e has $W_e > u_e/2$, then no solution exists. This is because, the maximum protection capacity available on e $(u_e - W_e)$ is strictly less than $u_e/2$. If there was a solution, then there must exist a path P_e in $G - e$ between the end points of e all whose links e' have maximum protection capacity available $u_{e'} - W_{e'} > u_e/2$. But then T_A is not a maximum spanning tree based on the maximum protection capacity available on the links, a contradiction.
Case 2, there exists a link $e_c \notin T_A$, $p < W_e$: The proof for this case uses arguments similar to case 1.
Case 3, there exists a link $e_c \notin T_A$, $w > p^A_{e_c}$: Note that in this case $M \neq \emptyset$. Let $e_c = (u_c, v_c)$ and let P_{e_c} be the unique path in T_A connecting nodes u_c and v_c. Let $e \in M$ be a marked link on the path P_{e_c} with $w = W_e = w^A_e$, at the time when e_c is considered by the algorithm. Let P_e be a feasible detour for link e. Thus the maximum protection capacity available on all links e' on P_e is $u_{e'} - W_{e'} \geq w$. Also $P_e \cup \{e\}$ forms a cycle in G, thus at least one link on P_e is a cross link for T_A, that forms a cycle containing link e when added to T_A. This link must have been considered before link e_c since it has strictly more maximum protection capacity available on it. Hence link e must already be marked before link e_c is considered, a contradiction.
Case 4, there is an unmarked link $e = (u, v)$ *at the end*: In this case in $G - e$ nodes u and v are not connected. Hence, no detour is possible for link e.

From the above two lemmas, we have the following theorem.

Theorem 1. *Algorithm A is correct.*

Remark: For each link e in T_A, we can lower its protection capacity until lowering it further would require decreasing the working capacity of some other links, or would make the working capacity of link e so large that there is no feasible detour for it. This trimming may reduce the total protection capacity however, it has no implication on our analysis of the worst-case approximation ratio for the algorithm.