Andrzej Pelc
Michel Raynal (Eds.)

LNCS 3499

Structural Information and Communication Complexity

12th International Colloquium, SIROCCO 2005
Mont Saint-Michel, France, May 2005
Proceedings

Springer

Andrzej Pelc Michel Raynal (Eds.)

Structural Information and Communication Complexity

12th International Colloquium, SIROCCO 2005
Mont Saint-Michel, France, May 24-26, 2005
Proceedings

Springer

Volume Editors

Andrzej Pelc
Université du Québec en Outaouais
Département d'informatique
Gatineau, Québec J8X 3X7, Canada
E-mail: Andrzej.Pelc@uqo.ca

Michel Raynal
Université de Rennes 1, IRISA
Campus de Beaulieu, Avenue du General Leclerc, 35042 Rennes, France
E-mail: raynal@irisa.fr

Library of Congress Control Number: Applied for

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

ISSN 0302-9743
ISBN-10 3-540-26052-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-26052-3 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 Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11429647 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3499

Preface

The Colloquium on Structural Information and Communication Complexity (SIROCCO) is an annual meeting focused on the relationship between algorithmic aspects of computing and communication. Over its 12 years of existence, SIROCCO has become an acknowledged forum bringing together specialists interested in the fundamental principles underlying interplay between information, communication and computing.

SIROCCO 2005 was the twelfth in this series, held in Mont Saint-Michel, France, May 24–26, 2005. Previous SIROCCO colloquia took place in Ottawa (1994), Olympia (1995), Siena (1996), Ascona (1997), Amalfi (1998), Lacanau-Océan (1999), L'Aquila (2000), Val de Nuria (2001), Andros (2002), Umeå (2003) and Smolenice Castle (2004).

SIROCCO covers topics such as distributed and parallel computing, information dissemination, communication complexity, interconnection networks, high-speed networks, wireless networks, mobile computing, optical computing, and related areas.

The 48 contributions submitted to SIROCCO 2005 were subject to a thorough refereeing process and 22 high-quality submissions were selected for publication. We thank the Program Committee members for their excellent and careful work. Our gratitude extends to the numerous subreferees for their valuable refereeing. We also acknowledge the effort of all authors who submitted their contributions.

We thank the invited speakers at this colloquium, Amotz Bar-Noy (New York) and Cyril Gavoille (Bordeaux) for accepting our invitation to share their insights on new developments in their areas of interest. Amotz Bar-Noy delivered a talk "*Cellular Networks: Where Are the Mobile Users?*" and Cyril Gavoille presented "*Distributed Data Structures: a Survey.*"

We would like to express our sincere gratitude to the Steering Committee chair David Peleg (Rehovot) for his enthusiasm and invaluable consultations, and to the organizing team chaired by Elisabeth Lebret from INRIA. Special thanks are due to Rastislav Královič who, besides being a Program Committee member, helped us a lot with all technical work. Finally, we would like to thank IRISA, INRIA, the Université de Rennes 1 and the "Fondation Michel Metivier" for their support.

May 2005 Andrzej Pelc and Michel Raynal

Organization

Program Committee

Carole Delporte-Gallet (Paris, France)
Paola Flocchini (Ottawa, Canada)
Leszek Gąsieniec (Liverpool, UK)
Rachid Guerraoui (Lausanne, Switzerland)
Dariusz Kowalski (Warsaw, Poland)
Rastislav Královič (Bratislava, Slovakia)
Danny Krizanc (Middletown, CT, USA)
Shay Kutten (Haifa, Israel)
Bernard Mans (Macquarie, Australia)
Toshimitzu Masuzawa (Osaka, Japan)
Achour Mostefaoui (Rennes, France)
Jaroslav Opatrny (Montreal, Canada)
Andrzej Pelc, **Co-chair** (Gatineau, Canada)
Sergio Rajsbaum (UNAM, Mexico)
Michel Raynal, **Co-chair** (IRISA, Rennes, France)
Alex Shvartsman (Storrs, CT, USA)
Ugo Vaccaro (Salerno, Italy)

Organizing Committee

Elisabeth Lebret (Chair, INRIA)
Lydie Mabil (INRIA)
Florence Santoro (INRIA)
Webmaster and publication chair: Rastislav Královič (Bratislava, Slovakia)

Steering Committee Chair

David Peleg (Rehovot, Israel)

Referees

Christoph Ambuhl
Surender Baswana
Paolo Boldi
Marek Chrobak
Paolo D'Arco
Shantanu Das
Gianluca De Marco
Xavier Defago
Rafal Dowgird
Pavol Ďuriš
Stefan Dziembowski
Thomas Erlebach
Hugues Fauconnier
Sandor Fekete
Guillaume Fertin
Alexey Fishkin
Satoshi Fujita
Jana Gajdošiková
Clemente Galdi
Anna Gambin
Cyril Gavoille
Franciszek Grzegorek
Jaime Gutierrez
Taisuke Izumi
Hirotsugu Kakugawa
George Karakostas
Branislav Katreniak
Richard Královič
Lukasz Kowalik
Maciej Kurowski
Bruce Litow
Adam Malinowski
David Manlove
Giovanna Melideo
Marcin Mucha
Yoshihiro Nakaminami
Fukuhito Ooshita
Martin Pál
Dana Pardubská
Jakub Pawlewicz
Franck Petit
Giuseppe Prencipe
Andrzej Proskurowski
André Raspaud
Adele Rescigno
Piotr Sankowski
Paul Sant
Nicola Santoro
Mordechai Shalom
Sunil Shende
Samia Soussi
Lukasz Sznuk
Vincent Villain
Ivan Visconti
Koichi Wada
Prudence Wong

Lecture Notes in Computer Science

For information about Vols. 1–3402

please contact your bookseller or Springer

Vol. 3525: A.E. Abdallah, C.B. Jones, J.W. Sanders (Eds.), Communicating Sequential Processes. XIV, 321 pages. 2005.

Vol. 3517: H.S. Baird, D.P. Lopresti (Eds.), Human Interactive Proofs. IX, 143 pages. 2005.

Vol. 3516: V.S. Sunderam, G.D.v. Albada, P.M.A. Sloot, J.J. Dongarra (Eds.), Computational Science – ICCS 2005, Part III. LXIII, 1143 pages. 2005.

Vol. 3515: V.S. Sunderam, G.D.v. Albada, P.M.A. Sloot, J.J. Dongarra (Eds.), Computational Science – ICCS 2005, Part II. LXIII, 1101 pages. 2005.

Vol. 3514: V.S. Sunderam, G.D.v. Albada, P.M.A. Sloot, J.J. Dongarra (Eds.), Computational Science – ICCS 2005, Part I. LXIII, 1089 pages. 2005.

Vol. 3510: T. Braun, G. Carle, Y. Koucheryavy, V. Tsaoussidis (Eds.), Wired/Wireless Internet Communications. XIV, 366 pages. 2005.

Vol. 3508: P. Bresciani, P. Giorgini, B. Henderson-Sellers, G. Low, M. Winikoff (Eds.), Agent-Oriented Information Systems II. X, 227 pages. 2005. (Subseries LNAI).

Vol. 3503: S.E. Nikoletseas (Ed.), Experimental and Efficient Algorithms. XV, 624 pages. 2005.

Vol. 3502: F. Khendek, R. Dssouli (Eds.), Testing of Communicating Systems. X, 381 pages. 2005.

Vol. 3501: B. Kégl, G. Lapalme (Eds.), Advances in Artificial Intelligence. XV, 458 pages. 2005. (Subseries LNAI).

Vol. 3500: S. Miyano, J. Mesirov, S. Kasif, S. Istrail, P. Pevzner, M. Waterman (Eds.), Research in Computational Molecular Biology. XVII, 632 pages. 2005. (Subseries LNBI).

Vol. 3499: A. Pelc, M. Raynal (Eds.), Structural Information and Communication Complexity. X, 323 pages. 2005.

Vol. 3498: J. Wang, X. Liao, Z. Yi (Eds.), Advances in Neural Networks – ISNN 2005, Part III. L, 1077 pages. 2005.

Vol. 3497: J. Wang, X. Liao, Z. Yi (Eds.), Advances in Neural Networks – ISNN 2005, Part II. L, 947 pages. 2005.

Vol. 3496: J. Wang, X. Liao, Z. Yi (Eds.), Advances in Neural Networks – ISNN 2005, Part II. L, 1055 pages. 2005.

Vol. 3495: P. Kantor, G. Muresan, F. Roberts, D.D. Zeng, F.-Y. Wang, H. Chen, R.C. Merkle (Eds.), Intelligence and Security Informatics. XVIII, 674 pages. 2005.

Vol. 3494: R. Cramer (Ed.), Advances in Cryptology – EUROCRYPT 2005. XIV, 576 pages. 2005.

Vol. 3492: P. Blache, E. Stabler, J. Busquets, R. Moot (Eds.), Logical Aspects of Computational Linguistics. X, 363 pages. 2005. (Subseries LNAI).

Vol. 3489: G.T. Heineman, I. Crnkovic, H.W. Schmidt, J.A. Stafford, C. Szyperski, K. Wallnau (Eds.), Component-Based Software Engineering. XI, 358 pages. 2005.

Vol. 3488: M.-S. Hacid, N.V. Murray, Z.W. Raś, S. Tsumoto (Eds.), Foundations of Intelligent Systems. XIII, 700 pages. 2005. (Subseries LNAI).

Vol. 3486: T. Helleseth, D. Sarwate, H.-Y. Song, K. Yang (Eds.), Sequences and Their Applications - SETA 2004. XII, 451 pages. 2005.

Vol. 3483: O. Gervasi, M.L. Gavrilova, V. Kumar, A. Laganà, H.P. Lee, Y. Mun, D. Taniar, C.J.K. Tan (Eds.), Computational Science and Its Applications – ICCSA 2005, Part IV. XXVII, 1362 pages. 2005.

Vol. 3482: O. Gervasi, M.L. Gavrilova, V. Kumar, A. Laganà, H.P. Lee, Y. Mun, D. Taniar, C.J.K. Tan (Eds.), Computational Science and Its Applications – ICCSA 2005, Part III. LXVI, 1340 pages. 2005.

Vol. 3481: O. Gervasi, M.L. Gavrilova, V. Kumar, A. Laganà, H.P. Lee, Y. Mun, D. Taniar, C.J.K. Tan (Eds.), Computational Science and Its Applications – ICCSA 2005, Part II. LXIV, 1316 pages. 2005.

Vol. 3480: O. Gervasi, M.L. Gavrilova, V. Kumar, A. Laganà, H.P. Lee, Y. Mun, D. Taniar, C.J.K. Tan (Eds.), Computational Science and Its Applications – ICCSA 2005, Part I. LXV, 1234 pages. 2005.

Vol. 3479: T. Strang, C. Linnhoff-Popien (Eds.), Location- and Context-Awareness. XII, 378 pages. 2005.

Vol. 3478: C. Jermann, A. Neumaier, D. Sam (Eds.), Global Optimization and Constraint Satisfaction. XIII, 193 pages. 2005.

Vol. 3477: P. Herrmann, V. Issarny, S. Shiu (Eds.), Trust Management. XII, 426 pages. 2005.

Vol. 3475: N. Guelfi (Ed.), Rapid Integration of Software Engineering Techniques. X, 145 pages. 2005.

Vol. 3468: H.W. Gellersen, R. Want, A. Schmidt (Eds.), Pervasive Computing. XIII, 347 pages. 2005.

Vol. 3467: J. Giesl (Ed.), Term Rewriting and Applications. XIII, 517 pages. 2005.

Vol. 3465: M. Bernardo, A. Bogliolo (Eds.), Formal Methods for Mobile Computing. VII, 271 pages. 2005.

Vol. 3463: M. Dal Cin, M. Kaâniche, A. Pataricza (Eds.), Dependable Computing - EDCC 2005. XVI, 472 pages. 2005.

Vol. 3462: R. Boutaba, K. Almeroth, R. Puigjaner, S. Shen, J.P. Black (Eds.), NETWORKING 2005. XXX, 1483 pages. 2005.

Vol. 3461: P. Urzyczyn (Ed.), Typed Lambda Calculi and Applications. XI, 433 pages. 2005.

Vol. 3460: Ö. Babaoglu, M. Jelasity, A. Montresor, C. Fetzer, S. Leonardi, A. van Moorsel, M. van Steen (Eds.), Self-star Properties in Complex Information Systems. IX, 447 pages. 2005.

Vol. 3459: R. Kimmel, N.A. Sochen, J. Weickert (Eds.), Scale Space and PDE Methods in Computer Vision. XI, 634 pages. 2005.

Vol. 3458: P. Herrero, M.S. Pérez, V. Robles (Eds.), Scientific Applications of Grid Computing. X, 208 pages. 2005.

Vol. 3456: H. Rust, Operational Semantics for Timed Systems. XII, 223 pages. 2005.

Vol. 3455: H. Treharne, S. King, M. Henson, S. Schneider (Eds.), ZB 2005: Formal Specification and Development in Z and B. XV, 493 pages. 2005.

Vol. 3454: J.-M. Jacquet, G.P. Picco (Eds.), Coordination Models and Languages. X, 299 pages. 2005.

Vol. 3453: L. Zhou, B.C. Ooi, X. Meng (Eds.), Database Systems for Advanced Applications. XXVII, 929 pages. 2005.

Vol. 3452: F. Baader, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XI, 562 pages. 2005. (Subseries LNAI).

Vol. 3450: D. Hutter, M. Ullmann (Eds.), Security in Pervasive Computing. XI, 239 pages. 2005.

Vol. 3449: F. Rothlauf, J. Branke, S. Cagnoni, D.W. Corne, R. Drechsler, Y. Jin, P. Machado, E. Marchiori, J. Romero, G.D. Smith, G. Squillero (Eds.), Applications of Evolutionary Computing. XX, 631 pages. 2005.

Vol. 3448: G.R. Raidl, J. Gottlieb (Eds.), Evolutionary Computation in Combinatorial Optimization. XI, 271 pages. 2005.

Vol. 3447: M. Keijzer, A. Tettamanzi, P. Collet, J.v. Hemert, M. Tomassini (Eds.), Genetic Programming. XIII, 382 pages. 2005.

Vol. 3444: M. Sagiv (Ed.), Programming Languages and Systems. XIII, 439 pages. 2005.

Vol. 3443: R. Bodik (Ed.), Compiler Construction. XI, 305 pages. 2005.

Vol. 3442: M. Cerioli (Ed.), Fundamental Approaches to Software Engineering. XIII, 373 pages. 2005.

Vol. 3441: V. Sassone (Ed.), Foundations of Software Science and Computational Structures. XVIII, 521 pages. 2005.

Vol. 3440: N. Halbwachs, L.D. Zuck (Eds.), Tools and Algorithms for the Construction and Analysis of Systems. XVII, 588 pages. 2005.

Vol. 3439: R.H. Deng, F. Bao, H. Pang, J. Zhou (Eds.), Information Security Practice and Experience. XII, 424 pages. 2005.

Vol. 3437: T. Gschwind, C. Mascolo (Eds.), Software Engineering and Middleware. X, 245 pages. 2005.

Vol. 3436: B. Bouyssounouse, J. Sifakis (Eds.), Embedded Systems Design. XV, 492 pages. 2005.

Vol. 3434: L. Brun, M. Vento (Eds.), Graph-Based Representations in Pattern Recognition. XII, 384 pages. 2005.

Vol. 3433: S. Bhalla (Ed.), Databases in Networked Information Systems. VII, 319 pages. 2005.

Vol. 3432: M. Beigl, P. Lukowicz (Eds.), Systems Aspects in Organic and Pervasive Computing - ARCS 2005. X, 265 pages. 2005.

Vol. 3431: C. Dovrolis (Ed.), Passive and Active Network Measurement. XII, 374 pages. 2005.

Vol. 3429: E. Andres, G. Damiand, P. Lienhardt (Eds.), Discrete Geometry for Computer Imagery. X, 428 pages. 2005.

Vol. 3428: Y.-J. Kwon, A. Bouju, C. Claramunt (Eds.), Web and Wireless Geographical Information Systems. XII, 255 pages. 2005.

Vol. 3427: G. Kotsis, O. Spaniol (Eds.), Wireless Systems and Mobility in Next Generation Internet. VIII, 249 pages. 2005.

Vol. 3423: J.L. Fiadeiro, P.D. Mosses, F. Orejas (Eds.), Recent Trends in Algebraic Development Techniques. VIII, 271 pages. 2005.

Vol. 3422: R.T. Mittermeir (Ed.), From Computer Literacy to Informatics Fundamentals. X, 203 pages. 2005.

Vol. 3421: P. Lorenz, P. Dini (Eds.), Networking - ICN 2005, Part II. XXXV, 1153 pages. 2005.

Vol. 3420: P. Lorenz, P. Dini (Eds.), Networking - ICN 2005, Part I. XXXV, 933 pages. 2005.

Vol. 3419: B. Faltings, A. Petcu, F. Fages, F. Rossi (Eds.), Constraint Satisfaction and Constraint Logic Programming. X, 217 pages. 2005. (Subseries LNAI).

Vol. 3418: U. Brandes, T. Erlebach (Eds.), Network Analysis. XII, 471 pages. 2005.

Vol. 3416: M. Böhlen, J. Gamper, W. Polasek, M.A. Wimmer (Eds.), E-Government: Towards Electronic Democracy. XIII, 311 pages. 2005. (Subseries LNAI).

Vol. 3415: P. Davidsson, B. Logan, K. Takadama (Eds.), Multi-Agent and Multi-Agent-Based Simulation. X, 265 pages. 2005. (Subseries LNAI).

Vol. 3414: M. Morari, L. Thiele (Eds.), Hybrid Systems: Computation and Control. XII, 684 pages. 2005.

Vol. 3412: X. Franch, D. Port (Eds.), COTS-Based Software Systems. XVI, 312 pages. 2005.

Vol. 3411: S.H. Myaeng, M. Zhou, K.-F. Wong, H.-J. Zhang (Eds.), Information Retrieval Technology. XIII, 337 pages. 2005.

Vol. 3410: C.A. Coello Coello, A. Hernández Aguirre, E. Zitzler (Eds.), Evolutionary Multi-Criterion Optimization. XVI, 912 pages. 2005.

Vol. 3409: N. Guelfi, G. Reggio, A. Romanovsky (Eds.), Scientific Engineering of Distributed Java Applications. X, 127 pages. 2005.

Vol. 3408: D.E. Losada, J.M. Fernández-Luna (Eds.), Advances in Information Retrieval. XVII, 572 pages. 2005.

Vol. 3407: Z. Liu, K. Araki (Eds.), Theoretical Aspects of Computing - ICTAC 2004. XIV, 562 pages. 2005.

Vol. 3406: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing. XVII, 829 pages. 2005.

Vol. 3404: V. Diekert, B. Durand (Eds.), STACS 2005. XVI, 706 pages. 2005.

Vol. 3403: B. Ganter, R. Godin (Eds.), Formal Concept Analysis. XI, 419 pages. 2005. (Subseries LNAI).

Table of Contents

Cellular Networks: Where Are the Mobile Users?
(Invited Talk)

Amotz Bar-Noy

CUNY, New York

Abstract. Mobiles are roaming in a cellular network. Unless they report their new location each time they cross boundaries of cells, the system must conduct a search operation to find their exact location. Reporting new locations by mobiles consumes expensive up-link communication lines. Therefore, in current and future cellular networks, at each point in time for any particular mobile, the system knows only a zone of cells containing the one cell which is the location of this mobile. For this zone, the system maintains a profile that predicts the exact location of the mobile by associating a probability with each cell in the zone. An efficient search should optimize usage of down-link communication lines and the time needed to find the mobile.

This model gives rise to many optimization problems. This talk discusses some of them. We first describe the optimal dynamic programming solution that finds a mobile that is located in a zone of n cells in no more than D rounds. This solution assumes an a priori knowledge of the mobile's profile. We then present solutions in which the system develops a mobile's profile while searching for that mobile more than once. The above solutions are for locating one mobile. Next, we address search operations involving m mobiles where m can be greater than one. One example is the call conference search in which the system must find all the m mobiles. Another example is the yellow pages search where the search is over once one out of the m mobiles is found. Finding an optimal solution to the conference call problem is NP-hard. We therefore present an efficient approximation solution. For the yellow pages problem we discuss work in progress. We conclude with the privacy issue by exploring the tradeoff between the accuracy of the profiles and the efficiency of the optimal solutions that are based on these profiles.

A. Pelc and M. Raynal (Eds.): SIROCCO 2005, LNCS 3499, p. 1, 2005.

Distributed Data Structures: A Survey
(Invited Talk)

Cyril Gavoille

LABRI, Bordeaux

Abstract. This survey concerns the role of data structures for compactly storing and representing various types of information in a localized and distributed fashion. Traditional approaches to data representation are based on global data structures, which require access to the entire structure even if the sought information involves only a small and local set of entities. In contrast, localized data representation schemes are based on breaking the information into small local pieces, or *labels*, selected in a way that allows one to infer information regarding a small set of entities directly from their labels, without using any additional (global) information.

A. Pelc and M. Raynal (Eds.): SIROCCO 2005, LNCS 3499, p. 2, 2005.

On Designing Truthful Mechanisms for Online Scheduling*

Vincenzo Auletta[1], Roberto De Prisco[1,2], and Paolo Penna[1], and Giuseppe Persiano[1]

[1] Dipartimento di Informatica ed Applicazioni "R.M. Capocelli", Università di Salerno, via **S. Allende** 2, I-84081 Baronissi (SA), Italy
{auletta, robdep, penna, giuper}@dia.unisa.it
[2] Faculty Group at Akamai Technologies, Cambridge, MA, USA

Abstract. We study the *online* version of the scheduling problem involving *selfish agents* considered by Archer and Tardos [FOCS 2001]: jobs must be scheduled on m parallel related machines, each of them owned by a different *selfish agent.*

Our study focuses on general techniques to translate approximation/competitive algorithms into equivalent approximation/competitive *truthful mechanisms.* Our results show that this translation is more problematic in the online setting than in the offline one. For $m = 2$, we develop an offline and an online "translation" technique which, given *any* ρ-approximation/competitive (polynomial-time) algorithm, yields an $f(\rho)$-approximation/competitive (polynomial-time) mechanism, with $f(\rho) = \rho(1 + \varepsilon)$ in the offline case, for every $\varepsilon > 0$. By contrast, one of our lower bounds implies that, in general, online ρ-competitive algorithms cannot be turned into $\rho(1+\varepsilon)$-competitive mechanisms, for some $\varepsilon > 0$ and every $m \geq 2$.

We also investigate the issue of designing new online algorithms from scratch so to obtain efficient competitive mechanisms, and prove some lower bounds on a class of "natural" algorithms. Finally, we consider the variant introduced by Nisan and Ronen [STOC 1999] in which machines can be *verified.* For this model, we give a $O(1)$-competitive online mechanism for *any* number of machines and prove that some of the above lower bounds can be broken.

1 Introduction

Optimization problems dealing with resource allocation are classical algorithmic problems and they have been studied for decades in several models. Typically, algorithms are evaluated by comparing the (measure of) the solutions they return to the best possible one. In particular, one tries to estimate the loss of

* Work supported by the European Project IST-2001-33135, Critical Resource Sharing for Cooperation in Complex Systems (CRESCCO).

A. Pelc and M. Raynal (Eds.): SIROCCO 2005, LNCS 3499, pp. 3–17, 2005.

performance due to the lack of computational resources (*approximation ratio*) or to the lack of information (*competitive ratio*).

In both settings, the underlying hypothesis is that the input is (eventually) available to the algorithm (either from the beginning in off-line algorithms or during its execution in on-line algorithms). This assumption cannot be considered realistic in the context of modern networks like the Internet where certain information regarding the resources are not directly available to the "protocol". Indeed, since the resources are owned/controlled/used by different *self-interested* entities (e.g., corporations, autonomous systems, etc.). Each of these entities, or *selfish agents*, hold some *private information* which is needed in order to compute an optimal resource allocation (e.g., routing the traffic over the Internet requires routers of different autonomous systems to exchange information on which routers can process traffic faster). Each agent can possibly *misreport* his/her piece of information if this leads the system to compute a solution that is more beneficial for him/her. This, in spite of the fact that such a solution may *not* be not globally optimal.

The field of *Mechanism Design* is the branch of Game Theory and Microeconomics that studies how to design complex auctions, also termed *mechanisms*, which guarantee that no agents has an incentive in misreporting his/her piece of information. Loosely speaking, a mechanism is a pair $M = (A, P)$, where A is an algorithm computing a solution, and $P = (P^1, \ldots, P^n)$ is the vector of payment functions (see Sect. 1.1 for a formal definition). Selfish agents are suppose to be rational and thus will deviate from the truth-telling strategy (in our problem, to report $r_i = s_i$) only if a better one exists. Therefore, one seeks for *truthful* mechanisms, that is, mechanisms that guarantee that every agent i can maximize his/her net profit or *utility* by playing the truth-telling strategy (see Sect. 1.1).

In this work we consider the *online* version of a basic scheduling/routing problem involving *selfish agents*, first addressed by Archer and Tardos [2]. We will investigate the approximation/competitive ratio of truthful mechanisms for this problem. Our goal is to quantify the (further) loss of optimality due to the combination of selfish agents with the online setting. Central to our study is the existence of general techniques that allow to translate ρ-approximation/competitive algorithms into a $f(\rho)$-approximation/online mechanisms, for some function $f(\cdot)$.

1.1 The Problem

Offline Selfish Version. Consider the problem of scheduling jobs on related machines (Q||C$_{\max}$): We are given a set of m machines with speed $s_1, s_2, \ldots, s_m$ and a set of n jobs of size $J_1, J_2, \ldots, J_n$. We want to assign every job to a machine so to minimize the *makespan*, that is, the maximum over all machines of w_i/s_i, where w_i is the sum of the job weights assigned to machine i. When the set of machines m is fixed, this problem version is commonly denoted to as Q_m||C$_{\max}$.

We study the selfish version of the Q||C$_{\max}$ problem in which each machine i is owned by a selfish agent and the corresponding speed s_i is known to that agent only. In particular, any schedule S that assigns load w_i to machine i is valuated by agent i as $v^i(S)$, where

$$v^i(S) \stackrel{\text{def}}{=} -w_i/s_i,$$

that is, the opposite of the completion time of machine i. Intuitively, $v^i(S)$ represents how much user i likes solution S. This model has been first considered by Archer and Tardos [2].

We stress that our goal is to compute a solution S which minimizes the makespan with respect to the *true* machine speeds $s_1, \ldots, s_m$. Hence, we need to provide some incentive (e.g., a payment P^i) to the each agent i in order to let him/her truthfully report his/her speed. Formally, a *mechanism* is a pair $M = (A, P_A)$, where $P_A = (P_A^1, \ldots, P_A^m)$, and A is a scheduling algorithm. Each agent i reports its type b_i which is not necessarily the true type $t_i \stackrel{\text{def}}{=} 1/s_i$. Algorithm A gets in input the reported types $b = (b_1, \ldots, b_m)$, and each agent i receives a payment equal to $P_A^i(b, J)$. Obviously, each agent i wants to maximize the resulting net profit or *utility* defined as

$$u_i^M(b, J) \stackrel{\text{def}}{=} P_A^i(b, J) + v^i(A(b, J)).$$

Each agent *knows* both algorithm A and the payment function P_A^i.

A mechanism is said to be *truthful with dominant strategies* (or simply *truthful*) if the payments P_A and the algorithm A guarantee that no agent obtains a larger utility when reporting $b_i \neq t_i$, independently of the other agents' reported types; that is, for all J, for all reported types $b_{-i} = (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_m)$ of all the agents except i, and for all possible declarations b_i of agent i, it holds that

$$u_i^M((t_i, b_{-i}), J) \geq u_i^M((b_i, b_{-i}), J),$$

where the writing (x, b_{-i}) denotes the vector $(b_1, \ldots, b_{i-1}, x, b_{i+1}, \ldots, b_m)$. We stress that no agent i has any advantage from knowing the true speeds t_{-i} of the other agents: indeed, the utility of agent i does *not* depend on the speeds of the other agents (the work/payment assigned to machine/agent i depend on the agent bids b only). If M guarantees that the utility is non-negative for all agents i that report their true type, then we say that the mechanism enjoys the *voluntary participation* property.

Online Selfish Version. In the *online* version of Q||C$_{\max}$, jobs arrive one-by-one and must be scheduled upon their arrival. Moreover, jobs cannot be reallocated. For any (possibly infinite) sequence of jobs $J = J_1 J_2 \cdots$, we let J^k denote the prefix $J_1 J_2 \cdots J_k$ of the first k jobs, for $1 \leq k \leq |J|$. Before any job appears, each agent declares her type and we denote by $b = (b_1, \ldots, b_m)$ the vector of declared types. An *online mechanism* for Q||C$_{\max}$ is a pair $M = (A, P)$ where P is a sequence of payment functions P_i^k, for $i = 1, \ldots, m$ and $k > 0$ such that

- The algorithm A is an online algorithm for $\mathrm{Q}||\mathrm{C}_{\max}$; we denote by $w_i^A(b, J^k)$ the sum of the job sizes assigned to machine i by the solution computed by A on input J^k and vector b of declared types.
- When the k-th jobs arrives, it is assigned by A to a machine and each agent i receives *non-negative* payment $P_i^k(b, J^k)$. That is, we are not allowed to ask money back from the agents.

The total payment received by agent i after k jobs is equal to $P_i(b, J^k) = \sum_{j=1}^k P_i^j(b, J^j)$.

Definition 1 (online truthful mechanism). *We say that an online mechanism is* truthful *with respect to* dominant strategies *if for any prefix J^k of J, for all b_{-i}, and for all types t_i, the function $u_i^M((b_i, b_{-i}), J^k)$ is maximized for $b_i = t_i$.*

Verifiable Machines. We also study the online version of the model proposed by Nisan and Ronen [9] of verifiable machines. Here the payment for each job is awarded after the job is released by the machine (we stress that a machine cannot release a job assigned to it before the job has been executed). Intuitively, if a machine has received positive load, the mechanism can verify whether the machine lied declaring to be *faster* and, if so, the machine receives no payment.

1.2 Previous Results

Archer and Tardos [2] have characterized the (offline) algorithms A for $\mathrm{Q}||\mathrm{C}_{\max}$ for which there exist payment functions P such that (A, P) is a truthful mechanism. In particular they show that if an algorithm A is monotone (that is, it satisfies $w_i^A((b'_i, b_{-i}), J) \leq w_i^A((b_i, b_{-i}), J)$, for all $b'_i > b_i$) then there exists a payment function P such that (A, P) is truthful. Under mild assumptions on A, it is possible to define the payment function to guarantee voluntary participation. They also gave a monotone optimal (exponential-time) algorithm for $\mathrm{Q}||\mathrm{C}_{\max}$ and a $(3+\varepsilon)$-approximate randomized (polynomial-time) monotone algorithm. In [4] we gave a $(4+\varepsilon)$-approximate deterministic (polynomial-time) monotone algorithm for $\mathrm{Q}_m||\mathrm{C}_{\max}$. Recently and independently from this work, Andelman *et al* [1] provided an elegant technique for turning any ρ-approximation algorithm for $\mathrm{Q}_m||\mathrm{C}_{\max}$ into a $\rho(1+\varepsilon)$-approximation monotone mechanism. As a result, given any polynomial-time $(1+\varepsilon)$-approximation algorithm for this problem, one can obtain a $(1+\varepsilon)$-approximation mechanism running in polynomial time. They indeed settle the approximation guarantee of the $\mathrm{Q}_m||\mathrm{C}_{\max}$ by obtaining a fully polynomial-time approximation scheme which is monotone. Moreover, they provide a 5-approximation truthful mechanism for the $\mathrm{Q}||\mathrm{C}_{\max}$ problem, i.e., for any number of machines.

Nisan and Ronen [9] considered the case of unrelated machines and gave a randomized 7/4-approximate truthful mechanism for two machines and a deterministic m-approximate truthful mechanism for any number of machines. Moreover, they proved that no deterministic truthful mechanism can be $(2-\varepsilon)$-