Decentralization and Mechanism Design for Online Machine Scheduling

Birgit Heydenreich*, Rudolf Müller, and Marc Uetz

Maastricht University, Quantitative Economics, P.O.Box 616,
6200 MD Maastricht, The Netherlands.
Email: {b.heydenreich,r.muller,m.uetz}@ke.unimaas.nl

Abstract We study the online version of the classical parallel machine scheduling problem to minimize the total weighted completion time from a new perspective: We assume that the data of each job, namely its release date r_j, its processing time p_j and its weight w_j is only known to the job itself, but not to the system. Furthermore, we assume a decentralized setting where jobs choose the machine on which they want to be processed themselves. We study this problem from the perspective of algorithmic mechanism design. We introduce the concept of a myopic best response equilibrium, a concept weaker than the dominant strategy equilibrium, but appropriate for online problems. We present a polynomial time, online scheduling mechanism that, assuming rational behavior of jobs, results in an equilibrium schedule that is 3.281-competitive. The mechanism deploys an online payment scheme that induces rational jobs to truthfully report their private data. We also show that the underlying local scheduling policy cannot be extended to a mechanism where truthful reports constitute a dominant strategy equilibrium.

1 Introduction

We study the online version of the classical parallel machine scheduling problem to minimize the total weighted completion time – $\mathrm{P}\,|\,r_j\,|\,\sum w_j\,C_j$ in the notation of Graham et al. [1] – from a new perspective: We assume a strategic setting, where the data of each job, namely its release date r_j, its processing time p_j and its weight w_j is only known to the job itself, but not to the system. Any job j is interested in being finished as early as possible, and the weight w_j represents its indifference cost for spending one additional unit of time waiting. While jobs may strategically report false values $(\tilde{r}_j, \tilde{p}_j, \tilde{w}_j)$ in order to be scheduled earlier, the total social welfare is maximized whenever the weighted sum of completion times $\sum w_j\,C_j$ is minimized. Furthermore, we assume a restricted communication paradigm, referred to as *decentralization*: Jobs may communicate with machines, but neither do jobs communicate with each other, nor do machines communicate with each other. In particular, there is no central coordination authority hosting all the data of the problem. This leads to a setting

* Supported by NWO grant 2004/03545/MaGW 'Local Decisions in Decentralised Planning Environments'.

Dagstuhl Seminar Proceedings 06461
Negotiation and Market Engineering
http://drops.dagstuhl.de/opus/volltexte/2007/1003

where the jobs themselves must select the machine to be processed on, and any machine sequences the jobs according to a (known) local sequencing policy.

The problem $\mathrm{P} \,|\, r_j \,|\, \sum w_j \, C_j$ is well-understood in the non-strategic setting with centralized coordination. First, scheduling to minimize the weighted sum of completion times with release dates is NP-hard, even in the off-line case [2]. Second, no online algorithm for the single machine problem can be better than 2-competitive [3] regardless of the question whether or not P=NP, and lower bounds exist for parallel machines, too [4]. The best possible algorithm for the single machine case is 2-competitive [5]. For the parallel machine setting, the currently best known online algorithm is 2.61-competitive [6].

In the strategic setting, selfish agents trying to maximize their own benefit can do so by reporting strategically about their private information, thus manipulating the resulting schedule. In the model we propose, a job can report an arbitrary weight, an elongated processing time (e.g. by adding unnecessary work), and it can artificially delay its true release date r_j. We do not allow a job to report a processing time shorter than p_j, as this can easily be discovered and punished by the system, e.g. by preempting the job after the declared processing time $\tilde{p}_j$ before it is actually finished. Furthermore, as we assume that any job j comes into existence only at its release date r_j, it obviously does not make sense that a job reports a release date smaller than the true value r_j.

Our goal is to set up a mechanism that yields a reasonable overall performance with respect to the objective function $\sum w_j \, C_j$. To that end, the mechanism needs to motivate the jobs to reveal their private information truthfully. In addition, as we require decentralization, each machine must be equipped with a local sequencing policy that is publicly known, and jobs must be induced to select the machines in such a way that $\sum w_j \, C_j$ is not too large. Known algorithms with the best competitive ratio, e.g. [6,7], crucially require central coordination to distribute jobs over machines. An approach by Megow et al. [8], developed for an online setting with release dates and stochastic job durations, however, turns out to be appropriate for being adopted to the decentralized, strategic setting.

Related Work and Contribution. Mechanism design in combination with the design of approximation algorithms for scheduling problems has been studied, e.g., by Nisan and Ronen [10], Archer and Tardos [11], and Kovacs [12]. In those papers, not the jobs but the machines are the selfishly behaving parts of the system, and their private information is the time they need to process the jobs. A scheduling model where the jobs are the selfish agents of the system has been studied by Porter [13]. He addresses a single machine scheduling problem, where the private data of each job consists of a release date, its processing time, its weight, and a deadline. In all mentioned papers, the only action of an agent (machine or job, respectively) is to reveal its private data; the resulting mechanisms are also called direct mechanisms. The model suggested in this paper does not give rise to a direct mechanism, since in addition to the revelation of private data, jobs must select the machine to be processed on.

In the algorithm of Megow et al. [8], jobs are locally sequenced according to an online variant of the well known WSPT rule [9], and arriving jobs are assigned to machines in order to minimize an expression that approximates the (expected) increase of the objective value. This algorithm achieves a competitive ratio of 3.281. The mechanism we propose develops their idea further. We present a polynomial time, decentralized online mechanism, called Decentralized LocalGreedy Mechanism. Thereby we provide also a new algorithm for the non-strategic, centralized setting, inspired by the MinIncrease Algorithm of [8], but improving upon the latter in terms of simplicity. We show that the Decentralized LocalGreedy Mechanism is 3.281-competitive. This coincides with the bound that is known for the non-strategic, centralized setting [7,8]. The currently best known bound for the non-strategic setting, however, is 2.61 [6].

As usual in mechanism design, the Decentralized LocalGreedy Mechanism defines *payments* that have to be made by the jobs for being processed. Naturally, we require from an *online* mechanism that also the payments are computed online. Hence they can be completely settled by the time at which a job leaves the system. We also show that the payments result in a balanced budget. The payments induce the jobs to select 'the right' machines. Intuitively, the mechanism uses the payments to mimic a corresponding LocalGreedy online algorithm in the classical (non-strategic, centralized) parallel machine setting $\mathrm{P} \,|\, r_j \,|\, \sum w_j \, C_j$. Moreover, the payments induce rational jobs to truthfully report about their private data. With respect to release dates and processing times, we can show that truthfulness is a dominant strategy equilibrium. With respect to the weights, however, we can only show that truthful reports are myopic best responses (in a sense to be made precise later). In addition, we show that there does not exist a payment scheme extending the allocation rule of the Decentralized LocalGreedy Mechanism to a mechanism where truthful reporting of all private information is a dominant strategy equilibrium.

An extended abstract of this paper has been published in *Algorithm Theory - SWAT 2006, Volume 4059 of Lecture Notes in Computer Science, pp 136-147. Springer, 2006.* A full version is currently under journal review.

References

1. Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: A survey. Ann. Discr. Math. **5** (1979), 287–326
2. Lenstra, J.K., Rinnoy Kan, A.H.G., Brucker, P.: Complexity of machine scheduling problems. Ann. of Discr. Math. **1** (1977), 343–362
3. Hoogeveen, J.A., Vestjens, A.P.A.: Optimal on-line algorithms for single machine scheduling. In: Cunningham, W.H., McCormick, S.T., Queyranne, M., eds.: IPCO 1996, LNCS 1084 (1996), 404-414
4. Vestjens, A.P.A.: On-line Machine Scheduling. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands (1997)
5. Anderson, E.J., Potts, C.N.: Online scheduling of a single machine to minimize total weighted completion time. Math. Oper. Res. **29** (2004), 686-697
6. Correa, J.R., Wagner, M.R.: LP-based online scheduling: from single to parallel machines. In: Jünger, M., Kaibel, V., eds.: IPCO 2005. LNCS 3509 (2005), 196-209

7. Megow, N., Schulz, A.S.: On-line scheduling to minimize average completion time revisited. Oper. Res. Letters **32** (2004), 485-490
8. Megow, N., Uetz, M., Vredeveld, T.: Models and algorithms for stochastic online scheduling. Math. Oper. Res., to appear.
9. Smith, W.: Various optimizers for single stage production. Nav. Res. Log. Quarterly **3** (1956), 59-66
10. Nisan, N., Ronen, A.: Algorithmic mechanism design. Games and Economic Behavior **35** (2001), 166-196
11. Archer, A., Tardos, E.: Truthful mechanisms for one-parameter agents. In: Proc. 42nd FOCS. IEEE Computer Society (2001), 482-491
12. Kovacs, A.: Fast monotone 3-approximation algorithm for scheduling related machines. In: Brodal, G.S., Leonardi, S., eds.: ESA 2005. LNCS 3669 (2005), 616-627
13. Porter, R.: Mechanism design for online real-time scheduling. Proc. 5th ACM Conf. Electronic Commerce, ACM Press (2004), 61-70
14. Eastman, W.L., Even, S., Isaacs, I.M.: Bounds for the optimal scheduling of n jobs on m processors. Management Science **11** (1964), 268–279
15. Lavi, R., Mu'alem, A., Nisan, N.: Towards a characterization of truthful combinatorial auctions. In: Proc. 44th FOCS. IEEE Computer Society (2003), 574–583