Optimal and nearly optimal online and semi-online algorithms for some scheduling problems

Ph.D. thesis

Made by: Dósa György

Supervisor: Vízvári Béla

University of Szeged, Faculty of Science
Doctoral School in Mathematics and Computer Science,
director: Prof. László Hatvani,

Ph.D Program in Informatics,
Operations Research and Combinatorial Optimization Subprogram

Veszprém, 2007 February

1 Introduction

In this thesis we deal with some scheduling problems, investigate some algorithms which are introduced to solve the problems, and analyze their efficiency. The most of the text is based on five common papers [4, 5, 6, 7, 22], which all are coauthored by Dósa György and Yong He. Yong He is unfortunately died very early in 2005. The chapters 1.-5. of the thesis contain new results.

The algorithms being in these papers, and the analysis of them as well, are mainly (in 70 %) made by the author of this thesis. In the remained 30 % the results made by joint contribution of the authors.

All treated problems are NP-hard, and are intensively investigated in the last some years by leading researchers, and also are published in current international journals. In the first chapter of the thesis we give a short review of the classification of some scheduling problems, and the directions of the research of the last years. Now we expose the matter of the next four chapters. (Since the thesis contains 11 new algorithms, there is not enough room here to show all of them, thus we give here only one of them.)

2 Semi-online scheduling problems

In the second chapter we deal with some special semi online versions of problem $P_m|online|C_{\max}$, and give some optimal or nearly optimal algorithms. It is well-known, that algorithm LS is an optimal algorithm for solving the problem $P_2|online|C_{\max}$, and has competitive ratio 3/2 [17, 20]. In case of semi online problems it is an interesting question whether the semi online condition makes the problem to be easier in the next sense: Has the semi online problem an algorithm with smaller competitive ratio than the pure online problem?

Kellerer, Kotov, Speranza and Tuza [33] considered the versions where the sum of the sizes of the jobs is known in advance, or one type of additional algorithm extension is allowed: a buffer is available where some of jobs can be stored, or two parallel processors are available which allows to yield two schedules simultaneously and best one is chosen at last. The first version was also studied by Zhang [39]. It has been shown that for all above versions, optimal semi-online algorithms have competitive ratio 4/3.

To further shed light on usefulness of different types of information, some papers considered whether the combination of two types of information can admit to construct a semi-online algorithm with smaller competitive ratio than that of the case where only one type of information is available. Tan and He [37] presented several kinds of combinations which even together can not decrease the competitive ratio of the online case. Moreover they showed in the same paper that if both the total size and largest size of all jobs are known in advance, or the total size is known in advance and all jobs come in non-increasing order of their sizes, optimal algorithms have competitive ratios $6/5, 10/9$, respectively.

In the second chapter of thesis we study semi-online versions, where two semi online conditions from [33] are combined. Two versions are considered. For the semi-online version where a buffer of length 1 is available and the total size of all jobs is known in advance, we present the optimal algorithm $ALG2.1$, and prove that

Theorem 2.1. *The algorithm $ALG2.1$ is $5/4$-competitive.*

We also show that it does not help that the buffer length is greater than 1.

Theorem 2.2 *There is not such algorithm A for the problem which has competitive ratio less than $5/4$, even if the size of the puffer is more than 1.*

For the semi-online version where two parallel processors are available and the total size of all jobs is known in advance, we present an optimal algorithm $ALG2.2$, and prove that

Theorem 2.3. *The algorithm $ALG2.2$ is $6/5$-competitive..*

Theorem 2.4. *The competitive ratio of an arbitrary algorithm A for the problem is at least $6/5$.*

Thus we can conclude that in both problems by the combination of two semi online conditions the problem can be solved more efficiently, than if only one of that semi online condition holds.

In the remained part of the second chapter we deal with the special version of problem $P_3|online|C_{\max}$, (called as *semi-online with tightly-grouped processing times*), where we know in advance that all jobs have their sizes between 1 and r. It is clear that the information is useless if r is sufficiently large, hence we are interested in the maximum r, denoted by $r_{\max}$, for which a semi-online algorithm can have a better performance than that for the pure online problem. Then the sequence satisfying $r < r_{\max}$ can be called tightly-grouped. The semi-online scheduling problem with tightly-grouped processing times was proposed in [27], which deals only with case $m = 2$. It was shown that the optimal semi-online algorithm has a competitive ratio of $(1 + r)/2$ for any $1 \leq r \leq 2$ and $3/2$ for any $r > 2$. It states that a job sequence is tightly-grouped for $m = 2$ iff $r < 2$. However, the optimal semi-online algorithm is just LS, the same as that for the pure online problem. (In case of the pure online problem, the algorithm

List Scheduling (LS in short) proposed by Graham [20] has competitive ratio $R_{LS} = 2 - 1/m$, and Faigle, Kern and Turán [17] observed that LS is an optimal online algorithm for $m = 2, 3$.) There were no results previously for the $m = 3$ case. Noting that LS is also an optimal algorithm for the pure online problem if $m = 3$, it is interesting to know whether it is still optimal for every $r \geq 1$. We present a comprehensive competitive ratio of LS, which is a continuous, piecewise function on r and can be formulated as follows:

Theorem 2.7. *The competitive ratio of algorithm LS is*

$$R_{LS} = \begin{cases} 1 + \frac{2(r-1)}{3}, & \text{if} \quad 1 \leq r \leq \frac{3}{2}, \\ 2 - \frac{3}{r+3}, & \text{if} \quad \frac{3}{2} < r \leq \sqrt{3}, \\ \frac{r+1}{2}, & \text{if} \quad \sqrt{3} < r \leq 2, \\ 2 - \frac{1}{r}, & \text{if} \quad 2 < r \leq 3, \\ \frac{5}{3}, & \text{if} \quad 3 < r. \end{cases}$$

furthermore LS is an optimal semi online algorithm in cases $1 \leq r \leq 3/2$, $\sqrt{3} \leq r \leq 2$ *or* $r \geq 6$.

Note, that only cases $1 \leq r \leq 3/2$, and $r \geq 3$ were treated previously in [17, 20, 27]. For $r < 3$ we thus obtain competitive ratios below $5/3$ given by the pure on-line optimal algorithm, and algorithm LS is not optimal for all values of r, and for the investigated problem $r_{\max} = 6$ holds. In case $2 < r < 6$ we give three improved algorithms as follows:

a, In case $r \in (2, 5/2]$ we introduce algorithm $ALG(\gamma)$. For this holds the next theorem:

2.20. Theorem 2.20. *For arbitrary* $2 < r \leq 5/2$ *holds that the competitive ratio of* $ALG(3/2)$ *is* $3/2$, *thus it is an optimal semi online algorithm for this case.*

b, In case $r \in (5/2, 3]$ we introduce a near-optimal algorithm$ALG2.3$.

2.24. Theorem 2.24. *The competitive ratio of* $ALG2.3$ *for any* $\frac{5}{2} < r \leq 3$ *is* $\frac{4r+2}{2r+3}$.

We also show that $\frac{7r+4+\sqrt{r^2+8r+4}}{2r+2+2\sqrt{r^2+8r+4}}$ is a lower bound of the problem for arbitrary $r \in (5/2, 3]$. The largest gap between the competitive ratio and the lower bound is at most 0.01417.

c, In case $r \in (3, 6)$ we also present an improved algorithm $ALG2.4$ with smaller competitive ratio than that of LS:

2.28. Theorem 2.28. *The competitive ratio of* $ALG2.4$*, for arbitrary* $3 < r < 6$ *is* $R_{ALG2.4} \leq \Delta = \frac{5}{3} - \frac{\delta}{18} < \frac{5}{3}$*, where* $\delta = \min\left\{\frac{6-r}{18}, \frac{3}{103}\right\} > 0$.

On the base of the above results, we conclude that a job sequence is tightly-grouped for $m = 3$ iff $r < 6$, further the optimal semi-online algorithms strongly depend on the value of r, and LS is not always optimal. All previous algorithms run in $O(n)$ time. The problem of finding optimal algorithms for $r \in (3/2, \sqrt{3})$ and $r \in (5/2, 6)$ is still open.

3 Scheduling uniform machines with rejection

In the third chapter we consider uniform machine scheduling with rejection or USR for short. Each job is characterized by its size (processing time) and its penalty. A job can be either rejected, in which case we pay its penalty, or scheduled on machines, in which case it contributes its processing time to the completion time of that machine. We deal with the USR problem where the number of machines is two. The speed of the machines M_1 and M_2 is $s_1 = 1$, $s_2 = s \geq 1$, respectively. We treat the preemptive, and also the non-preemptive cases, as well. The objective is to minimize the sum of the makespan of the schedule for all accepted jobs and the total penalty of all the rejected jobs. As the straight antecedent of our work, for the on-line non-preemptive USR problem, He and Min [25] provided an on-line algorithm $LSR(\alpha)$. The algorithm is optimal for every $s \geq \phi$, where ϕ denotes the golden ratio.

Regarding our work, in the third chapter, (following [6]), we focus on the on-line USR problem on two uniform machines. Both preemptive and non-preemptive versions are considered.

(i) We present an optimal on-line preemptive algorithm $PMPT(\alpha)$. For this algorithm holds the next theorem:

3.1. Theorem 3.1. *The competitive ratio of algorithm* $PMPT(\alpha)$ *is* $1+\alpha = \frac{s+\sqrt{s^2+4s}}{2s} = \frac{1}{2} + \sqrt{\frac{1}{4} + \frac{1}{s}}$*, thus it is optimal for any* $s \geq 1$.

To our best knowledge, no paper ever considered on-line preemptive USR problem even for two machine case. We note, that for the preemptive case without rejection papers [16, 38] give optimal algorithms, where there are two uniform machines, it is a natural way to combine such algorithm with an appropriate rejection strategy to get an optimal on-line algorithm for the discussed problem. However, we gave a much simpler preemptive scheduling algorithm, which can work well enough. This phenomenon, that a simple and non-optimal scheduling algorithm becomes optimal with a proper rejection strategy also occurs for the identical machine case [35].

(ii) As algorithm $LSR(\alpha)$ is optimal for every $s \geq \phi$, we show that by properly choosing α for $1 \leq s < \phi$, the algorithm can work better. We introduce algorithm $LSRM(\alpha)$ (modified algorithm $LSR(\alpha)$), and show that it has better competitive ratio than $LSR(\alpha)$ for every $1 \leq s < \phi$, furthermore it is even optimal for some values of s, (where the previous algorithm is not optimal). In details, the results of Chapter 3 are the next:

Let $x_0 = \frac{1}{2s}\left(-s^2 + \sqrt{s^4 - 4s^3 + 4s^2 + 4s}\right)$ for any $1 \leq s < \phi$, then x_0 is the only one positive root of equation

$$\frac{1+s-s^2}{xs} = s + x.$$

Furthermore let

$$\beta = \frac{x_0(s+1)}{1+s-s^2} = \frac{(s+1)}{s(s+x_0)} \quad \text{és} \quad \alpha = \frac{\beta}{1+s}.$$

Theorem 3.7. *For any $1 \leq s < \phi$ the competitive ratio of algorithm $LSRM(\alpha)$ is $\frac{1}{2s}\left(s^2 + \sqrt{s^4 - 4s^3 + 4s^2 + 4s}\right)$, which is less than that of algorithm $LSR(\alpha)$.*

Furthermore, noticing that the lower bound in [25] is trivial in a sense that it is also the lower bound of the uniform machine scheduling without rejection, we further present nontrivial lower bounds of the problem under consideration, as follows: Let $1 < s_1 \approx 1.1915$, $s_2 \approx 1.3831$, and $s_3 \approx 1.3852 < \phi$, these real number are the solutions of the next equations, respectively:

$$\frac{1}{2} + \sqrt{\frac{1}{4} + \frac{1}{s}} = \frac{2s+1}{s+1}, \frac{2s+1}{s+1} = \frac{s(s-1)+x_0}{1+s-s^2}, \frac{s(s-1)+x_0}{1+s-s^2} = s + x_0.$$

Theorem 3.8. *The competitive ratio of any algorithm for the treated problem is at least*

$$\begin{cases} \frac{1}{2} + \sqrt{\frac{1}{4} + \frac{1}{s}}, \text{ if } 1 \leq s \leq s_1 \\ \frac{2s+1}{s+1}, \text{ if } s_1 \leq s \leq s_2 \\ \frac{s(s-1)+x_0}{1+s-s^2}, \text{ if } s_2 \leq s \leq s_3 \\ c(s) = s + x_0, \text{ if } s_3 \leq s < \phi \\ \frac{s+1}{s}, \text{ if } \phi \leq s. \end{cases}$$

3.9. Corollary 3.9. *For any $s_3 \approx 1.3852 \leq s < \phi$ algorithm $LSRM(\alpha)$ is optimal.*

Note, that in case $1 \leq s < 1.3852$ the maximum gap between the algorithm's competitive ratio and the lower bound is 0.0534.

4 Scheduling with machine cost

In Chapters 4. and 5. we deal with scheduling problems with machine cost. The partly expanded version of these chapters is appeared in [15]. In [29], Imreh and Noga proposed a variant of the classical parallel machine scheduling problem. The differences are that 1) no machines are initially provided, 2) when a job is revealed the algorithm has the option to purchase new machines, and 3) the objective is to minimize the sum of the makespan and cost of machines. We refer to this problem as the *List Model*. For the List Model problem, Imreh and Noga [29] presented an on-line $(1+\sqrt{5})/2 \approx 1.618$-competitive algorithm A_ρ. For the semi-online problem with known largest size, He and Cai [26] presented an algorithm with a competitive ratio at most 1.5309. For the previous cases the lower bound is $4/3$. These semi-online algorithms are essentially modified from A_ρ.

In this thesis, we first present a new on-line algorithm $H1$ which applies a new strategy to decide when to purchase a new machine, and is the following (k means the number of the next job, while m means the number of the already purchased machines.)

Algorithm $H1$:

1. Assign p_1 to the first machine. Let $k = 2, m = 1$.
2. Assign p_k to the machine with minimum current load if the new load will be no more than $2m$, and go to 4. Else go to 3.
3. Assign p_k to a new machine, $m = m + 1$, go to 4.
4. Let $k = k + 1$, if $k > n$, stop. Otherwise, return 2.

For this algorithm the next theorem holds:

4.9. Theorem 4.9. *The competitive ratio of algorithm $H1$ is at most $(2\sqrt{6}+3)/5 \approx 1.5798$.*

Thus, this competitive ratio improves the known upper bound 1.618. Up to the author's best knowledge, there has not been published yet better algorithm for this problem. Note, that paper of Imreh Csanád [28] again deals with algorithm A_ρ in more general content, where the cost of purchasing a new machine is not constant for all machines, and get a new, efficient algorithm for this general case.

Then, for a special case where every job has a size no greater than the machine cost (called *small job case*). This semi online version is treated first by Dósa and

He [4]. For this case we introduce algorithm $H2$. Note, that in the case of small jobs algorithm A_ρ can not be better than 3/2-competitive. Foe algorithm $H2$ holds the next theorem:

4.11. Theorem 4.11. *In the small job case the competitive ratio of algorithm $H2$ is $4/3$, thus it is an optimal semi online algorithm for this case.*

Note, that this is the first optimal algorithm for some scheduling problem with machine cost. In [28], for general cost function, also given an optimal algorithm for the appropriate small job case.

Last, we present a new two-phase algorithm $H3$ for the semi-online problem with known largest size, which also improves the known result:

4.13. Theorem 4.13. *The competitive ratio of algorithm $H3$ is at most $3/2$.*

We prove the theorem with help of 14 lemmas and two remarks.

5 Scheduling with machine cost and rejection

In Chapter 5 we define that variant of the classical parallel machine scheduling problem which has the special feature that we have a possibility to purchase new machines (introduced by Imreh Csanád and John Noga, [29, 4]), and jobs can be rejected at a certain cost ([1]).

Recently, Nagy-György and Imreh presented a 2.618-competitive online algorithm for the general problem ([30]), optimal algorithm is not known yet. We consider a special case of the problem, called *small job case*, where we assume that all jobs have sizes not greater than 1. The small job case was first proposed in Dósa - He, [4] in the case of the scheduling problem with machine cost. The content of this chapter is appeared in [7]. Note, that our problem MCR in the small job case is even a generalization of the Ski-Rental Problem (SRP for short), which is well known to have optimal deterministic online algorithms with competitive ratio 2, thus 2 is also a lower bound for problem MCR even in the small job case.

In this chapter we present a simple optimal two phase online algorithm for the small job case with a competitive ratio 2.

In the first phase, we reject the first few jobs to avoid the situation that the first machine is purchased too early. In the second phase we avoid that the total penalty of all rejected jobs, or sum of the makespan (for the accepted jobs) and the number of purchased machines would be too large. Because of this carefully approach, opposite to the *Greedy* methods, we call the algorithm as *Carefully.*

Theorem 5.1. *Algorithm $Carefully$ is optimal, and* 2-competitive.

To the scheduling problem with machine cost and rejection it has not introduced yet other optimal algorithm (up to the best knowledge of the author). The proof of the previous theorem also has some new features. We hope that the idea of the algorithm design and analysis presented can be further applied to solve efficiently also the general problem.

References

[1] Bartal, Y., Leonardi, S., Marchetti-Spaccamela, A., Sgall, J., Stougie, L.: Multiprocessor scheduling with rejection, *SIAM J. Discrete Math.,* 13, 64-78(2000).

[2] S.Y. Cai, Y. He, Semi-online algorithms for scheduling non-increasing processing jobs with processor cost, *Acta Automatica Sinica*, 2003, to appear. (in Chinese)

[3] J. Csirik, Cs. Imreh, J. Noga, S. S. Seiden, G. Woeginger, Buying a constant competitive ratio for paging, *Proceedings of ESA* 2001, 98–108.

[4] Dósa, Gy., He, Y., Better on-line algorithms for scheduling with machine cost, *SIAM J. on Computing*, **33,** 1035-1051(2004).

[5] Dósa, Gy., He, Y., Semi-online algorithms for parallel machine scheduling problems, *Computing* 72 (2004), no. 3-4, 355-363.

[6] Dósa, Gy., He, Y., Preemptive and non-preemptive on-line algorithms for scheduling with rejection on two uniform machines, *Computing* 76, 149-164, (2006).

[7] Dósa, Gy., He, Y., Scheduling with machine cost and rejection, *J. Comb. Optim.*, (2006) 12: 337-350 (online)

[8] Dósa, Gy., Graham's example is the only tight one for $P||C_{\max}$, *Annales Univ. Sci. Budapest.*, **47** (2004), 207-210.

[9] Dósa, Gy., Multifit típusú módszerek párhuzamos gépek ütemezésére, *Alkalmazott Matematikai Lapok* 19 (1999) 155-168.

[10] Dósa, Gy., Általánosított Multifit típusú módszerek II, *Alkalmazott Matematikai Lapok* 20 (2000) 91-111.

[11] Dósa, Gy., An optimal algorithm for scheduling jobs with decreasing sizes and machine cost, working paper

[12] Dósa, Gy., Vizvári, B., Az LPT(k)' algoritmus egyforma párhuzamos gépek ütemezésére, *Alkalmazott Matematikai Lapok*, (2004), 269-290.

[13] Dósa, Gy., Vizvári, B., Az általánosított LPT(k) algoritmuscsalád egyforma párhuzamos gépek ütemezésére, *Alkalmazott Matematikai Lapok*, 23 (2006), 17-37.

[14] Dósa, Gy., Generalized Multifit-type methods for scheduling parallel identical machines, *Pure Math. Appl.* **12** (2001), no 3, 287-296.

[15] Dósa, Gy., Optimális és javított algoritmusok a gépköltséges ütemezési feladatra, Pályamű az MTA Veszprémi Területi Bizottsága pályázatára, 2005.

[16] Epstein, L., Noga, J., Seiden, S. S., Sgall, J., Woeginger, G. J.: Randomized online scheduling for two related machines. *J. of Scheduling 4*, 71-92 (2001).

[17] U. Faigle, W. Kern, G. Turán, On the performance of on-line algorithm for particular problems. *Acta Cybernetica*, 9 (1989), 107-119.

[18] G. Galambos, G. Woeginger, An on-line scheduling heuristic with better worst-case ratio than Graham's list scheduling, *SIAM J. Comput.*, 1993, 22, 349-355.

[19] M. R. Garey, D. S. Johnson, Computer and Intractability: A Guide to the theory of NP-Completeness, New York, Freeman, 1979.

[20] R. L. Graham, Bounds for certain multiprocessing anomalies, *The Bell System Technical J.*, 45, 1563-1581, 1966.

[21] Graham, R.L., Bounds on multiprocessor timing anomalies, *SIAM J. Appl. Math.* 17(1969) 416-429.

[22] He, Y., Dósa, Gy., Semi-online scheduling jobs with tightly-grouped processing times on three identical machines, *Discrete Appl. Math.* 150 (2005), no. 1-3, 140-159.

[23] He, Y., Dósa, Gy., Extension of algorithm LS for a semi-online scheduling problem, Central European Journal of Op. Res., (2006) online

[24] He, Y., Kellerer, H., Kotov, V., Linear Compound Algorithms for the Partitioning Problem, Naval Research Logistics, Vol. 47 (2000).

[25] He, Y., Min, X.: On-line machine scheduling with rejection, *Computing*, 65, 1-12 (2000).

[26] He, Y., Cai, S.Y., Semi-online scheduling with machine cost, Journal of Computer Science and Technology 17 (6): 781-787 NOV 2002.

[27] Y. He, G. C. Zhang, Semi on-line scheduling on two identical machines. *Computing*, 62 (1999), 179-187.

[28] Imreh, Cs., On-line scheduling with general machine cost functions, Electronic Notes in Discrete Mathematics, Vol 27, ODSA 2007 Conference, 49-50.

[29] C. Imreh, J. Noga, Scheduling with machine cost, In *Proc. of RANDOM-APPROX'99,* Lecture Notes in Computer Science 1671, Springer-Verlag, 1999, pp. 168-176.

[30] Nagy-György, J., Imreh, Cs., On-line scheduling with machine cost and rejection, working paper, 2006.

[31] Imreh, B., Imreh, Cs., Kombinatorikus optimalizálás, Novadat, 2006

[32] Y.W. Jiang, Y. He, Preemptive online algorithms for scheduling with machine cost, *Acta Informatica*, 41, 315-240, 2005.

[33] H. Kellerer, V. Kotov, M. G. Speranza, Z. Tuza, Semi on-line algorithms for the partition problem. *Operations Research Letters*, 21 (1997) 235-242.

[34] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys, "Sequencing and scheduling: algorithms and complexity", in: *Handbooks in Operation Research and Management Science*, Vol. 4, North-Holland, Amsterdam, 1993, pp. 445-522.

[35] Seiden, S. S.: Preemptive multiprocessor scheduling with rejection. Theoretical Computer Science *262*, 437-458 (2001).

[36] J. Sgall, On-line scheduling, On-line algorithms: The state of art, *Lecture Notes in Computer Sciences* 1442, Springer Verlag, 196-231, 1998.

[37] Z. Y. Tan, Y. He, Semi-on-line problems on two identical machines with combined partial information. *Operations Research Letters,* 30 (2002) 408-414.

[38] Wen, J. J., Du, D. L.: Preemptive on-line scheduling for two uniform processors. *Operations Research Letters 23*, 113-116 (1998).

[39] G. C. Zhang, A simple semi-online algorithm for $P2||C_{\max}$ with a buffer. *Information Processing Letters*, 61 (1997), 145-148.

[40] Zhong, W., Dósa, Gy., Tan, Z., On the machine scheduling problem with job delivery coordination, *European Journal of Operatioanal Research,* 2006, online