BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 62, No. 4, 2014
DOI: 10.2478/bpasts-2014-0088

Due window assignment and scheduling on parallel machines: a FPTAS for a bottleneck criterion

A. JANIAK[1]*, W. JANIAK[2], and M.Y. KOVALYOV[3]

[1] Systems Research Institute, Polish Academy of Sciences, 6 Newelska St., 01-447 Warsaw, Poland
[2] The International University of Logistics and Transport in Wroclaw, 19B Soltysowicka St., 51-168 Wroclaw, Poland
[3] United Institute of Informatics Problems, National Academy of Sciences of Belarus, Minsk, Belarus

Abstract. A fully polynomial time approximation scheme (FPTAS) with run time $O(\frac{n^m}{\varepsilon^{m-1}})$ is developed for a problem which combines common due window assignment and scheduling n jobs on m identical parallel machines. The problem criterion is bottleneck (min-max) such that the maximum cost, which includes job earliness, job tardiness and due window size costs, is minimized.

Key words: fully polynomial time approximation scheme, scheduling, due window assignment.

1. Introduction

There are n independent non-preemptive jobs to be scheduled for processing on m identical parallel machines. We assume that m is a given constant. Each machine can process at most one job at a time, and each job can be completely processed by any of the machines. Job j has a processing requirement p_j, and it is assigned a *common due window* $\langle e, d\rangle$, where e and d are the decision variables to be determined. A schedule determines an allocation of the jobs to the machines and the job start and completion times. Given a schedule S, we denote the completion time of job j by $C_j(S)$, $j = 1, \ldots, n$. The objective is to find an optimal schedule and optimal values of the decision variables e and d such that the weighted maximum

$$F(S,e,d) = \max\{\alpha \max_{1\le j\le n} E_j(S), \beta \max_{1\le j\le n} T_j(S), \gamma(d-e)\}$$

is minimized, where $E_j(S) = \max\{0, d - C_j(S)\}$ and $T_j = \max\{0, C_j(S) - d\}$ are *earliness* and *tardiness* of job j, respectively, $j = 1, \ldots, n$. Job processing times p_j, coefficients α, β and γ, and values of the decision variables e and d are assumed to be non-negative rational numbers. We denote this problem as Pm-DW-max and its optimal solution value as F^*.

The above formulated problem Pm-DW-max belongs to the area of scheduling problems with due windows, see e.g. Janiak et al. [1] for a survey. In this survey it is denoted as $Pm|\langle e,d\rangle|\max\{\alpha\max E_j, \beta\max T_j, \gamma(d-e)\}$.

For a more general problem

$$P|\langle e,d\rangle|\max\{\alpha\max_j E_j, \beta\max_j T_j, \gamma e, \delta(d-e)\}$$

including the due window start time cost, Mosheiov [2] proposed an $O(\max\{nm, n\log n\})$ time asymptotically optimal heuristic algorithm. If the job processing times are independent and identically distributed random variables with bounded support, then the absolute error of the solution obtained by this algorithm approaches zero. Moreover, if job processing times are unit, then this algorithm delivers an optimal solution.

Mosheiov and Sarig [3] studied other due date assignment and scheduling problems with the min-max criterion, namely, $1|\langle e,d\rangle|\max\{\alpha_j E_j + \beta_j T_j + \gamma e + \delta(d-e)\}$, $Pm|\langle e,d\rangle|\max\{\alpha_j E_j + \beta_j T_j + \gamma e + \delta(d-e)\}$ and $Fm|\langle e,d\rangle|\max\{\alpha_j E_j + \beta_j T_j + \gamma e + \delta(d-e)\}$, where 1 and Fm in the first field denote single machine and flow-shop processing environments, respectively. Mosheiov and Sarig showed that for a given job sequence optimal due window can be found for these problems in polynomial time by solving appropriate linear programs.

For one of the problems in [3] with unit time jobs, $Pm|\langle e,d\rangle, p_j = 1|\max\{\alpha_j E_j + \beta_j T_j + \gamma e + \delta(d-e)\}$, Janiak et al. [4] established properties of optimal solutions and reduced it to an assignment problem solvable in $O(n^{4.5}\log^{0.5} n/m^2)$ time. More efficient solution procedures were given for several special cases.

Mosheiov and Sarig [5] presented algorithms for the problems $1|\langle e,d\rangle|\max\{\alpha E_j + \gamma e + \delta(d-e), \beta T_j + \gamma e + \delta(d-e)\}$ and $F2|\langle e,d\rangle|\max\{\alpha E_j + \gamma e + \delta(d-e), \beta T_j + \gamma e + \delta(d-e)\}$ with $O(n)$ and $O(n^2\log n)$ run times, respectively. The case of variable number of identical parallel machines was proved strongly NP-hard, and simple heuristics and lower bounds were introduced and numerically tested for identical parallel machines and uniform machines.

Coming back to the problem Pm-DW-max, note that its single machine version is polynomially solvable, see Mosheiov [2] and Janiak et al. [6]. However, the version with a variable number of identical parallel machine machines is strongly NP-hard as the classic m-identical parallel machine scheduling problem to minimize the makespan, $P||C_{\max}$, pseudo-polynomially reduces to it, see Janiak et al. [6]. A heuristic algorithm and a lower bound for the problem Pm-DW-max are given by Mosheiov [2].

*e-mail: adam.antoni.janiak@gmail.com

Janiak et al. [6] mentioned that a fully polynomial time approximation scheme (FPTAS) of Kovalyov [7] for the problem $Pm||C_{\max}$ with run time $O(n^m/\varepsilon^{m-1})$ can be used to acceptably solve the problem Pm-DW-max. However, an adaptation of this FPTAS for the problem Pm-DW-max is not obvious. This note describes details of this adaptation and presents a FPTAS for the problem $Pm||C_{\max}$, which is a tuned version of the FPTAS in Kovalyov [7].

A family of algorithms $\{A_\varepsilon\}$ is a *Fully Polynomial Time Approximation Scheme (FPTAS)* for the problem Pm-DW-max if for any $\varepsilon > 0$ and any problem instance algorithm A_ε delivers a solution with value $F^A \le (1+\varepsilon)F^*$, and its time requirement is bounded by a polynomial of the problem instance length in binary encoding and in $1/\varepsilon$.

It should be noted that FPTAS, if it exists for an NP-hard problem, is theoretically the best type of algorithms to solve the problem, see Garey and Johnson [8]. Furthermore, FPTASs have shown good performance in computer experiments, see, for example, Martello and Toth [9], Kovalyov et al. [10], Depetrini and Locatelli [11], Bazgan et al. [12], Hu et al. [13].

Applications of the scheduling problems can be found in [14–17].

Next section contains statements which justify our FPTAS for the problem Pm-DW-max. FPTAS for the problem $Pm||C_{\max}$ is described in Sec. 3. The note concludes with a short summary and suggestions for future research.

2. Basic statements

Consider an arbitrary schedule S. Introduce values

$$C_{\min}(S) = \min_{1\le j\le n}\{C_j(S)\}$$

$$\text{and} \quad C_{\max}(S) = \max_{1\le j\le n}\{C_j(S)\}.$$

Re-number jobs in the *Shortest Processing Time (SPT)* order such that $p_1 \le \dots \le p_n$.

Lemma 1. (Corollary of statements in Janiak et al. [6]) There exists an optimal solution (S^*, e^*, d^*) for problem Pm-DW-max, which satisfies the following properties:

- job $n-i+1$ is the first job on machine i and it completes at time $C_{\min}(S^*) = p_n$, $i = 1,\dots,m$,
- $F(S^*, e^*, d^*) = \theta(C_{\max}(S^*) - p_n)$, where $\theta = \dfrac{\alpha\beta\gamma}{\alpha\beta+\alpha\gamma+\beta\gamma}$.

Lemma 2. (Corollary of statements in Janiak et al. [6]) Given a job schedule S such that $C_{\min}(S) = p_n$, the due window $\langle e(S), d(S)\rangle$, which minimizes $F(S, e(S), d(S))$, can be determined as follows:

- $e(S) = \dfrac{\beta\gamma}{\alpha\beta+\alpha\gamma+\beta\gamma}(C_{\max}(S) - p_n) + p_n$,
- $d(S) = \dfrac{\alpha\beta+\beta\gamma}{\alpha\beta+\alpha\gamma+\beta\gamma}(C_{\max}(S) - p_n) + p_n$.

Thus, the problem reduces to finding a job schedule which satisfies Lemmas 1 and 2 and minimizes $C_{\max}(S)$.

It can be easily seen that if $n \le 2m$, then the following solution (S^*, e^*, d^*) is optimal for the problem Pm-DW-max:

$$C_{\max}(S^*) = p_{n-m} + p_n,$$

$$e^* = \frac{\beta\gamma}{\alpha\beta+\alpha\gamma+\beta\gamma}p_{n-m} + p_n,$$

$$d^* = \frac{\alpha\beta+\beta\gamma}{\alpha\beta+\alpha\gamma+\beta\gamma}p_{n-m} + p_n,$$

$$F(S^*, e^*, d^*) = \theta p_{n-m},$$

job $n-i+1$ is the first job on machine i and it completes at time p_n, $i = 1,\dots,m$, and job j is the second job on machine j and it starts at time p_n, $j = 1,\dots,n-m$.

Assume without loss of generality that $n \ge 2m+1$. Consider the classic problem $Pm||C_{\max}$ with the set of jobs $K = \{1,\dots,k\}$, where $k = n-m$. Its solution can be described by an $m \times k$-dimensional 0-1 matrix x with entries x_{ij} such that $x_{ij} = 1$ if and only if job j is processed by machine i, $i = 1,\dots,m$, $j = 1,\dots,k$. The order of jobs assigned to the same machine can be arbitrary.

Let x^* and $C^*(K)$ denote an optimal solution of the problem $Pm||C_{\max}$ with the job set K and its optimal makespan value, respectively. Optimal solution for the problem Pm-DW-max is obtained by scheduling jobs $n-m+1, n-m+2,\dots,n$ as it is indicated in Lemma 1, and assigning jobs $1,\dots,n-m$ to the machines as it is indicated by x^*. Furthermore,

$$e^* = \frac{\beta\gamma}{\alpha\beta+\alpha\gamma+\beta\gamma}C^*(K) + p_n,$$

$$d^* = \frac{\alpha\beta+\beta\gamma}{\alpha\beta+\alpha\gamma+\beta\gamma}C^*(K) + p_n$$

and

$$F^* = \theta C^*(K).$$

Let $\{H_\varepsilon\}$ be a FPTAS for the problem $Pm||C_{\max}$ with the job set K. By the definition, it delivers a solution x^0 with the makespan value $C^0(K) \le (1+\varepsilon)C^*(K)$. Let S^0 be solution of the problem Pm-DW-max extended from x^0 by jobs $n-m+1, n-m+2,\dots,n$, as it is indicated in Lemma 1. Making use of Lemma 2, we set

$$e^0 = \frac{\beta\gamma}{\alpha\beta+\alpha\gamma+\beta\gamma}C^0(K) + p_n$$

and

$$d^0 = \frac{\alpha\beta+\beta\gamma}{\alpha\beta+\alpha\gamma+\beta\gamma}C^0(K) + p_n.$$

We have

$$F(S^0, e^0, d^0) = \theta C^0(K) \le \theta(1+\varepsilon)C^*(K) = (1+\varepsilon)F^*.$$

Hence, the family of algorithms $\{H_\varepsilon\}$, together with the procedure that extends S^0 from x^0, constitutes a FPTAS for the problem Pm-DW-max.

Our FPTAS for the problem $Pm||C_{\max}$ with the job set K is described in the next section. It is a tuned version of the FPTAS, which was suggested in Ph.D. thesis of Kovalyov [7] and which was not published.

3. A FPTAS for the problem $Pm||C_{\max}$

Problem $Pm||C_{\max}$ with the job set K can be formulated as the following integer programming problem. Denote $T_i(x) = \sum_{j=1}^{k} p_j x_{ij}$, $i = 1, \ldots, m$.

$$\text{Minimize } T(x)$$

$$= \max\Big\{ \max_{1\le i\le m-1} T_i(x), \sum_{j=1}^{k} p_j - \sum_{i=1}^{m-1} T_i(x) \Big\},$$

$$\text{subject to} \quad \sum_{i=1}^{m} x_{ij} = 1,$$

$$x_{ij} \in \{0,1\}, \quad j = 1, \ldots, k, \quad i = 1, \ldots, m.$$

For the problem $Pm||C_{\max}$ with the job set K, there exists the well-known list scheduling algorithm, which assigns jobs in the *Longest Processing Time (LPT)* order $k, k-1, \ldots, 1$ to the first available machine. This algorithm delivers a solution with value $C^{(1)}(K) \le \Delta C^*(K)$, where $\Delta = \frac{4}{3} - \frac{1}{3m}$, in $O(k \log k)$ time, see Graham [18]. Set $L = \frac{C^{(1)}(K)}{\Delta}$. We have

$$0 < L \le C^*(K) \le \Delta L.$$

Let us introduce a scaling parameter $\delta = \left\lfloor \frac{\varepsilon L}{k} \right\rfloor$ and scaled job processing times $p'_j = \left\lfloor \frac{p_j}{\delta} \right\rfloor$, $j = 1, \ldots, k$. Denote by x' optimal solution for the problem $Pm||C_{\max}$ with the job set K and scaled processing times p'_j.

Lemma 3. $T(x') \le (1+\varepsilon) T(x^*)$.

Proof. The following chain of relations proves this lemma:

$$T(x') = \delta \max\left\{ \max_{1\le i\le m-1} \left\{ \frac{\sum_{j=1}^{k} p_j x'_{ij}}{\delta} \right\}, \frac{\sum_{j=1}^{k} p_j \left(1 - \sum_{i=1}^{m-1} x'_{ij}\right)}{\delta} \right\}$$

$$\le \delta \max\left\{ \max_{1\le i\le m-1} \Big\{ \sum_{j=1}^{k} p'_j x'_{ij} + k \Big\}, \sum_{j=1}^{k} p'_j \left(1 - \sum_{i=1}^{m-1} x'_{ij}\right) + k \right\}$$

$$= \delta \max\left\{ \max_{1\le i\le m-1} \left\{ \sum_{j=1}^{k} p'_j x'_{ij} \right\}, \sum_{j=1}^{k} p'_j \left(1 - \sum_{i=1}^{m-1} x'_{ij}\right) \right\} + k\delta$$

$$\le \delta \max\left\{ \max_{1\le i\le m-1} \left\{ \sum_{j=1}^{k} p'_j x^*_{ij} \right\}, \sum_{j=1}^{k} p'_j \left(1 - \sum_{i=1}^{m-1} x^*_{ij}\right) \right\} + k\delta$$

$$\le \max\left\{ \max_{1\le i\le m-1} \left\{ \sum_{j=1}^{k} p_j x^*_{ij} \right\}, \sum_{j=1}^{k} p_j \left(1 - \sum_{i=1}^{m-1} x^*_{ij}\right) \right\} + k\delta$$

$$= T(x^*) + k\delta = T(x^*) + \varepsilon L \le (1+\varepsilon) T(x^*).$$

Denote by $t(x)$ function $T(x)$, in which processing times p_j are replaced by scaled job processing times p'_j, $j = 1, \ldots, k$. Observe that $T(x^*) \le \Delta L$ implies

$$t(x') \le t(x^*) \le \left\lfloor \frac{\Delta L}{\delta} \right\rfloor := U.$$

Problem $Pm||C_{\max}$ with the scaled job processing times can be solved by the following dynamic programming algorithm, denoted as DP. In iteration j of this algorithm, j-th column of partial solution x is determined, that is, job j is assigned to a machine. Each partial solution x, in which first j columns are determined, is associated with the *state* $(t_1, \ldots, t_{m-1})$, where $t_i = \sum_{h=1}^{j} p_h x_{ih}$, $i = 1, \ldots, m-1$. It is sufficient to consider $t_i \le U$, $i = 1, \ldots, m-1$. The set of states in iteration j is denoted as G_j. Sets $G_1, \ldots, G_k$ are recursively constructed. An optimal solution corresponds to a state in the set G_k. A formal description of algorithm DP is given below.

Algorithm DP.

Step 1 (Initialization) Set $G_0 = \{(0, \ldots, 0)\}$. Calculate $P'_j = \sum_{h=1}^{j} p'_h$, $j = 1, \ldots, k$.

Step 2 (Recursion) Calculate

$$G_j = \Big\{ (t_1 + p'_j x_{1j}, \ldots, t_{m-1} + p'_{m-1} x_{1,m-1}) \mid (t_1, \ldots, t_{m-1}) \in X_{j-1}, t_i + p'_j x_{ij} \le U,$$

$$P'_j - \sum_{i=1}^{m-1} t'_i \le U,$$

$$\sum_{i=1}^{m-1} x_{ij} \le 1, x_{ij} \in \{0,1\}, i = 1, \ldots, m-1 \Big\},$$

$$j = 1 \ldots, k.$$

Step 3 (Optimal solution) Calculate

$$C^*(K) = \max_{(t_1,\ldots,t_{m-1}) \in G_k} \Big\{ \max_{1\le i\le m-1} t_i, P'_k - \sum_{i=1}^{m-1} t_i \Big\}$$

and determine corresponding optimal solution by backtracking.

The set G_j can be generated in $O(m|G_{j-1}|)$ time, $j = 1, \ldots, k$. Since for each state $(t_1, \ldots, t_{m-1}) \in G_j$ we have $t_i \le U$, $i = 1, \ldots, m-1$, relation $|G_j| \le U^{m-1}$ takes place, $j = 1, \ldots, k$. Then, the run time of algorithm DP does not exceed $O(kmU^{m-1})$, or equivalently,

$$O\left(m \frac{k^m}{\varepsilon^{m-1}} \right) = O\left(m \frac{(n-m)^m}{\varepsilon^{m-1}} \right).$$

We deduce that the suggested solution approach constitutes a FPTAS for the problem Pm-DW-max, if m is a constant.

4. Conclusions

We described a FPTAS for the problem Pm-DW-max with a bottleneck (min-max) criterion. In the future it can be interesting to develop FPTASs for the due date assignment and scheduling problems on parallel machines with a min-sum criterion.

Acknowledgements. This research has been supported by Polish National Science Center, decision No. DEC-2011/03/B/ST6/00364.

REFERENCES

[1] A. Janiak, T. Kwiatkowski, and M. Lichtenstein, "Scheduling problems with a common due window assignment: A survey", *Int. J. Applied Mathematics and Computer Science* 23 (1), 231–241 (2013).

[2] G. Mosheiov, "A due-window determination in minmax scheduling problems", *INFOR* 39, 107–123 (2001).

[3] G. Mosheiov and A. Sarig, "Scheduling with a common due-window: Polynomially solvable cases", *Information Sciences* 180, 1492–1505 (2010).

[4] A. Janiak, W. Janiak, M.Y. Kovalyov, and F. Werner, "Soft due window assignment and scheduling of unit-time jobs on parallel machines", *4OR*, DOI: 10.1007/s10288-012-0201-4 (2012).

[5] G. Mosheiov and A. Sarig, "Minmax scheduling problems with a common due-window", *Computers & Operations Research* 36, 1886–1892 (2009).

[6] A. Janiak, M.Y. Kovalyov, and M. Marek, "Soft due window assignment and scheduling on parallel machines", *IEEE Trans. on Systems, Man and Cybernetics. Part* A 37, 614–620 (2007).

[7] M.Y. Kovalyov, "Interval ε-approximation algorithms of solving discrete extremum problems", *Candidate of Sciences Dissertation*, Institute of Engineering Cybernetics, Academy of Sciences of BSSR, Minsk, 1986, (in Russian).

[8] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W.H. Freeman and Co., San Francisco, 1979.

[9] S. Martello and P. Toth, "Approximation schemes for the subset-sum problem: survey and experimental analysis", *Eur. J. Operational Research* 22, 56–69 (1985).

[10] M.Y. Kovalyov, M.-C. Portmann, and A. Oulamara, "Optimal testing and repairing a failed series system", *J. Combinatorial Optimization* 12 (3), 279–295 (2006).

[11] D. Depetrini and M. Locatelli, "A FPTAS for a class of linear multiplicative problems", *Computational Optimization and Applications* 44 (2), 275–288 (2009).

[12] C. Bazgan, H. Hugot, and D. Vanderpooten, "Implementing an efficient FPTAS for the 0-1 multi-objective knapsack problem", *Eur. J. Operational Research* 198 (1), 47–56 (2009).

[13] S. Hu, Z. Li, and C.J. Alpert, "A fully polynomial-time approximation scheme for timing-constrained minimum cost layer assignment", *IEEE Trans. on Circuits and Systems II: Express Briefs* 56 (7), 580–584 (2009).

[14] T. Sawik, "A mixed integer program for cyclic scheduling of flexible flow lines", *Bull. Pol. Ac.: Tech.* 62 (1), 121-128 (2014).

[15] M. Klimek and P. Lebkowski, "Robustness of schedules for project scheduling problem with cash flow optimisation", *Bull. Pol. Ac.: Tech.* 61 (4), 1005–1015 (2013).

[16] T. Cichowicz, M. Drozdowski, M. Frankiewicz, G. Pawlak, F. Rytwinski, and J. Wasilewski, "Hyper-heuristics for cross-domain search", *Bull. Pol. Ac.: Tech.* 60 (4), 801–808 (2012).

[17] W. Frohmberg, M. Kierzynka, J. Blazewicz, and P. Wojciechowski, "G-PAS 2.0 an improved version of protein alignment tool with an efficient backtracking routine on multiple GPUs", *Bull. Pol. Ac.: Tech.* 60 (3), 491–494 (2012).

[18] R.L. Graham, "Bounds on multiprocessing timing anomalies", *SIAM J. Applied Mathematics* 17, 263–269 (1969).