SCHEDULING ON BATCH PROCESSING MACHINES WITH COMPATIBILITY GRAPH

Bouakaz Khaoula, Boudhar Mourad and Bendraouche Mohamed
Laboratoire RECITS, Faculté de Mathématiques, USTHB, Algérie

ABSTRACT

We consider the problem of minimizing the makespan on batch processing identical parallel machines, subject to compatibility constraints, where two jobs are compatibles if they can be processed simultaneously in the same batch. These constraints are modeled by a graph in which compatible jobs are represented by adjacent vertices. We show that several subproblems are polynomial. We propose certain exact polynomial algorithms to solve these subproblems.

Key words : batch scheduling, batch processing machine, compatibility graph, identical parallel machines.

1. INTRODUCTION

In classical scheduling theory, it is assumed that machines can only process one job at a time. However, in reality there is an other scheduling models that can be called "batch machines" and that refers to batches of jobs to be processed together (material painted together, material rolled together, material transported together, etc. . .). In these cases, the jobs in the same batch have to be compatible.

The principal motivation for batch scheduling is the scheduling of burn-in operations in the semiconductor industry where are exposed to high temperatures in a fixed capacity oven in order to weed out chips susceptible to premature failure.

In this work, we consider the problem of scheduling a set of jobs $J_1,...,J_n$ non-preemptively on batch processing identical parallel machines to minimize the makespan C_{max}. We assume that the jobs are subject to compatibility constraints modelled by an undirected graph G, we call the compatibility graph, in which each job is represented by a vertex, and each edge joins a pair of jobs that can be processed simultaneously in the same batch. By definition, a batch belongs to a clique of G. The capacity b of a batch, and hence the clique size, may be finite $b = k$, variable, or infinite $b = \infty$(it can process all jobs simultaneously). The job J_i has the processing time p_i and the processing time of a batch is equal to the maximum processing time or the sum of processing time of any job assigned to it. All jobs in a batch must be available at the same date, start at the same date and finish at the same date. We assume that a setup-time s must separate two successive batches and that the tasks of the same batch are processed without setup-time. The setup-time is independent of the batches sequence, it is identical between each two successive batches on all machines. We denote this batch scheduling problem with a fixed number of machines m by $Bm,max/G = (V,E),b, p_i,s/C_{max}$ and $Bm,sum/G = (V,E),b, p_i,s/C_{max}$.

We show in this paper that the scheduling problems $B1,max/G = (V,E),b = 2, p_i,s/C_{max}$ can be solved in $O(n^3)$, $B1,sum/G = (V,E),b = 2, p_i,s/C_{max}$ can be solved in $O(n^{2.5})$, $Bm,max/G = (V,E),b = 2, p_i = p,s/C_{max}$ can be solved in $O(n^{2.5})$ and $Bm,sum/G = (V,E),b = 2, p_i = p,s/C_{max}$ can be solved in $O(n^{2.5})$. We give polynomial algorithm and the value of the makespan.

2. LITERATURE REVIEW

The theory of batch scheduling for various scheduling objectives and additional constraints is already well established in [5,15]. Some very special cases of job compatibility, the Incompatible Families structures [9, 13, 14, 16] and the Compatible Families structures [5, 6, 12, 13] have previously been treated. Batch scheduling with a general compatibility graph G has been analyzed in [1]. In the same reference the authors have proved that the problem with an arbitrary compatibility graph G denoted $B1/G=(V,E),b=2/C_{max}$ can be solved in polynomial time of order $O(n^3)$ as a maximum weight matching problem. Note that the sub-problem with an identical processing time denoted $B1/G=(V,E),b=2,pi=1/C_{max}$ can be solved in polynomial time of order $O(n^{2.5})$ as a maximum cardinality matching problem. The special cases with bipartite compatibility graph and split compatibility graph have been analyzed respectively in [4] and [3]. In [3], each job has a release date and a processing time equal to 1. Intensive research has subsequently been developed on this subject for various scheduling objectives and additional constraints, see for instance the surveys [5].

3. COMPLEXITY RESULTS

We, here, briefly present new results regarding the problems $B1,max/G=(V,E),b=2,s/C_{max}$, $B1,sum/G=(V,E),b=2,s/C_{max}$, $Bm,max/G=(V,E),b=2,p_i=p,s/C_{max}$ and $Bm,sum/G=(V,E),b=2,p_i=p,s/C_{max}$. These problems with an arbitrary compatibility graph G can be solved in polynomial time.

3.1. Case of a single machine

We show in this section that the problem of scheduling a set of n jobs $J_1,...,J_n$ non-preemptively on a single batch processing machine to minimize the makespan is polynomial. Where the jobs are subject to compatibility constraints modelled by an undirected graph G, each task has a processing time p_i, each batch has a capacity $b=2$ and a setup-time s between each two batches.

Theorem 1 *The problem $B1,max/G=(V,E),b=2,s/C_{max}$ reduces to the maximum weight matching.*

The following algorithm solves the problem $B1,max/G=(V,E),b=2,s/C_{max}$.

Algorithm 1

Input: $G=(V;E)$, p_i, s
Result: schedule σ

1. From the graph $G=(V;E)$, construct a new valued graph $H_\alpha=(G,\alpha)$ where each edge $e=(J_i,J_j)\in E$ is valued by $\alpha(e)=min\{p(J_i);p(J_j)\}+s$.
2. Find a maximum weight matching M in the graph H_α.
3. Form the batches of σ :
   - for each edge of the matching M, process the corresponding two jobs in the same batch.
   - Other jobs are processed as single job batches. (run the batches in an arbitrary order).
4. $C_{max}(\sigma)=\sum_{J\in V}p(J)+s(n-1)-\sum_{k/e_k\in M}\alpha(e_k)$.

The best known algorithm for the maximum weight matching is in $O(n^3)$ [11]. Hence, also the algorithm 1 runs in $O(n^3)$.

Example 1. Let us process 6 jobs $J_1; J_2; J_3; J_4; J_5$ and J_6 on a single batch processing machine of capacity equal to 2 and setup-time equal to 2. The processing times of jobs are given in TABLE 1. The compatibility graph is given in FIGURE 1.

J_i	J_1	J_2	J_3	J_4	J_5	J_6
p_i	6	5	4	3	1	1

TABLE 1 – Processing times of jobs of Example 1

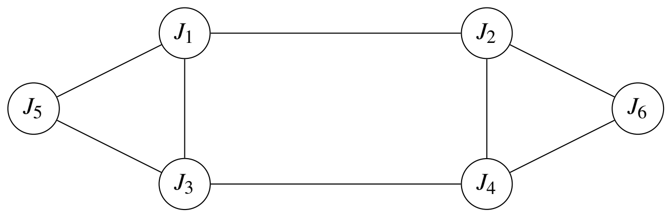

FIGURE 1 – The graph $G = (V, E)$ of Example 1

We obtain the following optimal solution :

The number of batches is equal to 3 ;
$B_1 = (J_1; J_2)$; $pb_1 = max\{6, 5\} = 6$;
$B_2 = (J_3; J_5)$; $pb_2 = max\{4, 1\} = 4$;
$B_3 = (J_4; J_6)$; $pb_3 = max\{3, 1\} = 3$;

We obtain a makespan $C_{max} = 17$, the schedule is in FIGURE 2.

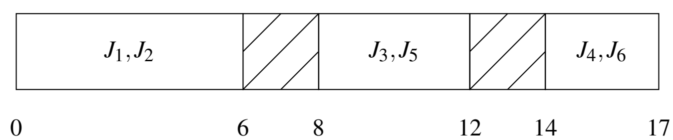

FIGURE 2 – Optimal schedule of Example 1

Theorem 2 *The problem $B1, sum/G = (V, E), b = 2, s/C_{max}$ reduces to the maximal cardinality matching.*

The following algorithm solves the problem $B1, sum/G = (V, E), b = 2, s/C_{max}$.

Algorithm 2

Input: $G = (V;E)$, p_i, s
Result: schedule σ

1. Find a maximal cardinality matching M in the graph G.
2. Form the batches of σ :
   - for each edge of the matching M, process the corresponding two jobs in the same batch.
   - Other jobs are processed as single job batches. (run the batches in an arbitrary order).
3. $C_{max}(\sigma) = \sum_{J \in V} p(J) + s(n - |M| - 1)$.

The best known algorithm for the maximal cardinality matching is in $O(n^{2.5})$ [10]. Hence, also the algorithm 2 runs in $O(n^{2.5})$.

Example 2. We solve the instance of Example 1 by algorithm 2.

We obtain the following optimal solution :

The number of batches is equal to 3 ;
B_1 = $(J_1;J_2)$; $pb_1 = 11$;
B_2 = $(J_3;J_5)$; $pb_2 = 5$;
B_3 = $(J_4;J_6)$; $pb_3 = 4$;

We obtain a makespan $C_{max} = 24$, the schedule is in FIGURE 3.

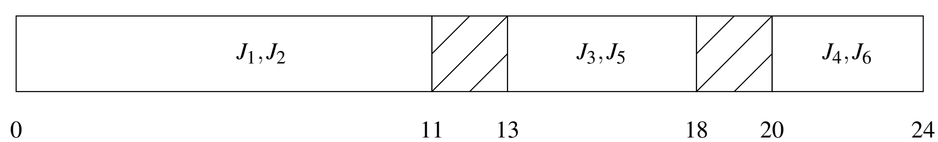

FIGURE 3 – Optimal schedule of Example 2

3.2. Case of several machines

We show in this section that the problem of scheduling a set of n jobs $J_1, ..., J_n$ non-preemptively on m batch processing identical parallel machines to minimize the makespan is polynomial. Where the jobs are subject to compatibility constraints modelled by an undirected graph G, each task has a processing time p, each batch has a capacity $b = 2$ and a setup-time s between each two batches.

Theorem 3 *The problem $Bm, max/G = (V,E), b = 2, p_i = p, s/C_{max}$ reduces to the maximal cardinality matching.*

The following algorithm solves the problem $Bm, max/G = (V,E), b = 2, p_i = p, s/C_{max}$.

Algorithm 3

Input: $G=(V;E),p,s$
Result: schedule σ

1. Find a maximal cardinality matching M in the graph G.
2. Form the following batches of σ :
   - For each edge of the matching M, process the corresponding two jobs in the same batch.
   - Other jobs are processed in single job batches.
3. Run the batches at the first available machine among the m machines.
4. $C_{max}(\sigma)=\lceil \frac{n-|M|}{m} \rceil p+(\lceil \frac{n-|M|}{m} \rceil -1)s$.

The best known algorithm for the maximal cardinality matching is in $O(n^{2.5})$ [10]. Hence, also the algorithm 3 runs in $O(n^{2.5})$.

Example 3. Let us process 6 jobs $J_1;J_2;J_3;J_4;J_5$ and J_6 on $m=3$ batch processing machines of capacity $b=2$, setup-time $s=1$ and $p_i=2,\forall J_i\in V, i=1...n$. The compatibility graph is given in figure 4.

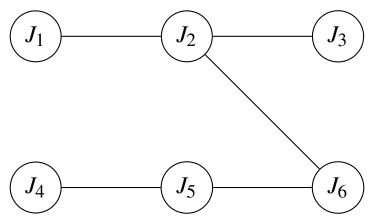

FIGURE 4 – The graph $G=(V,E)$ of Example 3

we obtain the following optimal solution :

The number of batches is equal to 4 :
B_1 = $(J_1;J_2)$;
B_2 = $(J_4;J_5)$;
B_3 = (J_3) ;
B_4 = (J_6) ;
We obtain a makespan $C_{max}=5$, the schedule is in figure 5.

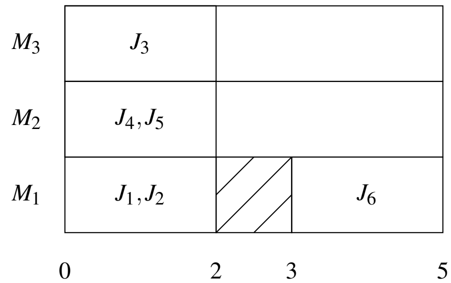

FIGURE 5 – Optimal schedule of Example 3

Theorem 4 *The problem $Bm, sum/G = (V,E), b = 2, p_i = p, s/C_{max}$ reduces to the maximal cardinality matching.*

The following algorithm solves the problem $Bm, sum/G = (V,E), b = 2, p_i = p, s/C_{max}$.

Algorithm 4

Input: $G = (V;E), p, s$
Result: schedule σ

1. Find a maximal cardinality matching M in the graph G.
2. Form the following batches of σ :
   - For each edge of the matching M, process the corresponding two jobs in the same batch.
   - Other jobs are processed in single job batches.
3. Run the batches at the first available machine among the m machines according to the decreasing order of their processing times.
4. The makespan equal to :

$$C_{max}(\sigma) = \begin{cases} \lceil \frac{|M|}{m} \rceil 2p + (\lceil \frac{|M|}{m} \rceil - 1)s & if\ n - 2|M| \leq k \\ \lceil \frac{|M|}{m} \rceil 2p + (\lceil \frac{|M|}{m} \rceil)s & if\ k < n - 2|M| \leq 2k \\ (2\lceil \frac{|M|}{m} \rceil + \lceil \frac{c}{m} \rceil)p + (\lceil \frac{|M|}{m} \rceil + \lceil \frac{c}{m} \rceil - 1)s & if\ c > 0\ and \\ & reste(c,m) \leq reste(|M|,m) \\ (2\lceil \frac{|M|}{m} \rceil + \lceil \frac{c}{m} \rceil)p + (\lceil \frac{|M|}{m} \rceil + \lceil \frac{c}{m} \rceil)s & if\ c > 0\ and\ (reste(c,m) = 0 \\ & or\ \text{reste(c, m)>reste}(|M|,m)) \end{cases}$$

Where the function $reste(c,m)$ is the remainder of the division of c by m.

The best known algorithm for the maximal cardinality matching is in $O(n^{2.5})$ [10]. Hence, also the algorithm 4 runs in $O(n^{2.5})$.

Example 4. We solve the instance of Example 3 by algorithm 4.

We obtain the following optimal solution :

The number of batches is equal to 4 :
B_1 = $(J_1;J_2)$, $pb_1 = 4$;
B_2 = $(J_4;J_5)$, $pb_2 = 4$;
B_3 = (J_3), $pb_3 = 2$;
B_4 = (J_6), $pb_4 = 2$;
We obtain a makespan $C_{max} = 5$, the schedule is in FIGURE 6.

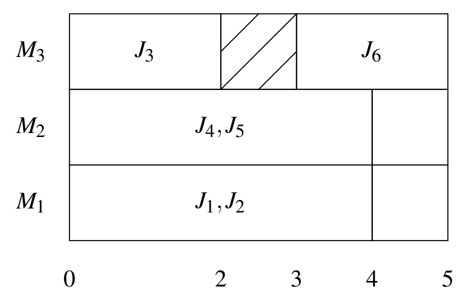

FIGURE 6 – Optimal schedule of Example 4

4. CONCLUSION

In this work we considered some polynomial case of the problem of minimizing the makespan on batch processing identical parallel machines. For futur research, we show the complexity of the NP-hard problems and we implement heuristics and exact methods for the solution of the problem.

5. REFERENCES

[1] Boudhar, M. and Finke, G. : *Scheduling on a batch machine with job compatibilities*, Belgian J. Oper. Res., Statist. Comput. Sci. (JORBEL) 40 (2000), 69–80.

[2] Boudhar, M. : *Static scheduling on a single batch processing machine with split compatibility graphs*, Cahiers du Laboratoire Leibniz-IMA Grenoble (France) 28, 2001.

[3] Boudhar, M. : *Dynamic scheduling on a single batch processing machine with split compatibility graphs*, J. Math. Modelling Algorithms 2 (2003), 17–35.

[4] Boudhar, M. : *Scheduling a batch processing machine with bipartite compatibility graphs*, Math. Methods Oper. Res. 57 (2003), 513–527.

[5] Brucker, P., Gladky, A., Hoogeveen, H., Kovalyov, M. Y., Potts, C., Tautenhahn, T. and Van De Velde, S. : *Scheduling a batching machine*, J. Scheduling 1 (1998), 31–54.

[6] Chandru, V., Lee, C. Y. and Uzsoy, R. : *Minimizing total completion time on batch processing machines*, Internat. J. Product. Res. 31 (1993), 2097–2121.

[7] H Chen, B Du, GQ Huang. *Scheduling a batch processing machine with non-identical job sizes : a clustering perspective*. International journal of production ...., 2011, Taylor and Francis.

[8] P Damodaran, K Srihari, SS Lam. *Scheduling a capacitated batch-processing machine to minimize makespan*. Robotics and Computer-Integrated ..., 2007 - Elsevier.

[9] Dobson, G. and Nambinadom, R. S. : *The batch loading and scheduling problem*, Research Report, Simon School of Business Administration, University of Rochester, NY, 1992.

[10] S.Even and O.Kariv. *AN* $O(N^{2.5})$ *ALGORITHM FOR MAXIMUM MATCHING IN GENERAL GRAPH*. Foundations of Computer Science(sfcs 1975 . . ., 1975 - ieeexplore.ieee.org).

[11] H.N. GABOW, 1974. *Implementation of Algorithms for Maximum Matching on Nonbipartite Graphs*. Ph.D. Thesis, Stanford University.

[12] Hochbaum, D. S. and Landy, D. : *Algorithms and heuristics for scheduling semiconductor burnin operations*, Research Report ESRC 94-8, University of California, Berkeley, USA, 1994.

[13] Kubiak, W. and Jolai Ghazvini, F. : *Minimizing earliness/tardiness criteria on a batch processing machine with job families*. Proc. Second Annual Internat. Conf. Industrial Engineering 2 (1997), 785–790.

[14] Mehta, S. V. and Uzsoy, R. : *Minimizing total tardiness on a batch processing machine with incompatible job families*, IIE Trans. Schedul. Logistics 31 (1998), 165–178.

[15] Potts, C. N. and Kovalyov, Y. K. : *Scheduling with batching* : A review, Europ. J. Oper. Res. 120 (2000), 228–249.

[16] Uzsoy, R. : *Scheduling batch processing machines with incompatible job families, Internat.* J. Product. Res. 33 (1995), 2685–2708.

[17] J Zhang, X Yao, Y Li. *Improved evolutionary algorithm for parallel batch processing machine scheduling in additive manufacturing*. International Journal of Production Research, 2020, Taylor and Francis.