Scheduling with Batch Compatible Tasks

Gerd Finke
Laboratoire G-SCOP, University Joseph Fourier Grenoble, France, gerd.finke@g-scop.inpg.fr

We analyze batch-scheduling problems that arise in connection with certain industrial applications. The models concern processing on a single max-batch machine with the additional feature that the tasks of the same batch have to be compatible. Compatibility is a symmetric binary relation – the compatible pairs are described with an undirected “compatibility graph”, which is often an interval graph according to some natural practical conditions that we present. We consider several models with varying batch capacities, processing times or compatibility graphs and summarize all known results.

Keywords: Batch-scheduling, Task compatibilities, Interval graphs, Bounded coloring.

1 Introduction

This presentation is mainly based on the article [7] and the thesis [12] and describes the developments of this branch since the original application [4] in 1999. A *batch machine* refers to a machine that can process several tasks simultaneously. We consider here the so-called *max-batch* or *parallel-batch* (sometimes abbreviated *p-batch*) machine, where the processing time of a group of tasks, called a *batch*, is the longest processing time of the tasks it contains. The initial motivation for this branch of scheduling theory was the scheduling of semiconductor burn-in operations [11]. Intensive research has subsequently been developed on this subject for various scheduling objectives and additional constraints, see for instance the surveys [5] and [15].

Here we focus on minimizing the makespan C_{max}. In addition, we assume that the tasks in the same batch have to be *compatible*, for instance they must share similar physical properties(form, weight, etc). The problem that we want to analyze may be formulated as follows. There are n independent tasks T_j ($j = 1, \ldots, n$) to be scheduled on a single max-batch machine. The batch machine has *capacity* b, which means that at most b tasks can be processed simultaneously (b may be finite or infinite). Each task T_j has a (minimal) processing time p_j. A batch B has processing time $p(B) = \max\{p_j : T_j \in B\}$ and all tasks in the same batch start and finish at the same time. Preemption is not allowed. Tasks in the same batch have to be *pairwise compatible*. This relation is represented by a *compatibility graph* $G = (V, E)$ where V is the set of tasks and a pair of tasks is an element of the edge set E if and only if they are compatible. By definition, a batch forms a *clique* (a complete subgraph, not necessarily maximal) in the compatibility graph G. Since all tasks have to be executed, the problem is to find a decomposition of G into cliques $B_1, B_2, \ldots, B_l$ where l is not known in advance, such that the schedule length $C_{max} = \sum_i p(B_i)$ is minimized. These batches may then be scheduled in any order, without any idle time.

The concept of scheduling with batch compatible tasks has been treated in [1]–[4], [7, 9, 12, 13] for general graphs and also for some special graphs. This theory is related to chromatic scheduling [6] where the complementary graph (graph of incompatibilities) is considered, leading to a graph coloring problem. For chromatic scheduling, there are usually no capacity constraints.

Our motivation for this type of batch scheduling problem originates in several industrial applications. The first one comes from the sheet metal industry [4], another from a rolling-mill [14], and finally from the process of tire making [13]. For these quite different industrial applications, we shall illustrate that interval graphs occur quite naturally as compatibility graphs in batch processing. In the following Section 2, the batch scheduling models are formulated that result from these applications.

2 Batch Scheduling Models

We consider batch scheduling problems on a single max-batch machine with task compatibilities. We use the notations:

$$1\,/\,p\text{-}batch,\ G = \beta_1,\ \beta_2\,/\,C_{max}$$

where the compatibility graph is specified by the parameter β_1; we use $\beta_1 = INT$ for an interval graph and $\beta_1 = (V, E)$ for a general graph. The parameter β_2 specifies the batch capacity b as follows: "$b = k$" for a fixed capacity; "$b < n$" for a variable finite capacity b which is part of the input; or β_2 is void for infinite capacity. There may be other parameters β_i representing additional restrictions, for instance structured processing times, release dates, etc. Referring to the various applications, one is lead to the following problem types.

(P1) $1\,/\,p\text{-}batch,\ G = INT,\ p_j = 1\,/\,C_{max}$

Problem $(P1)$ has also been solved approximately in [4], [9] for graphs that are slightly more general than interval graphs.

(P2) $1\,/\,p\text{-}batch,\ G = INT,\ b < n,\ p_j = 1\,/\,C_{max}$

We get two more models if we allow arbitrary processing times:

(P3) $1\,/\,p\text{-}batch,\ G = INT\,/\,C_{max}$

(P4) $1\,/\,p\text{-}batch,\ G = INT,\ b < n\,/\,C_{max}$

The aim of this presentation is to give an overview of the solution approaches to these problems and their extensions to more general graphs.

References

[1] M. Boudhar (2003), Scheduling a batch processing machine with bipartite compatibility graphs, *Mathematical Methods of Operations Research* **57**, 513 – 327.

[2] M. Boudhar (2003), Dynamic Scheduling on a Single Batch Processing Machine with Split-Compatibility Graphs, *J. of Mathematical Modelling and Algorithms* **2**, 17 – 35.

[3] M. Boudhar, G. Finke (2000), Scheduling on a batch machine with job compatibilities, *Belgian J. of Operations Research, Statistics and Computer Science* **40**, 69 – 80.

[4] N. Brauner, C. Dhaenens-Flipo, M.-L. Espinouse, G. Finke,H. Gavranovic (1999), Decomposition into parallel work phases with application to the sheet metal industry, *Proc. International Conf. on Industrial Engineering and Production Management (IEPM'99)* **1**, Glasgow, 389 – 396.

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

[6] M. Demange, D. de Werra, J. Monnot, V.T. Paschos (2001), Time slot scheduling of compatible jobs, *Cahier du Lamsade* **182**, Université Paris IX - Dauphine, Paris, OR Working Paper, EPF Lausanne.

[7] G. Finke, V. Jost, M. Queyranne, A. Sebő (2007), Batch processing with interval graph compatibilities between tasks, to appear in *DAM*.

[8] F. Gardi (2005), Ordonnancement avec exclusion mutuelle par un graphe d'intervalle ou une classe apparentée : complexité et algorithmes, *Ph.D. Thesis*, *Laboratoire d'informatique fondamentale - Faculté des sciences de Luminy, Marseille.*

[9] H. Gavranovic, G. Finke (2000), Graph partitioning and set covering for the optimal design of a production system in the metal industry, *Proc. Second Conference on Management and Control of Production and Logistics (MCPL'2000)* **2**, Grenoble, 603 – 608.

[10] P. Hansen, A. Hertz, J. Kuplinsky (1993), Bounded vertex colorings of graphs, *Discrete Mathematics* **111**, 305 – 312.

[11] D.S. Hochbaum, D. Landy (1997), Scheduling semiconductor burn-in operations to minimize total flowtime, *Operations Research* **45**, 874 – 885.

[12] V. Jost (2006), Ordonnancement chromatique: Polyèdres, Complexité et Classification, Ph.D. Thesis, *Université Joseph Fourier, Grenoble.*

[13] A. Kamgaing Kuiteing, A. Oulamara, G. Finke (2006), Flowshop scheduling with a batching machine and task compatibilities, *Proc. Conf. INCOM, St. Etienne.*

[14] P.L. Nguyen (1997), Planification tactique de la production : approche hiérarchisée pour une classe d'entreprises de sous-traitance et application au cas de laminage à froid, Ph.D. Thesis, *Université Joseph Fourier, Grenoble.*

[15] C.N. Potts, M.Y. Kovalyov (2000), Scheduling with batching: A review, *European Journal of Operational Research* **120**, 228 – 240.