A Constraint Programming Approach for a Batch Processing Problem with Non-identical Job Sizes

Arnaud Malapert[1,3], Christelle Guéret[2], and Louis-Martin Rousseau[3]

[1] École des Mines de Nantes, LINA UMR CNRS 6241
[2] École des Mines de Nantes, YRCCyN UMR CNRS 6597
[3] École Polytechnique de Montréal, CIRRELT
{arnaud.malapert,christelle.gueret}@mines-nantes.fr
louis-martin.rousseau@polymtl.ca

1 Introduction

This paper presents a constraint programming approach for a batch processing machine on which a finite number of jobs of non-identical sizes must be scheduled. A batch processing machine can process several jobs simultaneously. Such machines are encountered in chemical, pharmaceutical, aeronautical and semiconductor wafer industries where an oven, a drier or an autoclave is used during the process. These machines are often bottleneck because of their long processing times. Several papers have been proposed for identical job sizes and due date related performance measures. However, papers on problems with non-identical job sizes, concern mostly completion time related performance measures. For an extensive review on scheduling with batching, we refer the reader to Potts and Kovalyov [4]. Surprisingly, although they have been successfully used to solve various scheduling problems, only one constraint programming approach has been developed for this kind of problem (with sequence dependent setup times and job families).
In this paper, we propose a new constraint programming approach for a problem derived from a real application in aeronautical industry. To our knowledge, only two papers [2, 3] concern the resolution of this problem: the minimization of the maximum lateness of a batch processing machine with non-identical job sizes. In these papers, the authors propose a mathematical formulation and a branch-and-price. More formally, the problem can be described as follows. A set J of n jobs and one single batching machine with capacity b are given. Each job j is characterized by an integer triplet (p_j, d_j, s_j), where p_j is its processing time, d_j is its due date, and s_j is its size. The batch processing machine can process several jobs simultaneously as a batch as long as the sum of the sizes of the jobs that are in the batch does not exceed the capacity b of the machine. The processing time of a batch is equal to the longest processing time among the jobs in the batch. The completion time C_j of a job j is the completion time of the batch to which it belongs. The machine and jobs are assumed to be continuously available from time zero onwards, or equivalently, they have equal releases dates. Once the processing

of a batch has been initiated, no job can be removed from or added to the batch. The objective is to minimize the maximum lateness $L_{max} = \max_{1 \leq j \leq n}(C_j - d_j)$. This problem, denoted by $1|p-batch, b < n, non-identical|L_{max}$, is unary NP-hard since Brucker et al. [1] proved that the same problem with identical job sizes is unary NP-hard.

2 Constraint programming approach

The constraint programming formulation relies on the decomposition of the problem into finding an assignment of the jobs to the batches, and then minimizing the maximal lateness of the batches on a single machine.

The problem of assigning the jobs to the batches is equivalent to the *one-dimensional bin packing problem* defined as follows: given n indivisible items (jobs), each of a known non-negative size s_j, and m bins (batches), each of capacity b, can we pack the n items into the m bins such that the sum of the sizes of the items in any bin is not greater than b ? This problem has been shown NP-complete. Our formulation uses a global constraint inspired by Shaw [5].

Then, once the jobs are packed into the batches, the problem of scheduling the batches is equivalent to minimizing the maximal lateness of a set of jobs (batches) on a single machine. This problem, known as $1||L_{max}$, is polynomially solvable: an optimal schedule is obtained by applying Jackson's scheduling rule, also known as the earliest due date (EDD-)rule which schedules the tasks in order of non decreasing due dates. When coping with optimization problems, pruning can be done also on the basis of costs, *i.e.* optimality reasoning. The new constraint `sequenceEDD` uses a relaxation of the problem that yields a lower bound for the objective function to prune portions of the search space. The general idea is to infer primitive constraints on the basis of information on costs. We use optimization components within a global constraint representing a proper relaxation of the problem, which consists of minimizing the maximal lateness of batches on a single machine. The optimization components provide the optimal solution of the relaxed problem, its value and a gradient function computing the cost to be added to the optimal solution for some variable-value assignments. The optimal value of this solution improves the lower bound of the objective function and prunes portions of the search space whose lower bound is bigger than the best solution found so far.

At each node in the search tree, the branching selects a job, and assigns this job to a batch. We chose the variable selection heuristic called *complete decreasing* which packs jobs in order of non-increasing size. We propose a new value selection heuristic called *batch fit* which selects an available bin having the least impact on the schedule of the batches.

3 Experimental results

This section summarizes computational experiments conducted to evaluate our approach. Our algorithm has been tested on randomly generated instances ranging from 10 to 100 jobs. Our implementation is based on the `choco` solver. All the experiments were conducted on a cluster of Linux machines, each node with 48 GB of RAM and two quad core 2.4 GHz processor. A first set of experiments (not detailed here) show that the use of the constraint `sequenceEDD` as well as the use of the value selection heuristic *batch fit* reduce the computation time and improve solution quality.

Then, our approach is compared to a mathematical formulation solved by `IBM Ilog Cplex` 11.2.1. For small instances ($n \leq 50$), our approach outperforms the mathematical formulation: it solves more instances with solution times that are orders of magnitude lower. Figure 1 compares the quality gap on instances with strictly more than 50 jobs. The quality gap of a solution ub for a given instance is estimated by $(ub + d_{max}) \div (lb + d_{max})$. The time limit of `choco` is 1 hour, whereas it is increased until 12 hours for `Ilog Cplex`. Each point represents one instance and its x coordinate is the quality gap that we obtained, whereas its y coordinate is the quality gap obtained with `Ilog Cplex`. All points located above the line ($x = y$) are upper bounds improved by the use of our approach. Despite a smaller time limit, the constraint programming approach provides better solution than the mathematical formulation on the vast majority of instances. Furthermore, the difference between the two upper bounds is very tight when the mathematical formulation found the best upper bound, whereas it can be significant when the constraint approach did.

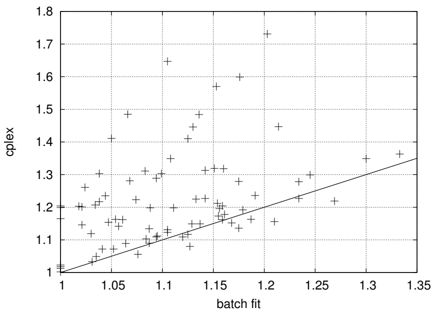

Fig. 1. Comparison to the mathematical formulation.

4 Conclusion

We have presented a constraint programming approach to schedule a batch processing machine on which a finite number of jobs of non-identical sizes must be scheduled. This approach exploits an optimization constraint based on a relaxed problem which applies cost based domain filtering rules and is enhanced with a dedicated branching heuristics.

Computational results demonstrate the positive effect of each component and give better solution with computation times that are orders of magnitude lower than a mathematical formulation based on bin packing and sequencing models.

In further research, we will apply our approach to problems with completion time related measures (C_{max}, $\sum C_j$, $\sum w_j C_j$). In addition, subsequent research topics include the study of parallel batching machines and additional constraints, for instance job release dates that remain incompatible with our approach.

Bibliography

[1] Brucker, P., Gladky, A., Hoogeveen, H., Koyalyov, M., Potts, C., Tautenham, T., van de Velde, S.: Scheduling a batching machine. Journal of Scheduling 1(1) (Jun 1998)

[2] Daste, D., Gueret, C., Lahlou, C.: A Branch-and-Price algorithm to minimize the maximum lateness on a batch processing machine. In: Proceedings of the 11th international workshop on Project Management and Scheduling PMS'08. pp. 64–69. IStanbul Turquie (2008)

[3] Daste, D., Gueret, C., Lahlou, C.: Génération de colonnes pour l'ordonnancement d'une machine à traitement par fournées. In: 7ième conférence internationale de modélisation et simulation MOSIM'08. vol. 3, pp. 1783–1790. Paris France (2008)

[4] Potts, C.N., Kovalyov, M.Y.: Scheduling with batching: A review. European Journal of Operational Research 120(2), 228–249 (January 2000)

[5] Shaw, P.: A constraint for bin packing. In: Wallace, M. (ed.) Principles and Practice of Constraint Programming - CP 2004, 10th International Conference, CP 2004, Toronto, Canada, September 27 - October 1, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3258, pp. 648–662. Springer (2004)