A Benders decomposition for the flexible cyclic jobshop problem

Félix Quinton[1], Idir Hamaz[2] and Laurent Houssin[3]

[1] ONERA – The French Aerospace Lab, F-31055, Toulouse, France
felix.quinton@onera.fr
[2] LIRMM UMR 5506, Université de Montpellier, 34932 Montpelier Cedex 5, France
idir.hamaz@lirmm.fr
[3] LAAS-CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France
houssin@laas.fr

Keywords: Cyclic scheduling, Decomposition method, Flexible scheduling.

1 Introduction

This paper tackles the Flexible Cyclic Jobshop Scheduling Problem (FCJSP). We propose a Mixed Integer Linear Programming (MILP) formulation for the FCJSP along with a Benders decomposition algorithm adapted for the FCJSP. The Basic Cyclic Scheduling Problem (BCSP) is a generalisation of the basic scheduling problem where a schedule of elementary tasks $i \in \mathcal{T} = \{1, ..., n\}$ is infinitely repeated. This problem has been studied a lot since it has applications in many fields such as parallel processing ([Hanen and Munier, 1995]), staff scheduling ([Karp and Orlin, 1981]), or robotic scheduling ([Kats and Levner, 1996]). [Hamaz et al., 2018a] studied the BCSP where the processing times are affected by an uncertainty effect.

To specify a BCSP, we are given a set of elementary tasks $i \in \mathcal{T}$ and their respective execution time p_i. Also, we define $t_i(k) \geq 0$ the starting time of occurrence k of task $i \in \mathcal{T}$. The order in which the tasks are to be executed is fixed by precedence constraints. A precedence constraint indicating that task i must be completed before starting the execution for task j can be expressed as a quadruplet (i, j, p_i, H_{ij}), where H_{ij} is called the height of the precedence constraint and represents the occurrence shift between tasks i and j. In simple terms, it states that the $k + H_{ij}$-th occurrence of task j cannot start before the end of the k-th occurrence of task i. Precisely, it states that : $t_j(k + H_{ij}) \geq p_i + t_i(k)$.

2 The cyclic jobshop scheduling problem

In the Cyclic Jobshop Scheduling Problem (CJSP), each elementary task $i \in \mathcal{T} = \{1, ..., n\}$ is assigned to a machine $r_i \in \mathcal{R} = \{1, ..., R\}$, with $R < n$ and elementary tasks linked by precedence constraints constitute jobs. For instance, in a manufacturing context, a job might represent the manufacturing process of a product as a whole, while an elementary task represents only one step of the manufacturing process.

Due to its numerous applications in large-scale production scheduling, the CJSP has received at lot of attention from the research community. [Hanen, 1994] proposed a MILP for this problem and proved some of its properties. [Brucker et al., 2012] also proposed a MILP for a CJSP with transportation by a single robotic cell. [Draper et al., 1999] proposed an alternative formulation to solve CJSP problems as constraint satisfaction problems. [Hamaz et al., 2018b] studied the CJSP with uncertain processing time. The inclusion of machines in the CJSP leads to a lack of resources, since the tasks are competing for the working time of the machines. This lack of resources is represented by disjunction constraints, which state that for a pair of tasks $(i, j) \in \mathcal{T}^2, i \neq j$, that must be executed on the same machine, i.e. $r_i = r_j$, an occurrence of i and an occurrence of j, cannot be executed at the same time. In the following of this paper, we will denote by $\mathcal{D} = \{(i,j)|R(i) \cap R(j) \neq \emptyset\}$ the set of pairs of tasks linked by disjunction constraints. Mathematically, the disjunction between two tasks $(i,j) \in \mathcal{T}^2, i \neq j$ is modeled with the two following disjunction constraints (1):

$$t_j(k + K_{ij}) \geq t_i(k) + p_i \qquad \text{and} \qquad t_i(k + K_{ji}) \geq t_j(k) + p_j. \tag{1}$$

where K_{ij} (resp. K_{ji}) is the height of the disjunction constraint, i.e. the occurrence shift between tasks i and j (resp. j and i). It has been proven by [Hanen, 1994] that a feasible schedule for a CJSP must satisfy $K_{ij} + K_{ji} = 1$.

Note that in this problem the variables are the cycle time α, the starting times of each elementary tasks $(t)_{i\in\mathcal{T}}$, and the heights of the disjunctive constraints, $(K)_{(i,j)\in D}$. A feature of the CJSP is the Work In Process (WIP). It corresponds to the maximum number of occurrences of a job processed simultaneously. Mathematically, the role of the WIP can be modelled as the height of the precedence constraint from fictive task e to fictive task s, and can be explained by the equation :

$$s(k) \geq e(k - WIP).$$

In our study, we aim at minimizing the cycle time α, so the WIP and the $H_{ij}, (i,j) \in \mathcal{E}$ are given. Modelling of the CJSP for the minimisation of the cycle time α with known heights as been proposed by [Hanen, 1994] and is used by [Brucker et al., 2012] to solve a CJSP with transportation.

[Hanen, 1994] proposes to define the variable $\tau = \frac{1}{\alpha}$ and for all $i \in \mathcal{T}$, the variable $u_i = \frac{t_i}{\alpha}$. Then CJSP can then be considered as a MILP in the following form:

$$\max \tau$$

s.t.

$$\tau \leq \frac{1}{p_i}, \quad \forall i \in \mathcal{T} \tag{2a}$$

$$u_j + H_{i,j} \geq u_i + \tau p_i, \quad \forall (i,j) \in \mathcal{E} \tag{2b}$$

$$u_j + K(i,j) \geq u_i + \tau p_i, \quad \forall (i,j) \in \mathcal{D} \tag{2c}$$

$$K(i,j) + K(j,i) = 1, \quad \forall (i,j) \in \mathcal{D} \tag{2d}$$

$$K(i,j) \in \mathbb{Z}, \quad \forall (i,j) \in \mathcal{D} \tag{2e}$$

$$u_i \geq 0, \quad \forall i \in \mathcal{T}. \tag{2f}$$

The CJSP can then be solved by writing the problem as a MILP, which can be solved using mathematical programming or using a dedicated Branch-and-Bound procedure ([Fink et al., 2012], [Hanen, 1994]).

3 The flexible cyclic jobshop scheduling problem

The Flexible Cyclic Jobshop Scheduling Problem (FCJSP) is a CJSP where the elementary tasks are flexible. This means that the execution of a task $i \in \mathcal{T}$, is assigned to exactly one machine r in a set of machines that is a subset of the set of machines $\mathcal{R}$ specific to task i. This subset is denoted $R(i) \subset \mathcal{R}$. We model the assignment of a task $i \in \mathcal{T}$ to a machine $r \in R(i)$ as a decision variable $m_{i,r}$ defined as follows :

$$\forall i \in \mathcal{T}, \forall r \in R(i), \quad m_{i,r} = \begin{cases} 1 \text{ if task } i \text{ is assigned to machine } r \\ 0 \text{ otherwise.} \end{cases}$$

Each assignment of a task $i \in \mathcal{T}$ to a machine $r \in R(i)$ is associated with a given execution time denoted p_{ir}. Also, because we do not know *a priori* on which machine each task will be assigned, we do not know the set $(i, j) \in \mathcal{T}^2, i \neq j, R(i, j) \neq \emptyset$ of pairs of tasks which are connected by a disjunctive constraint.

Based on the model of Section 2 and variables $m_{i,r}$, we have proposed a MILP for the FCJSP. A first model was proposed in [Quinton et al., 2018] but substantial improvements have been made since this first model.

4 A Benders decomposition for the FCJSP

The FCJSP formulated as a MILP can be very hard to solve. Using CPLEX, difficult instances with a large number of tasks or with few robots might exceed any reasonable time limit. To tackle this issue, we propose a Benders decomposition for the FCJSP. In the usual Benders decomposition scheme, two problems coexist: the master problem and the sub-problem. The Master Problem (MP) consists in an integer linear problem composed of the constraints from the model described in Section 3 involving only the integer variables, and the optimality cuts generated at each iteration of the Benders algorithm. The remaining constraints, involving only continuous variables or a combination of continuous and integer variables, compose the primal sub-problem. It can be written as a linear problem. The full description of the algorithm is available in [Quinton et al., 2019].

5 Numerical results

The MILP solving is very efficient for the easiest instances (10 tasks and 5 machines). For those easy instances, it is much more efficient than the Benders decomposition. For instances of average difficulty with 10 tasks and 4 machines, the MILP is still more efficient than the Benders decomposition, but we can remark that the execution time of the MILP is increasing much faster than the execution time of the Benders decomposition. Finally, for the hard instances with 10 tasks and 3 machines, our Benders decomposition is always faster to find the optimal solution than the MILP. From these results, we learn that it is better to use the MILP for easy problems with numerous machines such as our instances with 10 tasks and 5 machines. However, for hard instances with a considerable number of disjunctions, such as the instances with 10 tasks and 3 machines, the execution times of the MILP rocket up and it is much better to use the Benders decomposition to obtain an optimal solution or a heuristic procedure to obtain a feasible solution (not presented here).

6 Conclusion

We have proposed a MILP for the FCJSP where the objective is the minimisation of the cycle time. The problem is highly combinatorial, so we also proposed a Benders decomposition algorithm that is more efficient for difficult instances. The Benders decomposition includes specific cuts to ensure the feasibility of the integer solution. Numerical instances have shown that the MILP becomes inefficient for difficult instances with many disjunctions. Also, if an optimal solution is required, our Benders decomposition is more efficient than the MILP for this type of instances.

References

[Brucker et al., 2012] Brucker, P., Burke, E. K., and Groenemeyer, S. (2012). A mixed integer programming model for the cyclic job-shop problem with transportation. *Discrete Applied Mathematics*, 160(13-14):1924–1935.

[Draper et al., 1999] Draper, D. L., Jonsson, A. K., Clements, D. P., and Joslin, D. E. (1999). Cyclic scheduling. In *IJCAI*, pages 1016–1021. Citeseer.

[Fink et al., 2012] Fink, M., Rahhou, T. B., and Houssin, L. (2012). A new procedure for the cyclic job shop problem. *IFAC Proceedings Volumes*, 45(6):69–74.

[Hamaz et al., 2018a] Hamaz, I., Houssin, L., and Cafieri, S. (2018a). A robust basic cyclic scheduling problem. *EURO Journal on Computational Optimization*, 6(3):291–313.

[Hamaz et al., 2018b] Hamaz, I., Houssin, L., and Cafieri, S. (2018b). The Cyclic Job Shop Problem with uncertain processing times. In *16th International Conference on Project Management and Scheduling (PMS 2018)*, Rome, Italy.

[Hanen, 1994] Hanen, C. (1994). Study of a np-hard cyclic scheduling problem: The recurrent job-shop. *European journal of operational research*, 72(1):82–101.

[Hanen and Munier, 1995] Hanen, C. and Munier, A. (1995). A study of the cyclic scheduling problem on parallel processors. *Discrete Applied Mathematics*, 57(2-3):167–192.

[Karp and Orlin, 1981] Karp, R. M. and Orlin, J. B. (1981). Parametric shortest path algorithms with an application to cyclic staffing. *Discrete Applied Mathematics*, 3(1):37–45.

[Kats and Levner, 1996] Kats, V. and Levner, E. (1996). Polynomial algorithms for scheduling of robots. *Intelligent Scheduling of Robots and FMS*, pages 77–100.

[Quinton et al., 2018] Quinton, F., Hamaz, I., and Houssin, L. (2018). Algorithms for the flexible cyclic jobshop problem. In *14th IEEE International Conference on Automation Science and Engineering, CASE 2018, Munich, Germany, August 20-24, 2018*, pages 945–950.

[Quinton et al., 2019] Quinton, F., Hamaz, I., and Houssin, L. (2019). A mixed integer linear programming modelling for the flexible cyclic jobshop problem. *Annals of Operations Research*.