The Cyclic Job Shop Problem with uncertain processing times

Idir Hamaz, Laurent Houssin, Sonia Cafieri

The Cyclic Job Shop Problem with uncertain processing times

Idir Hamaz[1], Laurent Houssin[1] and Sonia Cafieri[2]

[1] LAAS-CNRS, Universite de Toulouse, CNRS, UPS, Toulouse, France
{ihamaz, lhoussin}@laas.fr
[2] ENAC, Universite de Toulouse, F-31055 Toulouse, France
sonia.cafieri@enac.fr

Keywords: Cyclic scheduling, budgeted uncertainty set, robust optimization.

1 Introduction

Most models for scheduling problems assume deterministic parameters. In contrast, real world scheduling problems are often subject to many sources of uncertainty, for example activities duration can decrease or increase, machines can break down, new activities can be incorporated, *etc.* In this paper, we focus on scheduling problems that are cyclic and where activity durations are affected by uncertainty. Indeed, the best solution for a deterministic problem can quickly become the worst one in the presence of uncertainties.

In this paper, we consider the *Cyclic Job Shop Problem* (CJSP) where processing times are affected by uncertainty. Several studies were conducted on the deterministic CJSP. The CJSP with identical parts is studied in (Roundy, R. 1992). The author shows that the problem is NP-hard and designs a branch and bound algorithm to solve the problem. Hanen (1994) investigates the general CJSP and presents a branch and bound procedure to tackle the problem. A general framework for modeling and solving cyclic scheduling problems is presented in (Brucker, P. and Kampmeyer, T. 2008). The authors present different models for cyclic versions of the job shop problem. However, a few works consider cyclic scheduling problems under uncertainty. Che, A. *et. al.* (2015) investigate the cyclic hoist scheduling problem with processing time window constraints where the hoist transportation times are uncertain. The authors define a robustness measure for cyclic hoist schedule and a bi-objective mixed integer linear program to optimize the cycle time and the robustness.

In order to deal with uncertainty, we use a robust optimization approach. We model the uncertain parameters by using the idea of uncertainty set proposed by Bertsimas and Sim (2004). Each task duration belongs to an interval, and the number of parameters that can deviate from their nominal values is bounded by a positive number called *budget of uncertainty*. This parameter allows us to control the degree of conservatism of the resulting schedule. Finally, we propose a branch and bound procedure that computes the minimum cycle time for the robust CJSP such that, for each scenario in the uncertainty set, there exists a feasible cyclic schedule.

2 Problems description

2.1 Basic Cyclic Scheduling Problem (BCSP)

We are given a set of n generic operations $\mathcal{T} = \{1, ..., n\}$. Each operation $i \in \mathcal{T}$ is characterized by a non-negative processing time p_i and has to be performed infinitely often without preemption. We denote $< i, k >$ the k^{th} occurrence of the generic operation i and $t(i, k)$ the starting time of k^{th} occurrence of the operation i.

The operations are subjected to a set of *precedence constraints* (uniform constraints). The constraints between the occurrences $< i, k >$ and $< j, k + H_{ij} >$ are given by

$$t(i,k) + p_i \leqslant t(j, k + H_{ij}), \quad \forall\, i \in \mathcal{T},\ \forall\, k \geq 1 \tag{1}$$

where H_{ij} is an integer that represents the depth of the occurrence shift, usually referred to as *height*. The H_{ij} parameter is an occurrence shift between the operations i and j. For instance, for each execution of the occurrence $< i, k >$, the next execution of j is the occurrence $< j, k + H_{ij} >$.

A schedule S is an assignment of starting time $t(i, k)$ for each occurrence $< i, k >$ of task $i \in \mathcal{T}$. Such schedule is called *periodic* with cycle time α if it satisfies

$$t(i,k) = t(i,0) + \alpha k, \quad \forall\, i \in \mathcal{T},\ \forall\, k \geq 1 \tag{2}$$

where α is the cycle time and represents the difference between the stating times of two successive occurrences of the same task.

Therefore, a schedule S can be entirely defined by the staring times $t_i = t(i, 0)$ of the first occurrences and the cycle time.

In this study, the objective is to minimize the cycle time α while satisfying the precedence constraints between operations. Notice that different objective functions exist for cyclic scheduling problems, such as work in progress minimization or both cycle time and work in progress minimization.

A bi-valued directed graph $G = (\mathcal{T}, U)$ can be associated with any instance of BCSP. In this graph, a node (resp. an arc) of G corresponds to a generic operation (resp. constraints) in the BCSP. Each arc (i, j) of G has two valuations, the length $L_{ij} = p_i$ and the height H_{ij}. These arcs are called uniform arcs and are built by considering the precedence constraints. For instance, a precedence constraint between task i and task j leads to an arc (i, j) of G labeled with $L_{ij} = p_i$ and H_{ij}. We denote $H(c)$ (resp. $L(c)$) the height (resp. length) of a circuit c in graph G the sum of heights (resp. lengths) of the arcs composing the circuit c.

The minimum cycle time is given by the maximum circuit ratio of the graph which is defined by

$$\alpha = \max_{c \in \mathcal{C}} \frac{\sum_{(i,j) \in c} L_{ij}}{\sum_{(i,j) \in c} H_{ij}}$$

where $\mathcal{C}$ is the set of all circuits in G.

We call *critical circuit* the circuit c realizing the maximum circuit ratio. Several algorithms have been proposed for the computation of critical circuits. An experimental study about maximum circuit ratio algorithms was published in (Dasdan, A. 2004). The author remarks that, among the several tested algorithms, the most efficient one is the Howard's algorithm. Although the algorithm has a pseudo-polynomial complexity, it shows noteworthy practical results.

Once the cycle time is determined, the starting times $(t_i)_{i \in \mathcal{T}}$ can be determined by computing the longest path in the graph G where each arc $(i, j) \in U$ is valued with $p_i - \alpha H_{ij}$.

2.2 Cyclic Job Shop Problem (CJSP)

In the present work, we focus on the cyclic job shop problem (CJSP). The difference with the problem defined above is that for CJSP the number of machines is lower than the number of tasks to perform. As a result, the same resource must be shared between different operations. A CJSP can be considered as a BCSP equipped with resource constraints.

Each occurrence of an operation $i \in \mathcal{T}$ has to be executed, without preemption, on the machine $M_{(i)} \in \mathcal{M} = \{1, ..., m\}$. Operations are grouped on a set of jobs $\mathcal{J}$, where a job j represents a sequence of elementary operations that must be executed in order. To avoid overlapping between the tasks executed on the same machine, for each pair of operations i and j where $M_{(i)} = M_{(j)}$, the following *disjunctive constraint* holds

$$\forall\, i, j \ s.t.\ M_{(i)} = M_{(j)},\ \forall k, l \in \mathbb{N} : t(i,k) \leq t(j,l) \Rightarrow t(i,k) + p_i \leq t(j,l) \tag{3}$$

In summary, a cyclic job shop problem is defined by

- a set $\mathcal{T}$ of elementary tasks,
- a set $\mathcal{M}$ of machines,
- for each task $i \in \mathcal{T}$, a processing time p_i and a machine $M_{(i)} \in \mathcal{M}$ on which the task has to be performed,
- a set $\mathcal{P}$ of precedence constraints,
- a set $\mathcal{D}$ of disjunctive constraints that occur when two tasks are mapped on the same machine,
- a set $\mathcal{J}$ of jobs corresponding to a production sequence of generic operations. More precisely, a job J_1 defines a sequence $J_1 = t_{1,1} \dots t_{1,k}$ to be executed in that order.

The CJSP can be represented by directed graph $G = (V, \mathcal{P} \cup \mathcal{D})$, called *disjunctive graph*. The sequence of operations that belongs to the same job are linked by uniform arcs in $\mathcal{P}$ where the heights are equal to 0. Additionally, for each pair of generic operations i and j executed on the same machine, a disjunctive pair of arcs (i, j) and (j, i) occurs. These arcs are labeled respectively with $L_{ij} = p_i$ and $H_{ij} = K_{ij}$, and $L_{ji} = p_j$ and $H_{ji} = K_{ji}$ where K_{ij} is an occurrence shift variable that satisfies $K_{ij} + K_{ji} = 1$ (Hanen C 1994).

The following bounds on occurrence shift variables K_{ij} have been proposed in (Hanen C 1994):

$$K_{ij}^{-} \leq K_{ij} \leq 1 - K_{ij}^{-}. \tag{4}$$

with

$$K_{ij}^{-} = 1 - min\{H(\mu) \,|\, \mu \text{ } from\ j\ to\ i\ in\ G\}. \tag{5}$$

A schedule is an assignment of all the occurrence shifts, i.e., determine precedence relations on the operation occurrences mapped to the same machine. Note that once the occurrence shifts are determined the problem is equivalent to the BCSP, therefore, the minimum cycle time can be obtained by the cited algorithms.

Previous studies have shown that the problem is NP-Hard (Hanen C 1994) for cycle time minimization.

2.3 Robust Cyclic Job Shop Problem (RCJSP)

In this paper, we investigate the robust version of the CJSP. More precisely, we are interested in the CJSP where processing times are affected by uncertainty and belong to a finite uncertainty set $\mathcal{U}$. Based on the *budget of uncertainty* concept introduced in (Bertsimas, D. and Sim, M. 2004), the processing time deviations can be modeled trough the following uncertainty set:

$$\mathcal{U}^{\Gamma} = \left\{ (p_i)_{i \in \mathcal{T}} \in \mathbb{R}^n : p_i = \bar{p}_i + \hat{p}_i \xi_i,\ \forall\, i \in \mathcal{T};\ \xi_i \in \{0,1\};\ \sum_{i \in \mathcal{T}} \xi_i \leq \Gamma \right\}$$

where $\bar{p}_i$ represents the nominal processing time of operation i and $\hat{p}_i$ its deviation. The parameter Γ is a positive integer and represents an upper bound on the number of processing times deviating from their nominal value.

The objective of the problem is to find, for a given budget of uncertainty Γ, the minimum cycle time such that, for each $p \in \mathcal{U}^\Gamma$, there exists a vector $(t(p)_i)_{i\in\mathcal{T}}$ satisfying both the precedence and disjunctive constraints.

3 Branch and bound procedure for the RCJSP

Recently, an Howard's algorithm adaptation taking into account the uncertainty set $\mathcal{U}^\Gamma$ has been presented in (Hamaz, I. *et. al.* 2017). The computational experiments on the algorithm show small execution times for robust BCSP instances.

To take into account the uncertainty on the processing times for the RCJSP, we develop a branch and bound procedure that uses the robust version of the Howard's algorithm. The procedure starts by initializing the upper bound on the cycle time to $\sum_{i\in\mathcal{T}} p_i + p_f$ where p_f is the sum of the first Γ greatest deviations and the lower bound to the optimal cycle time of $G = (\mathcal{T}, U)$ computed by the Howard's algorithm adaptation.

We use the same branching scheme as in (Fink, M. *et. al.* 2012). The search tree is initialized with a node (the root) where the graph $G = (\mathcal{T}, U)$ contains only the uniform arcs U and no fixed disjunctions. Then, the branching is performed on unfixed disjunctions K_{ij}. For this purpose, a successor node is created for each value on the interval $[K_{ij}^-, 1-K_{ij}^-]$. The value of the node is then computed by running the robust version of the Howard's algorithm with $G = (\mathcal{T}, U^{'} \cup \{(i,j),(j,i)\})$, where $U^{'}$ contains the uniform arcs and a precedent fixed disjunctive arcs. When all the occurrence shifts are fixed, a feasible schedule is obtained, then the upper bound can be updated.

Preliminary numerical results show that the branch and bound procedure (implemented in C++ and executed on an Intel Xeon E5-2695 processor running at 2.30GHz CPU) delivers promising results. Besides, the algorithm is insensitive regarding the value of the budget of uncertainty.

Once the optimal cycle time computed by the branch and bound procedure, a periodic schedule $S^\Gamma = (\alpha, ((t(p)_i)_{i\in\mathcal{T}})$ can be determined for each $p \in \mathcal{U}^\Gamma$.

4 Conclusion

The RCJSP with budgeted uncertainty set is addressed in this paper. We present a branch and bound procedure that uses a Howard's algorithm adaptation. Further investigation will address dominance rules to speed up the branch and bound procedure.

References

Bertsimas, D., Sim, M., 2004, "The price of robustness", *Operations research*, Vol. 52(1), pp. 35-53.

Brucker, P., Kampmeyer, T., 2008, "A general model for cyclic machine scheduling problems", *Discrete Applied Mathematics*, Vol. 156(13), pp. 2561-2572.

Che, A., Feng, J., Chen, H., and Chu, C., 2015, "Robust optimization for the cyclic hoist scheduling problem", *European Journal of Operational Research*, Vol. 240(3), pp. 627-636.

Dasdan, A., 2004, "Experimental analysis of the fastest optimum cycle ratio and mean algorithms", *ACM Transactions on Design Automation of Electronic Systems*, Vol. 9(4), pp. 385-418.

Fink, M., Rahhou, T. B. and Houssin, L., 2012, "A new procedure for the cyclic job shop problem", *IFAC Proceedings Volumes*, Vol. 45(6), pp. 69-74.

Hamaz, I., Houssin, L. and Cafieri, S., 2017, "Robust Basic Cyclic Scheduling Problem", *Manuscript submitted for publication.*

Hanen, C., 1994, "Study of a NP-hard cyclic scheduling problem: The recurrent job-shop", *European journal of operational research*, Vol. 72(1), pp. 82-101.

Roundy, R., 1992, "Cyclic schedules for job shops with identical jobs", *Mathematics of operations research*, Vol. 17(4), pp. 842-865.