The time varying cyclic job shop problem

Laurent Houssin, Idir Hamaz, Sonia Cafieri

The time varying cyclic job shop problem

Laurent Houssin[1,2], Idir Hamaz[1,2], Sonia Cafieri[3]

[1] CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
[2] Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France
[3] ENAC, MAIAA, F-31055 Toulouse, France
University of Toulouse, IMT, F-31400 Toulouse, France
laurent.houssin@laas.fr

Keywords: cyclic scheduling, robust scheduling.

1 Introduction

In classical scheduling, a set of tasks is executed once while the determined schedule optimizes objective functions such as the makespan or earliness-tardiness. In contrast, cyclic scheduling means performing a set of generic tasks infinitely often while minimizing the time between two occurrences of the same task.Cyclic scheduling has several applications, e.g. in robotic industry, in manufacturing systems or multiprocessor computing. It has been studied from multiple perspectives, since there exist several possible representations of the problem such as graph theory, mixed integer linear programming, Petri nets or $(max, +)$ algebra. An overview about cyclic scheduling problems and the different approaches can be found in [Hanen and Munier, 1995a] and [Brucker and Kampmeyer, 2008]. However, few works have tackled the robust version of this problem although the literature for robust classical scheduling problem is large. In this paper, we consider a subset of tasks that have variable processing times. Besides, the minimum processing time and the maximum processing time of these tasks are known. We propose a method that finds the schedule which minimizes the expected cycle time. We show that the evaluation of a schedule can be considered as a volume calculus of polyhedron. More precisely, for each schedule we derive a set of polyhedron and the integral calculus (based on the Vinci library [Bueler et al., 2000]) of their volume leads to an evaluation of the performance of the considered schedule. This approach enables to derive a branch and bound procedure to choose the best schedule in the sense of the smallest cycle time with respect to the variability of the processing times.

2 Cyclic scheduling problems

The basic cyclic scheduling problem (BCS) involves generic tasks and precedence constraints between tasks but no resource constraints are considered. In this problem, a set of n *generic operations* are processed in parallel by an unbounded number of machines and there is a set of q precedence constraints. We denote this set $\mathcal{A} = \{a_1, \ldots, a_q\}$ where a_l corresponds to a constraint represented by a triple (i, j, h). More precisely, the *uniform constraint* (i, j, h) means that

$$s_i(k) + p_i \leq s_j(k + h), \quad \forall k \geq 1$$

where s_i denotes the beginning of the operation i and p_i is the processing time of i. In this framework, the asymptotic cycle time $\alpha(S)$ is usually minimized (with S a feasible schedule). Equivalently we can aim at maximizing the throughput $r(S) = \frac{1}{\alpha(S)}$.

We focus on the 1-periodic schedules (or periodic schedules in short). These schedules are characterized by a period α such that

$$s_i(k + 1) = \alpha + s_i(k), \quad \forall i \in \mathcal{T}, \quad \forall k \geq 1,$$

where $\mathcal{T}$ is the set of tasks to perform.

For our concern, we are interested in the cyclic job shop problem (CJSP). In this case, tasks are *a priori* mapped onto machines and the number of machines is smaller than the number of tasks to perform . More precisely, a cyclic job shop is defined by

- a set $\mathcal{T}$ of elementary tasks,
- a set $\mathcal{R}$ of machines,
- for each task $t \in \mathcal{T}$, a processing time p_t and a machine $m_t \in \mathcal{R}$ on which the task has to be performed,
- a set of jobs $\mathcal{J}$ corresponding to a production sequence of elementary tasks. More precisely, a job J_1 defines a sequence $J_1 = t_{11} \dots t_{1k}$ to be executed in that order.

A directed graph $G = (V, E)$ can be associated with a CJSP such that a node (resp. an arc) of G corresponds to a task (resp. constraints) in the CJSP. Each arc (i, j) of G is equipped with two values L_{ij} and H_{ij}. Two kinds of arcs can be differentiated.

The uniform arcs are built by considering the precedence constraints of tasks belonging to the same job. For instance, the sequence $t_i t_j$ in a job leads to an arc (i, j) of G labeled with $L_{ij} = p_i$ and $H_{ij} = 0$.

A disjunctive pair of arcs (i, j) and (j, i) occurs when the task i and the task j are mapped on the same machine. In this case, $L_{ij} = p_i$, $L_{ji} = p_j$ and we have $K_{ij} + K_{ji} = 1$ (see [Hanen, 1994] for further details). Two dummy nodes are introduced in the model. More precisely, $s_s(k)$ represents the start of an occurrence k and $s_e(k)$ represents the end of this occurrence. The arc between these two nodes is valuated with no processing time and with a positive height denoted H^*.

Since we want to minimize the cycle time α of the process, we consider the earliest functionning rule. It leads to:

$$s_s(k) = s_e(k - H^*) \quad = -\alpha H^* + s_e(k) \tag{1}$$

It means that the occurrence number k can start after the end of the occurrence number $k - H^*$. Hence, H^* describes the maximum Work In Process (WIP) of the system. Several works in the literature aim at minimizing this quantity but in our case, we consider a fixed value for H^*.

A schedule is an assignement of all the occurrence shifts. A well known result in cyclic scheduling is that the minimum cycle time of the system is given by the maximum mean cycle of the graph that is defined by

$$\alpha = \max_{c \in \mathcal{C}} \rho(c)$$

where

$$\rho(c) = \frac{\sum_{(i,j) \in c} L_{ij}}{\sum_{(i,j) \in c} H_{ij}}$$

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

Previous studies of this problem have shown the problem is NP-hard ([Hanen, 1994]) for throughput maximization. One can find the description of two classical methods of performance evaluation of cyclic job shop in [Hanen and Munier, 1995a]. The first one is based on graph theory. In this framework, the throughput computation problem of a cyclic job shop schedule becomes the search of maximum circuit ratio in a graph. The second approach considers the job shop as a $(max, +)$-linear system. Then, the spectrum of the evolution matrix of the system gives the cycle time. In both approaches, the complexity of the evaluation of the cycle time is the same (polynomial).

3 Varying processing times and critical circuit

Tasks are often subject to uncertainties that have a negative impact on the activity duration. In this part, we consider that a subset of tasks are subject to uncertainties but we know the minimum processing time and the maximum processing time. First we show that the critical circuit is a variable element.

Let us consider a task i such that $p_i \in [\underline{p_i}, \overline{p_i}]$ with $p_i = \underline{p_i} + \delta_i$ where $\delta_i \in [0, \overline{p_i} - \underline{p_i}]$ is the variation and C is a circuit that involves the task i. The mean cycle of C is then a function of δ_i and we have

$$\alpha_C(\delta_i) = \frac{\sum_{(i,j)\in C} L_{ij}}{\sum_{(i,j)\in C} H_{ij}} + \frac{\delta_i}{\sum_{(i,j)\in C} H_{ij}}.$$

If the critical circuit does not include the task i, the mean value of the critical circuit is not affected by this variation but if we consider a variation δ_i big enough, the circuit C could become the critical circuit of the graph. The cycle time of the schedule is then defined by $\alpha(p_i)$ where $\alpha(\cdot)$ is piecewise affine since several circuits can become critical when p_i is increasing. An illustration of a function $\alpha(\cdot)$ is drawn in fig. 1. We can distinguish that three circuits can be critical according to p_i.

In order to measure the quality of a schedule S with respect to this variation, we can consider the following integral:

$$V(S) = \int_{\underline{p_i}}^{\overline{p_i}} \alpha(p_i) dp_i.$$

The above reasoning still holds if more than one task are time varying and the integral calculus is computed with the Vinci library [Bueler et al., 2000].

Note that $\frac{1}{\overline{p_i} - \underline{p_i}} \times \int_{\underline{p_i}}^{\overline{p_i}} \alpha(p_i) dp_i$ is the mean value of the cycle time over the variation of p_i. The function $\alpha(\cdot)$ is built from an exact enumeration of circuits that involves the task i. The enumeration of these circuits is difficult and Tarjan's algorithm [Tarjan, 1973] or Johnson's algorithm [Johnson, 1975] are used in this purpose.

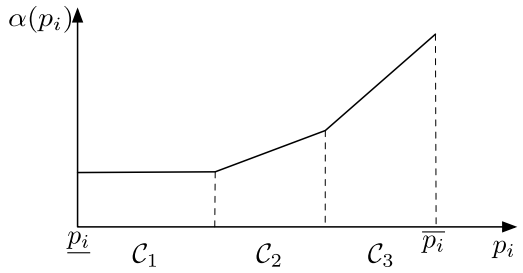

Fig. 1. The cycle time is a function of the variation of the task i.

4 A branch and boud method to compute the best schedule

In this part, we design a method to find the schedule S_{opt} with minimum volume $V(S_{opt})$. We assume that the set of varying duration tasks is known and bounds are available for each of these durations. First we compute all the circuits that involve one or more

varying duration task. It is a necessary step to build $V(S)$ as the cycle time can move from a circuit to another as we discussed in the previous section.

The same branch and bound scheme as in [Fink et al., 2012] can be used for solving this problem. More precisely we start by considering the problem with no disjunction constraints (that is an obvious relaxation of the problem) but only the uniform constraints. It is equivalent in the scheduling problem to consider that each task is performed on a induividual machine. The set of linear constraints builds a polyhedron and we can compute its volume with the Vinci library. Then we separate the search space by fixing one of the disjunctions, i.e. we set value to K_{ij} and K_{ji} for tasks i and j mapped on the same machine. A lower bound of the minimal volume is achieved by computing the volume of the current node. When all the occurrence shifts are fixed a complete schedule is obtained. The best solution of the procedure corresponds to the schedule with lowest volume of the polyhedron generated. As we have noticed in §3, this schedule is also the schedule with the minimal mean value.

5 Conclusion

We use a set of random intances to test the procedure. Until now tests have been processed with a number task from 10 to 15 tasks on 5 machines and 3 jobs. For each instance from 2 to 6 tasks have a varying processing time. All the instances (40) with 10 tasks were solved rapidly (less than 20 seconds) however only 11% of the instances with 15 tasks were solved optimaly in less than 10 minutes. Moreover this procedure has potential for improvement and should be subject to future research to achieve better results. We also plan to consider other criteria such that maximizing the number of scenarios (a scenario is an instantiation of all the varying processing times) such that the cycle time is below a given level or maximizing the range of value δ_i around a point of interest such that the cycle time is unchanged. This last criterion is in accordance with the definition of robustness since it leads to a schedule that is the most insensitive to data uncertainty.

References

[Brucker and Kampmeyer, 2008] Brucker, P. and Kampmeyer, T. (2008). A general model for cyclic machine scheduling problems. *Discrete Applied Mathematics*, 156(13):2561 – 2572.

[Bueler et al., 2000] Bueler, B., Enge, A., and Fukuda, K. (2000). Exact volume computation for polytopes: A practical study. *Polytopes / Combinatorics and Computation*, 29.

[Fink et al., 2012] Fink, M., Rahhou, T. B., and Houssin, L. (2012). A new procedure for the cyclic job shop problem. In *Proceedings of INCOM 2012*, volume 14, Bucharest, Romania.

[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, 1995a] Hanen, C. and Munier, A. (1995a). *Scheduling Theory and Its Applications*, chapter Cyclic Scheduling on Parallel Processors: An Overview. Wiley.

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

[Hillion and Proth, 1989] Hillion, H. P. and Proth, J. M. (1989). Performance evaluation of job-shop systems using timed event graphs. *IEEE Transaction on Automatic Control*, 34(1):3–9.

[Johnson, 1975] Johnson, D. B. (1975). Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing*, 4(1):77–84.

[Kats and Levner, 1997] Kats, V. and Levner, E. (1997). A strongly polynomial algorithm for no-wait cyclic robotic flowshop scheduling. *Operations Research Letters*, 21:171–179.

[Pinedo, 2005] Pinedo, M. (2005). *Planning and Scheduling in Manufacturing and Services*. Springer.

[Tarjan, 1973] Tarjan, R. (1973). Enumeration of the elementary circuits of a directed graph. *SIAM Journal on Computing*, 2(3):211–216.