Predictive-Reactive Scheduling for the Single Machine Problem

Mohamed Ali Aloulou, Marie-Claude Portmann and Antony Vignier
MACSI Team of INRIA-Lorraine and LORIA-INPL, France.
aloulou@loria.fr, portmann@loria.fr, vignier@loria.fr

Abstract

We propose a predictive-reactive approach for the single machine problem. It constructs a set of schedules following a partial order and, every time a decision should be taken, uses the temporal flexibility and the flexibility in job sequencing introduced to provide the decision-maker a set of alternatives, compatible with the modeled constraints.

1. Introduction

Scheduling is an important element of production systems because it allows to improve the performance of the system and serves as an overall plan on which many other shop activities are based. Several techniques have been proposed to generate, for a given problem, a unique schedule satisfying the shop constraints. However, when this pre-computed schedule is released for execution, continual adapting is required to take into account the presence of uncertainties. In the beginning, the schedule is slightly modified and the performance is a little bit affected. But when there are more important perturbations, the performance of the final schedule is generally much worse than the initial one.

To handle such problems, many approaches of scheduling and rescheduling have been proposed in literature to take into account the presence of uncertainties. Metha and Uzsoy [4] classified these approaches into four main categories: completely reactive approaches, predictive-reactive approaches (PRS), robust scheduling and knowledge-based scheduling. They proposed an approach, which tries to ensure that the predictive and realized schedules do not differ drastically in terms of the job completion times. They consider the one machine problem with dynamic job arrival, random breakdowns and total tardiness as objective function. They use an initial sequence of jobs computed by a heuristic called ATC (Apparent Cost Tardiness), and then insert additional idle times into the schedule in order to absorb the impact of possible breakdowns during its implementation. They show that their approach provides high predictability with minor sacrifices in the realized schedule performance.

Inserting idle time between the jobs provides, what can be considered as, temporal flexibility. Indeed, to each job is assigned an interval, greater than its processing time, in which it can be executed without decreasing the predicted performance. Such an approach, called Predictable Scheduling, is very interesting because it prepares the on-line control of the shop in presence of breakdowns. But when other disruptions, related for instance to job arrival or tool availability, are considered, this approach may not be efficient because it provides only one sequence. We can say that it does not give flexibility in job sequencing. An interesting idea, already considered by Billaut and Roubellat [2], is to propose a set of equivalent schedules following a given structure. Every time a decision has to be taken, a set of actions, compatible with the modeled constraints, is given to the decision maker who can choose the convenient action depending on his preferences and eventually non-modeled constraints, for example the presence of set-up times.

In this paper, we consider a single machine problem. We propose a predictive-reactive approach in two steps. The first step builds a set of schedules restricted to follow a partial order. Then, the flexibility in job sequencing, introduced in the first step, is used on-line to propose to the decision-maker a set of alternatives when necessary. We will focus in more details on the predictive approach.

2. A description of the predictive-reactive approach

The aim of this work is to conceive an interactive tool, for solving single machine scheduling problems, based on a predictive-reactive approach. Addressed problems have dynamic job arrivals and total weighted tardiness and makespan as objective functions.

2.1. Principle

A set of flexible schedules is first generated and released for execution. Then, a reactive algorithm is used, when necessary, to adapt the original solutions to the shop floor reality. However, unexpected events, like machine breakdowns, material availability, etc., can make unfeasible the predictive schedule or/and lead to a schedule with poor performance. Therefore, instead of proposing only one schedule, a set of schedules respecting a jobs' partial order is provided. This allows to dispose of some flexibility in job sequencing. The partial order computation takes into account the desired range of performance value given by the decision-maker. It may also be subject to technological constraints, company's preferences and possible disruptions.

Moreover, a temporal flexibility can be provided according to a time window, in which the jobs can be executed, and to the type of the chosen schedules: active, semi-active or non-delay.

In this paper, we consider semi-active schedules, where jobs cannot be shifted to start earlier without changing the job sequence or violating precedence constraints or release dates. Choosing semi-active schedules allows the presence of idle times between the jobs. Thus, it would be possible to execute a job in a time window larger than its processing time, which represents a temporal flexibility.

If the flexibility induced by the partial order and the possible idle times proved to be enough to absorb all unforeseen perturbations, there is no need to reschedule. Otherwise, a reactive algorithm is used to choose a convenient action with minimum changes on the partial order and minimum loss of performance. When the performance is considerably less than the desired performance, then rescheduling is necessary.

2.2. Flexibility and performance definitions

The main characteristics of a set of schedules following a partial order, called a solution, are its flexibility and its performance. The less precedence constraints imposed by the partial order, the more flexible the associated solution. The flexibility in job sequencing is defined by the number of different schedules feasible with respect to the partial order. Due to the combinatorial computation of this number, we suppose that a solution is more flexible than another one if the transitive graph representing the associated partial order contains fewer arcs. Besides, we propose a qualitative scale of five flexibility levels depending on the number of arcs in the transitive graph: zero flexibility (one sequence: $n*(n-1)/2$ arcs), low flexibility, middle flexibility, large flexibility and full flexibility (zero arc). The temporal flexibility of a solution can be defined as the ratio between the time window in which the jobs can be executed and the total processing time of the jobs.

Due to some real production circumstances, it may happen that only schedules with bigger costs, compared to the best schedule, can be chosen by the decision-maker. This implies that, to evaluate the performance of a set of schedules, minimum and maximum values of the scheduling criteria have to be determined. Thus, the performance of a solution is defined by the ranges of the objective function values calculated for all feasible schedules satisfying the given precedence constraints.

Observe that if the jobs are allowed to start arbitrarily late, then the maximum values of the regular objective functions can be arbitrarily large. In this case, they cannot be used to evaluate the performance of any solution. Therefore, the set of feasible schedules must be restricted so that the jobs do not start too late. As mentioned before, we restrict the schedules to be left shifted, i.e. semi-active.

However, it will be difficult to compare between different solutions if we use the proposed performance definition. Thus, we propose a surrogate measure of the performance. The following notations are used. Let:

- $\mathcal{S}$ be the set of flexible solutions (solution = set of schedules following a partial order),
- S be an element of $\mathcal{S}$,
- o_S be a schedule in S.

To each schedule o_S in S is associated a vector $\Gamma(o_S)$, s.t. $\Gamma^t(o_S) = (\Gamma_1(o_S),\Gamma_2(o_S),\Gamma_3(S))$, where $\Gamma_1(o_S)$ is the makespan of o_S, $\Gamma_2(o_S)$ its total weighted tardiness and $\Gamma_3(S)$ is the flexibility in job sequencing of the solution S. Hence, a solution S is represented by the set of vectors $\Gamma(o_S)$. Figure 1 gives the projection of the vectors $\Gamma(o_S)$ on the $(C_{max},\Sigma w_jT_j)$ space.

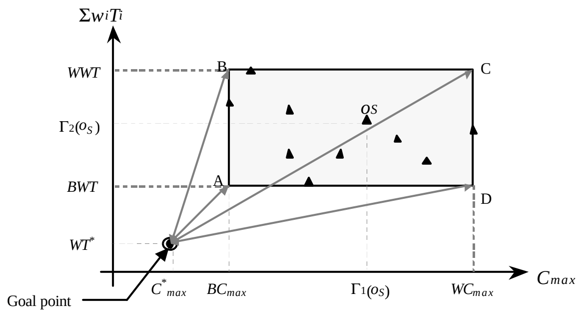

Figure 1. The representation of a solution in $(C_{max},\Sigma w_jT_j)$ space

We denote by *goal point*, the point in $(C_{max},\Sigma w_jT_j)$ space whose coordinates are respectively the best makespan C^*_{max} and the best total weighted tardiness WT^* of the problem (see figure 1).

A surrogate measure of the performance of a solution S can be defined as a linear combination of the distances between the goal point and the points $(\Gamma_1(o_S),\Gamma_2(o_S))$. But the schedules o_S are simply characterized by a partial order (they are not enumerated). So, we may rather consider a linear combination D_S of the distances between the goal point and the four points A, B, C and D (see figure 1). These points are determined by computing the best and worst makespan ant total weighted tardiness. The lower D_S is, the higher the performance of solution S is.

2.3 Best and worst performance computation

The objective functions considered are the total weighted tardiness Σw_jT_j and the makespan C_{max}. It is proved that the problem of minimizing the makespan with respect to precedence constraints, denoted by $1/r_j,prec/C_{max}$, can be solved in $O(n^2)$ times [3]. The problem of minimizing the total weighted tardiness, denoted by $1/r_j,prec/\Sigma w_jT_j$ is NP-hard in the strong sense [3]. Consequently, for this problem we implemented a genetic algorithm based heuristic and made comparison with known dynamic dispatching rules like ATC, X-RM and KZRM [6]. The results showed that genetic algorithms are efficient but too time consuming for a large number of jobs.

For maximization problems, the problem of maximizing C_{max} with respect to precedence constraints and release dates when semi-active schedules are considered, denoted by $1(semi\text{-}active)/r_j,prec/(C_{max}\rightarrow \max)$, can be solved in $O(n^2)$ times [1]. The problem of maximizing total weighted tardiness, denoted by $1(semi\text{-}active)/r_j,prec/(\Sigma w_jT_j\rightarrow \max)$, is NP-hard in the strong sense [1]. We developed several heuristics based on genetic algorithms and dispatching rules. We obtained the same conclusions as for minimization problems.

As a result, for a given solution (a set of schedules w.r.t. a partial order), we use polynomial time algorithms to compute the exact minimal and maximal makespan and approximate algorithms based on dispatching rules for estimated minimal and maximal total weighted tardiness.

3. Predictive schedules construction

The problem is to find the Pareto optimal solution set of the obtained bi-criteria problem: minimize D_S (see paragraph 2.2) and maximize the flexibility subject to restrictions on C_{max} and $\Sigma w_j T_j$ given by the decision-maker.

A Multi-Objective Genetic Algorithm (MOGA) is used to construct flexible predictive schedules. It is based on the use of ternary precedence constraint oriented matrix A = $(a(i,j))$, $1 \leq i,j \leq n$ to code the solutions of the problem (partial orders). It is defined by

- $a(i,j) = 1$ if job i must be performed before job j,
- $a(i,j) = -1$ if job i must be performed after job j,
- $a(i,j) = 0$ for $i=j$ or if there is no precedence constraints between the jobs.

The genetic operators are an adaptation of the crossover MT3 and the mutation MUT3 proposed in [5]. They guarantee the following property: if i precedes j in both parents *1* and *2*, i precedes j in the created offspring.

The decision-maker gives the different restrictions on the objective functions and eventually additional constraints or some preferences. Then, for a fixed flexibility value (in the qualitative scale), the algorithm searches the best solutions minimizing D_S and satisfying the restrictions, if they exist. In order to get the most flexible solutions satisfying the different constraints, the algorithm is applied several times for different flexibility values.

The output of the algorithm is a set of solutions that respect the restrictions and minimize the surrogate measure of performance D_S. The decision-maker is also given the ranges of the objective function values corresponding to these solutions. He can decide which solution will be released for execution.

4. Computational experiments

In order to prove the efficiency of the proposed approach, we will experiment the obtained solutions by means of simulation on different scenarii of shop floor circumstances.

References

[1] Aloulou, M.A., Kovalyov, M.Y. and Portmann, M.C. (2001). Maximization Problems in Single Machine Scheduling. *Technical report A01-R-170, LORIA, Nancy, France*.

[2] Billaut, J.C., Roubellat, F. (1996). A new method for workshop real time scheduling. *International Journal of Production Research*, vol. 34, n°6, 1555-1579.

[3] Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G. and Shmoys, D.B. (1993). Sequencing and scheduling: algorithms and complexity, in Graves, S.C., Rinnooy, Kan A.H.G. and Zipkin, P.H. (Eds.): *Logistics of Production and Inventory, Handbook in Operations Research and Management Science 4*, 445-452.

[4] Metha, S.V. and Uzsoy, R. (1999). Predictable scheduling of a single machine subject to breakdowns. *International Journal of Computer Integrated Manufacturing*, 12, 1, 15-38.

[5] Portmann, M.C. (1996). Genetic Algorithms and Scheduling: A State of the Art and some Propositions. *Workshop on Production Planning and Control, Mons, Belgium,* September 9-11 1996, I-XIV.

[6] Rachamadugu, R.V. and Morton, T.E. (1982). Myopic heuristics for the single machine weighted tardiness problem. *Working Paper 30-82-83, Graduate School of Industrial Administration, Carnegie Mellon University*.