Scheduling Problems of Chemical Experiments

Vassilissa Lehoux-Lebacque, Nadia Brauner, Gerd Finke, Christophe Rapine
G-SCOP Laboratory, 46 av. Félix Viallet, France,
{vassilissa.lebacque, nadia.brauner, gerd.finke, christophe.rapine}@g-scop.inpg.fr

The scheduling problem we describe was proposed by the Institut Français du Pétrole (IFP) and was studied in collaboration with the two laboratories Gilco-INPG and Leibniz-IMAG in Grenoble. We developed a software to solve efficiently the industrial problem and derived several interesting theoretical problems that are presented in this paper.

1 Description of the problem

We have to schedule chemical experiments with resource constraints. The aim of the experiments is to find the best conditions to perform the synthesis of certain chemicals. Each task is the set of all the experiments that will be conducted for a given chemical. This set consists of two series of experiments: stage 1 and stage 2. The aim of the first stage is to find the best heating duration for the chemical. We start with a given heating time. If it was too short, the experiment is started again with an increased heating duration, and if it was too long, with a decreased duration. Once the perfect heating time is found, we can start the other stage. There, we modify slightly the conditions of the experiments with the heating time obtained in stage 1 as to improve the quality of the chemical obtained. For this stage, the experiments are independent and can be performed in parallel.

To conduct these experiments, we use a set of cyclic containers in which we can perform several experiments at the same time, provided that they have exactly the same duration. Each experiment lasts between 3 and 21 days, and the total number of experiments is several thousands.

The purpose of the software we developed was both maximizing the filling rate of the containers in order to finish as early as possible the set of all the tasks and terminating evenly the tasks as to regularly obtain results on the quality of the chemicals.

The operator is needed at the beginning and at the end of each heating operation, but not during its processing. The planning must take into account the days-off, weekends, etc.

From this industrial problem, we developed several research directions that are presented in the following sections.

2 Considering periods where the tasks can neither start nor end

In the classical scheduling literature, the notion of machine unavailability periods is well known (for maintenance, for example). In our case, we have special periods (weekends, holidays, etc.) where the chemists are not available. However, the presence of the chemists is required to handle the starting and termination of the experiments. Therefore, the subsequent type of problems occurs: we have parallel machines, with a given set of operator-non-availability periods $S(I)$ described as open intervals. The solutions of these problems are schedules with no operation starting nor terminating in any interval of $S(I)$.

The problem of minimizing the makespan is NP-hard and this is already true on one machine with only one interval and independent operations. The problem may be formulated as follows. Given are n operations with processing times $p_1, p_2, \dots, p_n$ and a single operator-non-availability period (o-na-period) of duration Δ located at $[C, C+\Delta]$. We may distinguish between three classes of these problems:

(1) all processing times are smaller than Δ: $p_i < \Delta \quad \forall i$;

(2) there are processing times smaller than Δ and others greater than Δ;

(3) all processing times are greater than Δ: $p_i \geq \Delta \quad \forall i$.

In particular, case (3) applies to our applications. The length of the experiments are given in days (≥ 3) and the o-na-periods are usually 2 days (weekends). In this case, we prove that on one machine and one interval, makespan minimization can be done in $O(n \log n)$ where n is the number of experiments.

3 Scheduling chains of operations on a batching machine with task compatibility

At an intermediate step of stage 1, we are confronted with the scheduling of chains of operations which come from the different chemicals. Therefore in the classical notation of scheduling, we have tasks where the precedence graph is a chain. In addition, our machines are so-called *batch machines*, *i.e.*, machines that can handle simultaneously several operations [1]. We call two operations *batch compatible* if they have exactly the same length which is then also the length of the batch. In the scheduling context, we have problems of scheduling on batch machines with precedence constraints and batch compatibility. The objective is to minimize the makespan, *i.e.*, to finish as quickly as possible all operations.

The following example is composed of two tasks with respective sequences of operations

$$1 \to 21 \to 5 \to 14 \quad \text{and} \quad 1 \to 5 \to 14 \to 21$$

A possible schedule of length 61 is:

1	21	5	14	21
1		5	14	

where the operations of length 1, 5 and 14 are grouped in batches of capacity 2 and the operations of length 21 are scheduled alone. An optimal schedule of length 60 is:

1	5	14	21	5	14
1			21		

This problem contains as a subcase the shortest subsequence problem which is already NP-hard. Interesting new problems and results where derived from this study with a connection to sequence alignment in bio-informatics.

4 Minimizing the total completion time of the chemicals

At the final stage of the experiments, one has to schedule tasks that are all in stage 2. So for each task, we have a set of operations to finish as fast as possible (this would be in fact, the finishing of one chemical). To state on the interest of industrial production for a given chemical, the chemists need the results of all its experiments. We are therefore focussing on the end of the last experiment of each chemical and not that of each of its operation. Since we want to finish stage 2 for all given chemicals, the problem can be stated as follows: a task T_i is a set of operations and C'_i is the completion time of task T_i, *i.e.* the end its last operation. All tasks are independent and can be processed in parallel. The objective is to minimize $\sum C_i'$. Observe that this problem is closely related to [3]. Again we have to schedule such tasks on a system of parallel machines. Note that this problem is different from the classical scheduling problem $Pm||\sum C_i$ where C_i is the termination of each single operation without the set structure explained above. In the classical case, SPT gives an optimal solution in polynomial time [4]. The problem is not either directly related to $Pm||\sum w_iC_i$ since you cannot know which will be the last operation of a task.

Returning to our case, the problem is easy if restricted to a single machine: SPT applied to the sum of the durations of operations for the tasks is optimal. Therefore, $1||\sum C_i'$ is polynomially solvable.

For two-machines ($P2||\sum C_i'$), the problem is already NP-hard: it contains the minimization of the makespan ($P2||Cmax$) as a subcase, which is known to be NP-hard [2] (consider a single task).

References

[1] M. Boudhar and G. Finke (2000), Scheduling on a Batch Machine with Job Compatibilities., *Belgian Journal of Operations Research, Statistics and Computer Science - JORBEL* **40**, 69 – 80.

[2] M.R. Garey and D.S.Johnson (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco, CA.

[3] J.Y-T. Leung, H. Li and Pinedo (2005), Order scheduling models: an overview. In *First Multi-disciplinary International Conference on Scheduling: Theory and Applications (MISTA 2003)*, volume 1, 37 – 53, Nottingham, UK, 2003.

[4] M. Pinedo (1995), *Scheduling: Theory, Algorithms, and Systems*, Prentice Hall, Englewood Cliffs, NJ.