CHEMICAL ENGINEERING TRANSACTIONS Volume 25, 2011
Editors Jiří Jaromír Klemeš, Petar Sabev Varbanov, Hon Loong Lam
Copyright © 2011, AIDIC Servizi S.r.l., ISBN 978-88-95608-16-7 ISSN 1974-9791
DOI: 10.3303/CET1125083

Combinatorial Approach to Address Batch Scheduling Problems with Limited Storage Time

Mate Hegyhati, Tibor Holczinger, Andras Szoldatics, Ferenc Friedler*

Department of Computer Science and Systems Technology, University of Pannonia
Egyetem u. 10, Veszprem, H-8200, Hungary
friedler@dcs.uni-pannon.hu

Large number of papers has been published in the field of industrial batch process scheduling providing various approaches to support operation level decision making. Detailed reviews of the available methods are given by Floudas and Lin (2004), Mendez et. al (2006), and Hegyhati and Friedler (2010). One of the key parameters for batch scheduling problems is the storage policy of the process, that depends both on the physical and chemical properties of the process materials, and the storage capacity of the given plant. While the latter defines capacity constraints (unlimited-, finite- or non-intermediate storage), the former limits the storage time of the material (unlimited-, limited- or zero-wait).
Most of the published approaches tackle zero-wait or limited-wait policies by adding supplementary constraints to an existing MILP formulation (Shaik and Floudas, 2009). In the present work, the formerly introduced S-graph framework is extended with a combinatorial approach to address zero- and limited-wait policies.

1. Introduction

The S-graph framework was originally introduced by Sanmarti et. al. (2002) providing a combinatorial tool to solve makespan minimization problems under non-intermediate storage and unlimited wait (UW) policy. This framework consists of a directed graph model, called S-graph, and the corresponding branch-and-bound algorithm. In the S-graph, the vertices denote tasks and products, while each arc represents a timing precedence between the connected nodes. The arcs expressing the precedence of tasks in the recipe are called recipe-arcs. The weight of each recipe-arc is the processing time corresponding to its starting vertex. The sequencing of the tasks assigned to the same unit is expressed by the so-called schedule-arcs. The recipe-arcs are given as an input together with the vertices. The optimization algorithm during the procedure extends this graph with the schedule-arcs. An example schedule represented by an S-graph is given in Figure 1 together with the corresponding Gantt chart. The schedule-arcs are represented by arrows with dashed lines. The problem involves the production of two products in units *j1,j2,* and *j3*. In this particular schedule, unit *j1* starts performing task *i1* then it waits for unit *j3* to take the intermediate material and start performing task *i2*. Then, unit *j1* becomes available to perform task *i5*. The makespan is the longest path in the graph, it is 10 hours for this example.

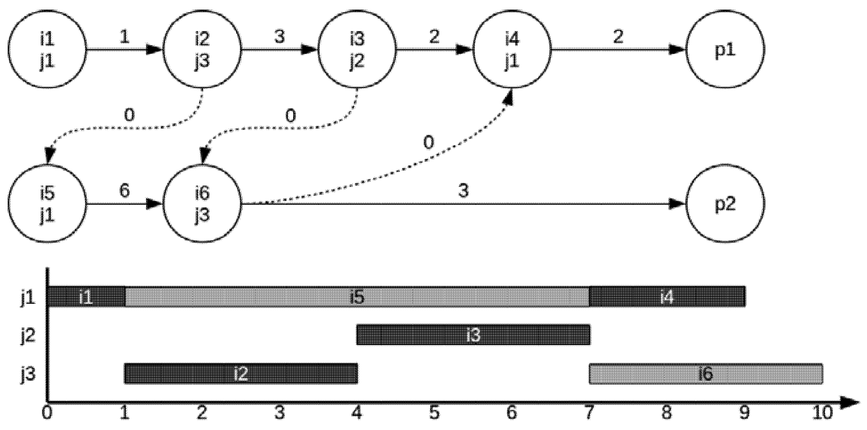

Figure 1: Schedule for two products represented by an S-graph and Gantt chart

Unlimited-wait policy is considered as the default policy for each material in this paper if it is not indicated otherwise. In batch process industries, materials may lose some of their physical or chemical properties necessary for the upcoming tasks if they are stored for too long. For this type of materials limited-wait (LW) policy is considered. If a material requires immediate processing, the policy is termed as zero-wait (ZW). A combinatorial algorithm is introduced first to tackle problems with ZW and UW policy. Then the approach is extended to LW policy, as well.

2. Combinatorial tool to solve batch scheduling problems with ZW operational policy

Materials with ZW storage requirements are represented in the S-graph framework by a new type of recipe-arc, as it is shown for the tasks of product *p1* in Figure 2. An additional number is given in braces to denote the maximum waiting time (in this case zero) for each material where ZW policy is considered.

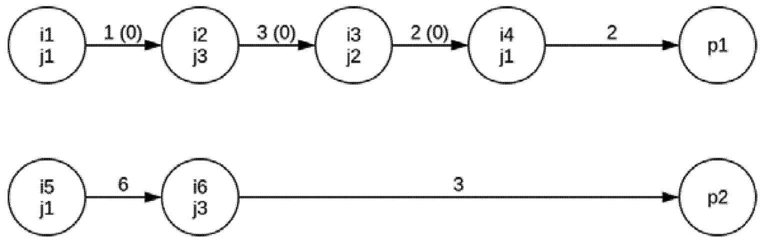

Figure 2: New type of recipe arc representing zero-wait materials

Note that the difference of the starting times of two tasks are fixed by the recipe if the two tasks are connected through materials with ZW storage limitations. Such tasks will be regarded later as ZW-connected tasks. For example, *i2* and *i4* are ZW-connected, and task *i4* must start exactly 3+2=5 hours after the start of task *i2* (see Figure 2).

For a scheduling problem, its UW-relaxation is the scheduling problem, where the restrictions on the storage time of each material is disregarded resulting in UW policy. Obviously, the set of the solutions of a scheduling problem is the subset of the solutions of its UW-relaxation.
A feasible schedule of an UW-relaxation is infeasible for the original problem if the fixed difference between the starting times of two ZW-connected tasks is violated by the schedule. The schedule given in Figure 1 is infeasible for the problem given in Figure 2, even though it is feasible for its UW-relaxation. The timing difference between the starting times of tasks *i2* and *i4* must be 5 hours by the recipe; however, it can not be shorter than 6 hours (see Figure 3).

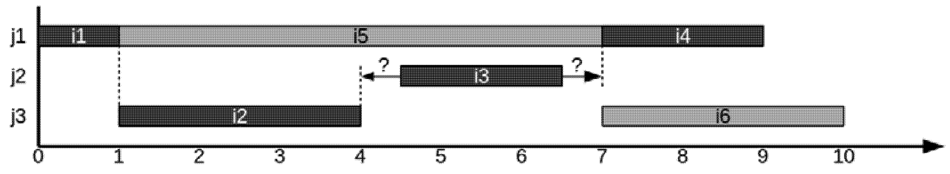

Figure 3: Example for infeasible ZW schedule, demonstrated on a Gantt-chart.

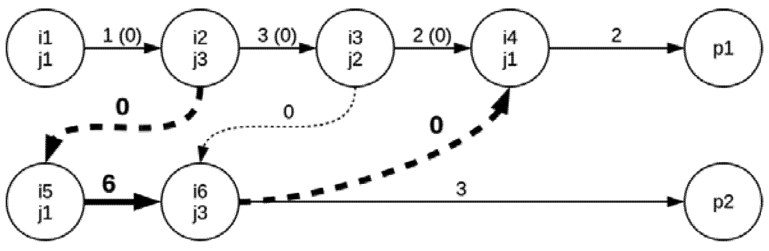

Figure 4: Example for infeasible ZW schedule, demonstrated on an S-graph

A partial schedule of the UW-relaxed subproblem is infeasible if it contains a cycle (Sanmarti et. al., 2002). A feasible partial schedule of the UW relaxation is infeasible for the original (unrelaxed) problem if there are two ZW-connected tasks in the corresponding S-graph, so that the longest path between them is longer than their fixed timing difference. In Figure 4, the schedule of Figure 3 is represented on an S-graph, where the longest path between ZW-connected tasks *i2* and *i4* is 6 hours.
The algorithm to find the optimal ZW schedule is based on the original branch-and-bound algorithm, which generates the schedules for the UW-relaxation. To prune partial schedules that are infeasible for the original problem, the algorithm is to be extended as follows:

- Feasibility test for the longest paths between ZW-connected tasks
- Insertion of additional schedule-arcs after a scheduling decision

Both of these modifications can be implemented effectively in a recent implementation (Smidla and Heckl, 2010). In the original branch-and-bound algorithm, a single schedule-arc is inserted when a decision is made about the sequencing of two tasks assigned to the same unit. In Figure 5, the bold arcs denote some of the additional arcs that are inserted after the new schedule-arc introduced between tasks *i10* and *i6*. For example, *i7* must start at least 7 hours later than *i6* and as a consequence at least 7+0 hours later than *i10*. Since *i10* must start exactly 2 h earlier than *i11*, *i7* must start at least 7+0-2 hours later than *i11*. The weight of the rest of the bold arcs can be evaluated applying the same logic. These additional arcs not only express the timing constraints by ZW stages, but improve the tightness of the bounding function as well.

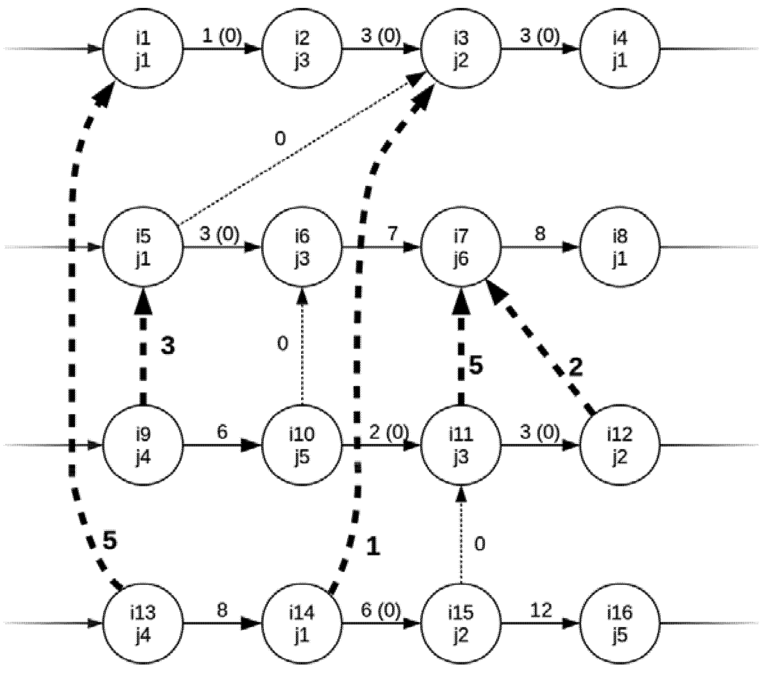

Figure 5: Illustration of the proposed algorithm (additional schedule arcs with bold lines are introduced by the algorithm because of the ZW policy)

3. Limited – Wait storage policy

LW storage policy is represented similarly to ZW policy, the number in the braces denotes the maximum waiting time for the material. In Figure 6, the output of task *i3* can be stored for at most 1 h after it is produced in 2 h.

This type of problems can be transformed to an equivalent ZW problem, to be solved by the previously described algorithm. The transformation consists of adding two auxiliary vertices to each LW stage as it is shown in Figure 7.

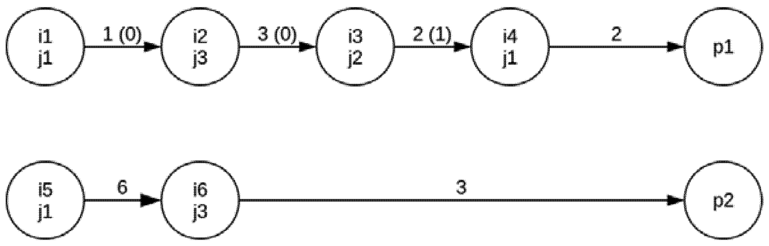

Figure 6: Example with limited-wait stage after i3

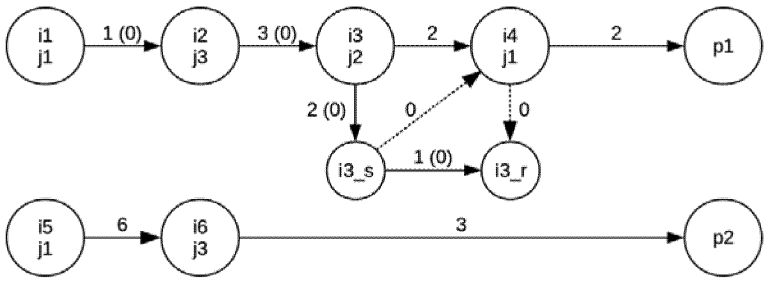

Figure 7: ZW equivalent of the scheduling problem given in Figure 6.

4. Case study

The described method has been tested on a case-study published by Ferrer-Nadal et. al. (2008). The problem involves the production of four products (*A-D*) in four units (*U1-U4)*. The recipe of each product consists of three consecutive tasks (see Figure 8 with processing times). Each intermediate material is considered to have ZW storage requirements.

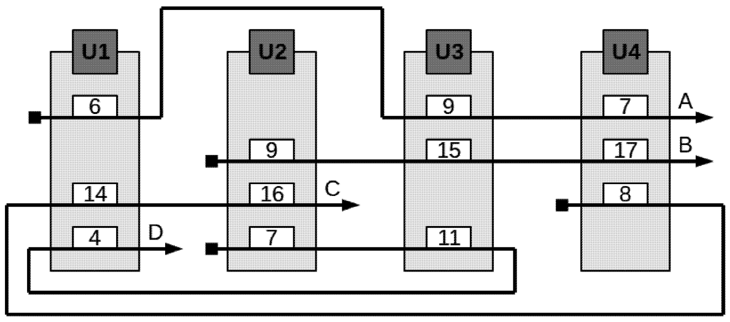

Figure 8: Recipe of the case-study

The problem has been solved for various batch numbers, the result is given in Table 1.

Table 1: Makespan for the case study with different batch numbers

Number of batches				Optimal makespan (h)
A	B	C	D	
1	1	1	1	58
2	1	1	1	62
2	2	1	1	79
2	2	2	1	92
2	2	2	2	92
3	2	2	2	101
3	3	2	2	118
3	3	3	2	133
3	3	3	3	140

5. Concluding remarks

The present work extends the formerly developed S-graph framework to address batch process scheduling problems with limited storage time. An industrial case-study illustrates the efficacy of the new approach. Among other available methods, it is the first combinatorial algorithm for this class of scheduling problems.

References

Ferrer-Nadal S., Capon-Garcia E., Méndez C. and Puigjaner L., 2008, Material transfer operations in batch scheduling. A critical modeling issue, Industrial & Engineering Chemistry Research, 47, 7721-7732

Floudas C. A. and Lin X., 2004, Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review, Computers & Chemical Engineering, 28, 2109-2129.

Hegyhati M. and Friedler F., 2010, Overview of Industrial Batch Process Scheduling, Chemical Engineering Transactions, 21, 895-900

Mendez C. A., Cerda J., Grossmann I. E., Harjunkoski I. and Fahl M., 2006, State-of-the-art review of optimization methods for short-term scheduling of batch processes, Computers & Chemical Engineering, 30, 913-946.

Sanmartí E., Holczinger T., Puigjaner L. and Friedler F., 2002, Combinatorial framework for effective scheduling of multipurpose batch plants, AIChE Journal 48(11), 2557-2570.

Shaik M. A. and Floudas C. A., 2009, Novel unified modeling approach for short-term scheduling, Industrial & Engineering Chemistry Research, 48(6), 2947-2964

Smidla J. and Heckl I., 2010, S-graph based parallel algorithm to the scheduling of multipurpose batch plants, Chemical Engineering Transactions, 21, 937-942