AN IMPROVED CONTINUOUS-TIME FORMULATION FOR SHORT-TERM SCHEDULING WITH RESOURCE CONSTRAINTS

Stacy L. Janak, Xiaoxia Lin and Christodoulos A. Floudas*
Princeton University
Princeton, NJ 08544

Abstract

An enhanced continuous-time formulation is presented for the short-term scheduling of multipurpose batch plants with intermediate due dates and a variety of resource constraints. The proposed formulation is based on the continuous-time formulation presented by Ierapetritou and Floudas (1998) and incorporates several features that include: (i) various storage policies (UIS, FIS, NIS, ZW), (ii) resource constraints, (iii) batch mixing and splitting, and (iv) sequence-dependent changeover times. The key features of the proposed formulation include a continuous-time representation utilizing a necessary number of event points of unknown location corresponding to the activation of a task. Also, tasks are allowed to continue over several event points enabling resource quantities to be correctly determined at the beginning of each resource utilization. An example problem is presented to demonstrate the effectiveness of the proposed approach. The computational results are compared with those in the literature and it is shown that the proposed formulation is significantly faster than other general resource constrained models, especially for problems requiring many event points.

Keywords

Process scheduling, Resource constraints, Utilities, Mixed storage policy, Continuous-time formulation.

Introduction

The problem of short-term scheduling for multiproduct/multipurpose batch plants has received a considerable amount of attention during the last two decades. Extensive reviews were written by Reklaitis (1992) and Pantelides (1993) and more recently by Floudas and Lin (2003a, 2003b). Most of the proposed approaches for short-term scheduling can be classified into two main groups based on time representation: discrete-time models and continuous-time models. Continuous-time approaches can be classified into two categories based on the type of process considered: sequential processes and general network-represented processes. The major difference between these two types of processes is that sequential processes are order or batch oriented and do not require the explicit consideration of mass balances. General network-represented processes correspond to the more general case in which batches can merge and/or split and material balances must be taken into account explicitly. For general network-represented processes, two types of approaches have been developed to formulate continuous-time scheduling models: global event based models and unit-specific event based models. Global event based models use a set of events or time slots that are common for all tasks and all units while unit-specific event based models define events on a unit basis, allowing tasks corresponding to the same event point but in different units to take place at different times. The latter is considered the most general and most rigorous representation of time used in short-term scheduling models. Unit-specific event based models have been developed by Ierapetritou and Floudas (1998a, 1998b), Ierapetritou et al. (1999), Lin and Floudas (2001, 2003),

* To whom all correspondence should be addressed

Lin et al. (2002), and Lin et al. (2003). They proposed a novel continuous-time formulation for short-term scheduling of batch, semi-continuous, and continuous processes. This formulation introduces the original concept of event points, which are a sequence of time instances located along the time axis of unit, each representing the beginning of a task or the utilization of a unit. Because of this definition of an event point, for the same scheduling problem, the number of event points required in the unit-specific event based formulation is smaller than that of a global event based model. This results in a substantial reduction of the number of binary variables.

In this work, we propose an enhanced state-task network MILP model for the short-term scheduling of multiproduct/multipurpose batch plants. The proposed approach extends the work of Ierapetritou and Floudas (1998) and Lin and Floudas (2001) to account for resource constraints, various storage policies (UIS, FIS, NIS, ZW), variable batch sizes and processing times, batch mixing and splitting, and sequence-dependent changeover times.

Problem Statement

The short-term scheduling problem of multipurpose batch chemical processes is defined as follows. Given (i) the production recipe (i.e., the processing times for each task at the suitable units and the amount of the materials required for the production of each product), (ii) the available equipment and their capacity limits, (iii) the material storage policy, (iv) the required utilities and their availabilities, (v) the initial raw materials and orders of final products (amounts and time), and (vi) the time horizon under consideration, then the objective is to determine (i) the optimal sequence of tasks taking place in each unit, (ii) the amount of material being processed at each time in each unit, and (iii) the processing time of each task in each unit, so as to optimize a performance criterion, for example, to minimize the makespan or to maximize the overall profit.

Mathematical Formulation

The proposed formulation is based on the continuous-time formulation presented by Ierapetritou and Floudas (1998) for the short-term scheduling of multipurpose batch plants and incorporates several added features including resource constraints, intermediate due dates, various storage policies (UIS, FIS, NIS, ZW), batch mixing and splitting, and sequence-dependent changeover times. The mathematical model for the proposed approach cannot be included in this short paper due to space limitations, but it can be found in Janak et al. (2004).

In order to incorporate resource constraints, tasks must be allowed to continue over several event points so that the total amount of a resource utilized by all suitable tasks can be accurately monitored at all times. For instance, consider two tasks (*i*) and (*ii*) that are suitable in different units and utilize the same resource. If task (*ii*) begins processing while task (*i*) is in progress, then the amount of resource required at the specific time point when task (*ii*) begins can only be accurately determined if the amount of resource required by both tasks is considered. Therefore, task (*i*) must be allowed to extend over at least two event points so that it can be active at an event point before task (*ii*) starts and also at the specific event point that task (*ii*) begins processing. Thus, the formulation must be organized so that tasks which use the same resource at the same time are active at the same event point and the timing of that event point in the respective units must coincide. In this way, all resource limitations can be accurately monitored by simply checking the amount of each resource utilized at the beginning of each event point.

One of the key consequences of this new formulation is that each processing task must have two sets of binary variables and one set of continuous variables associated with it. The binary variables indicate the beginning and ending of a task at an event point, respectively, while the continuous variables simply indicate if a task is active at an event point, regardless of whether the task is starting, finishing, or just processing. Also, tasks which represent the utilization of a resource or the storage of materials must be introduced so that timing between processing tasks which utilize a specific resource can be enforced and different material storage policies can be imposed.

The mathematical model for the proposed formulation employs many indices, sets, parameters, and binary and continuous variables. The model includes allocation constraints, capacity constraints for processing as well as storage tasks, batch-size matching constraints for processing and resource tasks, material balances, duration constraints for processing, storage, and resource tasks, sequencing constraints for processing, storage, and resource tasks, demand and due date constraints, and an objective function. Most of the constraints are modified from the original model of Ierapetritou and Floudas (1998) in order to allow tasks to occur over several event points. The allocation constraints relate the binary and continuous variables for a specific task so that all tasks which are initiated are completed, tasks can only finish if they have been started, and tasks can only start if all previous tasks have finished. Batch-size constraints are written so that the batch-sizes of tasks extending over several event points are consistent. Duration constraints are altered so that finishing times of tasks that extend over several event points are correctly related to their starting times. Demand and due date constraints are introduced to relate the delivery of materials to the amount required and the time due. The different objective functions considered include the maximization of sales, the minimization of makespan, and the minimization of due date tardiness. Note that additional constraints were introduced throughout the model to incorporate resource and storage tasks and associate them with related processing tasks in order to manage resource levels and enforce material storage policies.

Example Problem

The example problem comes from Maravelias and Grossmann (2003) and involves the state-task network given in Figure 1. This example includes resource constraints, mixed storage policies, and variable batch sizes, processing times, and utility requirements. The plant consists of six units involving ten processing tasks and fourteen states. Unlimited intermediate storage (UIS) is available for raw materials F1 and F2, intermediates I1 and I2, and final products P1-P3 and WS. Finite intermediate storage (FIS) is available for states S3 and S4 while no intermediate storage (NIS) is available for states S2 and S6 and a zero-wait policy (ZW) applies for states S1 and S5. There are three different renewable utilities: cooling water (CW), low-pressure steam (LPS), and high-pressure steam (HPS). Tasks T2, T7, T9, and T10 require CW; tasks T1, T3, T5, and T8 require LPS; and tasks T4 and T6 require HPS. The maximum availabilities of CW, LPS, and HPS are 25, 40, and 20 kg/min, respectively. The objective function is the maximization of sales and the time horizon of interest is 12 hours. The problem data is available in Maravelias and Grossmann (2003).

The example is implemented with GAMS 2.50 and is solved using CPLEX 8.1 with a 3.00 GHz Linux workstation. The default GAMS/CPLEX options are used with the exception that the CPLEX option for feasibility is activated and a relative optimality tolerance equal to 0.01% is used as the termination criterion.

With a time horizon of 12 hours, the optimal sales are $13,000 and eight event points are required. The production schedule and resource utilization levels can be seen in Figure 2. The problem involves 3318 constraints, 110 binary, and 1077 continuous variables and the optimal solution was found in 222 nodes and 1.71 s.

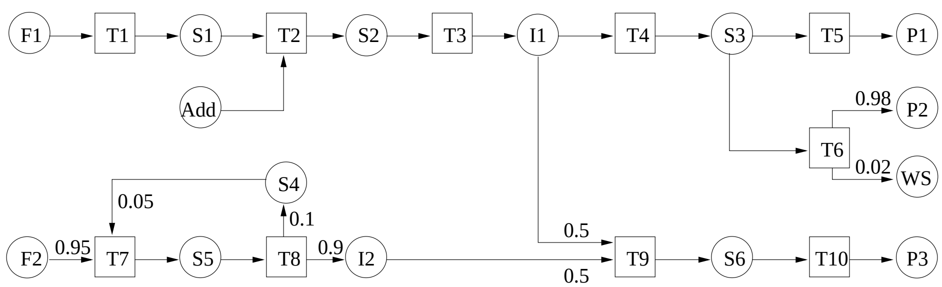

Figure 1. State-Task Network for Example.

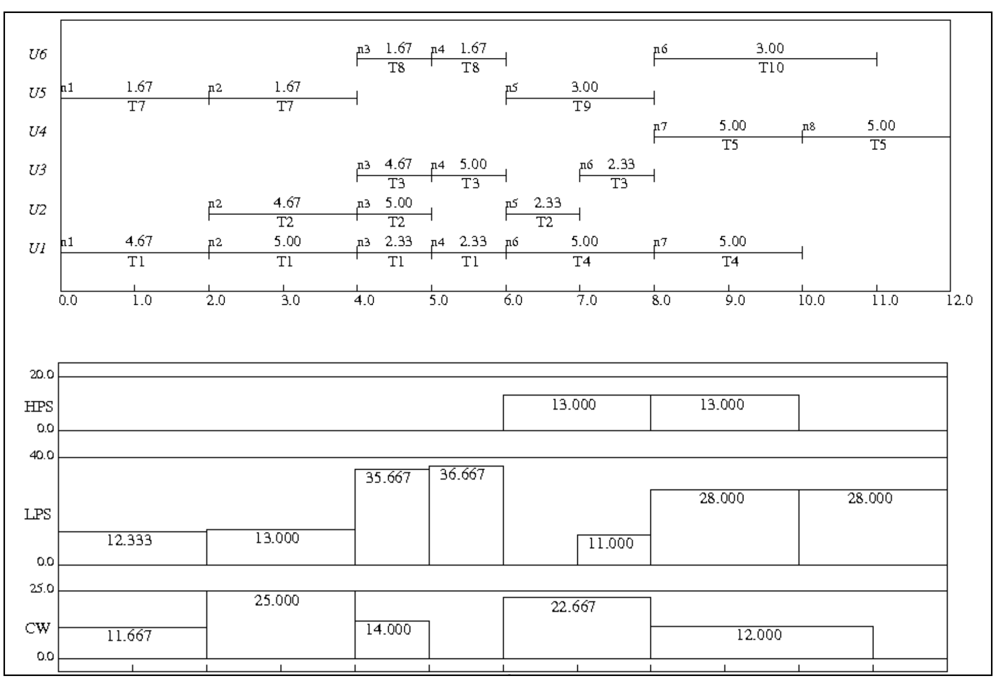

Figure 2. Schedule for Example with Time Horizon of 12 h.

The example was also solved using an implementation of model M* proposed in Maravelias and Grossmann (2003). Although this example was solved in their original paper, we have re-solved it here in order to compare the models using the same computational tools.

The model and solution statistics for both formulations can be seen in Table 2. For a time horizon of 12 hours using nine time points, the model M* involved 2396 constraints, 180 binary, and 1408 continuous variables. The same optimal solution of $13,000 was found in 23235 nodes and 64.92 s. Note that the model M* of Maravelias and Grossmann (2003) takes one more time point and thus involves more binary and continuous variables. Also, the model M* takes over 100 times more nodes to solve than the proposed approach. Note that in the full-length paper, Janak et al. (2004), several example problems are presented and compared with results in the literature to demonstrate the effectiveness of the proposed approach.

Table 2. Model and Solution Statistics for Example.

	Proposed Formulation	Maravelias and Grossmann (2003) Formulation[1]
Horizon (h)	12	12
Event Points	8	9
Bin. Var.	110	180
Cont.Var.	1077	1408
Constraints	3318	2396
Objective ($)	13000	13000
Nodes	222	23235 (2107)
CPU Time (s)	1.71	64.92

Conclusions

In this paper, an enhanced continuous-time formulation is presented for the short-term scheduling of multipurpose batch plants with intermediate due dates. The proposed formulation incorporates several features including various storage policies (UIS, FIS, NIS, ZW), resource constraints, variable batch sizes and processing times, batch mixing and splitting, and sequence-dependent changeover times. The key features of the proposed formulation include a continuous-time representation utilizing a necessary number of event points of unknown location corresponding to the activation of a task. Also, tasks are allowed to continue over several event points enabling resource quantities to be correctly determined at the beginning of each resource utilization. An example problem is presented to illustrate the effectiveness of the proposed formulation. The computational results are compared with those in the literature and it is shown that the proposed formulation is significantly faster than other general resource constrained models.

[1] Numbers in parenthesis represent values reported by Maravelias and Grossmann (2003).

Acknowledgments

The authors gratefully acknowledge support from the National Science Foundation.

References

Floudas, C. A., Lin, X. (2003a). Continuous-Time versus Discrete-Time Approaches for Scheduling of Chemical Processes: A Review. *Comput. Chem. Eng.*, submitted for publication.

Floudas, C. A., Lin, X. (2003b). Mixed Integer Linear Programming in Process Scheduling: Modeling, Algorithms, and Applications. *Ann. Oper. Res.*, submitted for publication.

Ierapetritou, M. G., Floudas, C. A. (1998a). Effective Continuous-Time Formulation for Short-Term Scheduling: 1. Multipurpose Batch Processes. *Ind. Eng. Chem. Res.*, 37, 4341.

Ierapetritou, M. G., Floudas, C. A. (1998b). Effective Continuous-Time Formulation for Short-Term Scheduling: 2. Continuous and Semi-continuous Processes. *Ind. Eng. Chem. Res.*, 37, 4360.

Ierapetritou, M. G., Hene, T. S., Floudas, C. A. (1999). Effective Continuous-Time Formulation for Short-Term Scheduling: 3. Multiple Intermediate Due Dates. *Ind. Eng. Chem. Res*., *38*, 3446.

Janak, S. L., Lin, X., Floudas, C. A. (2004). An Enhanced Continuous-Time Unit-Specific Event Based Formulation for Short-Term Scheduling of Multipurpose Batch Processes: Resource Constraints and Mixed Storage Policies. *Ind. Eng. Chem. Res.*, accepted for publication.

Lin, X., Floudas, C. A. (2001). Design, Synthesis and Scheduling of Multipurpose Batch Plants via an Effective Continuous-Time Formulation. *Comput. Chem. Eng.*, *25*, 665.

Lin, X., Floudas, C. A., Modi, S., Juhasz, N. M. (2002). Continuous-Time Optimization Approach for Medium-Range Production Scheduling of a Multiproduct Batch Plant. *Ind. Eng. Chem. Res.*, *41*, 3884.

Lin, X., Floudas, C. A. (2003). A Novel Continuous-Time Modeling and Optimization Framework for Well Platform Planning Problems. *Optimization Eng.*, *4*, 65.

Lin, X., Chajakis, E. D., Floudas, C. A. (2003). Scheduling of Tanker Lightering via a Novel Continuous-Time Optimization Framework. *Ind. Eng. Chem. Res.*, *42*, 4441.

Maravelias, C. T., Grossmann, I. E. (2003). New General Continuous-Time State-Task Network Formulation for Short-Term Scheduling of Multipurpose Batch Plants. *Ind. Eng. Chem. Res*., *42*, 3056.

Pantelides, C. C. (1993). Unified Frameworks for Optimal Process Planning and Scheduling. *In Proceedings of* *the Second International Conference on Foundations of Computer-Aided Process Operations*, 253.

Reklaitis, G. V. (1992). Overview of Scheduling and Planning of Batch Process Operations. *Presented at NATO Advanced Study Institute – Batch Process Systems Engineering*, Antalya, Turkey.