Continuous-Time Models for Short-Term Scheduling of Multipurpose Batch Plants: A Comparative Study

Munawar A. Shaik, Stacy L. Janak, and Christodoulos A. Floudas*

Department of Chemical Engineering
Princeton University
Princeton, New Jersey 08544-5263

Abstract: During the last two decades, the problem of short-term scheduling of multiproduct and multipurpose batch plants has gained increasing attention in the academic, research, and manufacturing communities, predominantly because of the challenges and the high economic incentives. In the last 10 years, numerous formulations have been proposed in the literature based on continuous representations of time. The continuous-time formulations have proliferated because of their established advantages over discrete-time representations and in the quest to reduce the integrality gap and the resulting computational complexities. The various continuous-time models can be broadly classified into three distinct categories: slot-based, global event-based, and unit-specific event-based formulations. In this paper, we compare and evaluate the performance of six such models, based on our implementations using several benchmark example problems from the literature. Two different objective functions, maximization of profit and minimization of makespan, are considered, and the models are assessed with respect to different metrics such as the problem size (in terms of the number of binary variables, continuous variables, and constraints), computational times (on the same computer), and number of nodes needed to reach zero integrality gap. Two additional computational studies with resource constraints such as utility requirements are also considered.

1. Introduction

The problem of short-term scheduling of multiproduct and multipurpose batch plants has received significant attention from both academic and industrial researchers in the past few years, primarily

* Corresponding author. Tel.: +1-609-258-4595. Fax: +1-609-258-0211. E-mail: floudas@titan.princeton.edu

because of the challenges and the high economic tradeoffs involved. Recently, Floudas and Lin[1, 2] presented state-of-the-art reviews comparing various discrete and continuous time-based formulations. The different continuous-time models proposed in the literature can be broadly classified into three distinct categories: slot-based, global event-based, and unit-specific event-based formulations.

One of the first methods used to formulate continuous-time models for the scheduling of network-represented or sequential processes is based on the concept of time slots. Time slots represent the time horizon in terms of ordered blocks of unknown, variable lengths, or slots, as presented by Pinto and Grossmann,[3-6] Karimi and McDonald,[7] Lamba and Karimi,[8, 9] and recently by Sundaramoorthy and Karimi.[10] In addition, alternate methods have been developed which define continuous variables directly to represent the timings of tasks without the use of time slots. These methods, for both network-represented and sequential processes, can be classified into two different representations of time, global event-based models and unit-specific event-based models. Global event-based models use a set of events that are common across all units, and the event points are defined for either the beginning or end (or both) of each task in each unit. Research contributions following this direction include those presented by Zhang and Sargent,[11, 12] Mockus and Reklaitis,[13-15] Schilling and Pantelides,[16, 17] Mendez and co-workers,[18-20] Castro and co-workers.,[21,22] Mendez and Cerda,[23] Majozi and Zhu,[24] Lee et al.,[25] Burkard et al.,[26] Wang and Guignard,[27] and Maravelias and Grossmann.[28] Note that the concept of time slots has changed over a period of time. Earlier models were based on the assignment of tasks to time slots, while in the recent models, continuous variables are used to directly assign tasks to different time points; and hence, these are more similar to global event-based models. The only difference seems to be that the length of the common-time grid is referred to as the time slot explicitly in some slot-based formulations (for instance by Sundaramoorthy and Karimi[10]).

On the other hand, unit-specific event-based models, originally developed by Floudas and co-workers,[29-36] define events on a unit basis, allowing tasks corresponding to the same event point but in different units to take place at different times. This representation is considered the most general, compact, and "true" continuous-time model (as is demonstrated later in this paper) used in short-term scheduling. Another unit-specific event-based continuous-time model was developed by Giannelos and Georgiadis.[37] However, because of special sequencing restrictions of the same start and finish times on tasks consuming or producing the same state, it is effectively transformed into a global event-based model. Their formulation is similar to that by Ierapetritou and Floudas;[29]

however, because of the special sequencing constraints, their model leads to suboptimal solutions for batch plants as noted by Sundaramoorthy and Karimi[10] and also as demonstrated later in this paper. Most of the above-mentioned formulations have been based on either state-task network (STN) or resource-task network (RTN) process representations, except the model of Sundaramoorthy and Karimi,[10] which is based on generalized recipe diagrams.

The different time representations are summarized in Figure 1. In the uniform time discretization depicted in Figure 1a, the time horizon is divided into intervals of equal length that are common across all units.

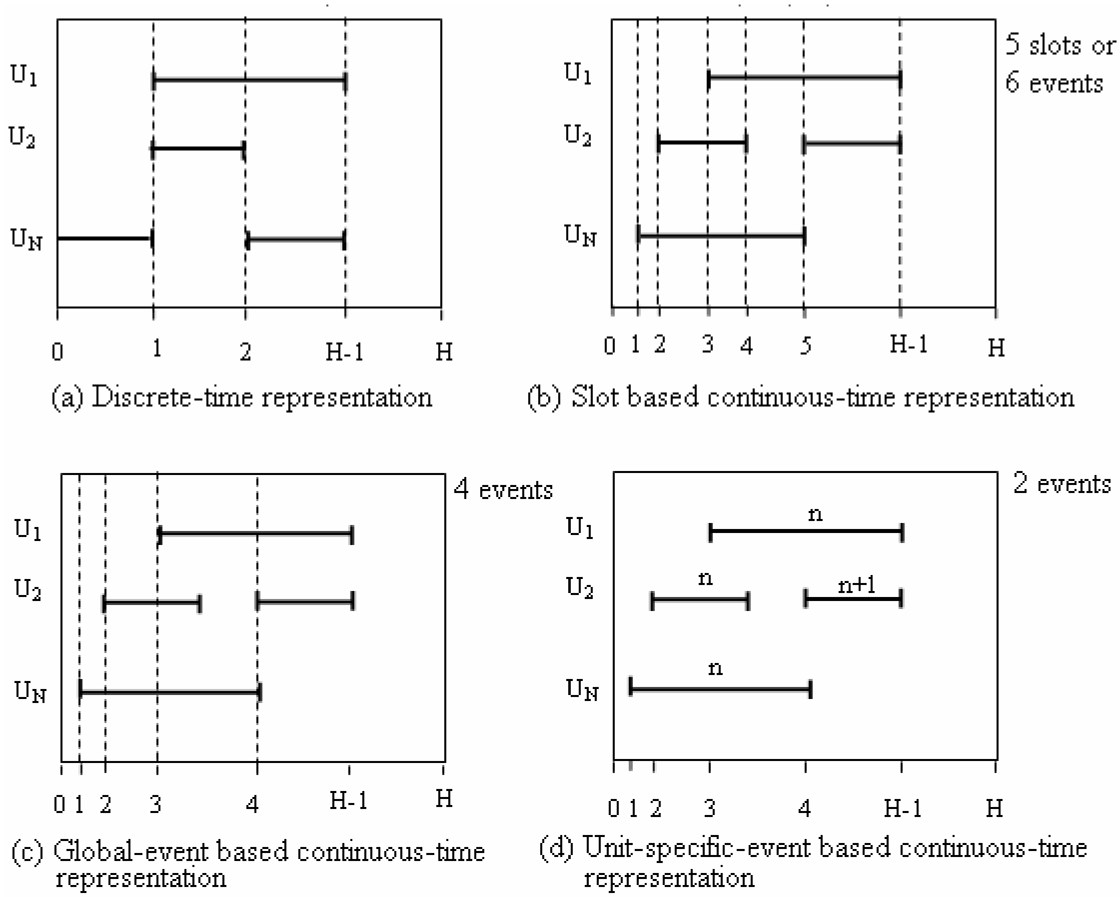


Figure 1. Different time representations.

Parts b–d of Figure 1 illustrate the different variations in the continuous-time representations. In the slot-based continuous-time representation of Figure 1b, the time horizon is divided into time intervals of unequal and unknown lengths, and typically tasks need to start and finish at an event (n slots are equivalent to n+1 events). In the global event-based continuous-time representation of Figure 1c, only the start times of the tasks need to be at an event point and the events considered are common across all units. In the unit-specific event-based time representation of Figure 1d, only the start time of each task in a unit has to be at an event point, whereas the occurrences of each event can be different across different units. For the specific instance of the four tasks considered on three units in parts b–d of Figure 1, the slot-based representation requires 5 slots (or 6 events), the global event-based representation requires 4 events, while the unit-specific event-based representation requires consideration of only 2 events. It should be noted that hybrid methods have also been developed[38] which combine mixed-integer linear programming (MILP) models with constraint programming. These are not within the scope of the presented comparison, which aims at evaluating pure MILP approaches for the aforementioned classes.

In this paper, we compare the ability of closing the integrality gap and evaluate the performance of the above-mentioned short-term scheduling formulations based on our implementations of these models. Specifically, we compare the slot-based models (Sundaramoorthy and Karimi[10]) versus global event-based models (Maravelias and Grossmann[28] and Castro and co-workers.[21,22]) versus the unit-specific event-based models (Ierapetritou and Floudas[29] and Giannelos and Georgiadis[37]), and study the computational effectiveness of each. Both network-represented and sequential processes are considered along with two scheduling objectives: maximization of profit and minimization of makespan. We also introduce two computational studies that compare the models of Maravelias and Grossmann,[28] Castro et al.,[22] and Janak et al.[35] for resource constraints.

The rest of the paper is organized as follows. In Section 2, we describe the different performance metrics with which the above-mentioned models are compared. The formulations for the different models used for comparison in this study are briefly discussed in Section 3, followed by the illustration of the benchmark examples in Section 4. The computational results and discussion for problems without resource constraints are detailed in Section 5 followed by computational studies with resource constraints in Section 6. The implemented formulations for each model are summarized in the appendices.

2. Description of Performance Metrics

Numerous formulations proposed in the literature, often claiming superiority over each other, exist for short-term scheduling of batch plants using continuous-time representations. Hence, for a fair and legitimate comparison of the different models with respect to their computational effectiveness, the following metrics are defined:

(a) Benchmark examples: The examples chosen are standard benchmark examples from the recent literature, used by many of the researchers in short-term scheduling of multipurpose batch plants. Both the STN- and RTN-based process representations are considered with variable batch sizes and processing times. The resource constraints considered are only those related to the raw material and equipment availability in the first three examples. Resource constraints related to utility requirements are considered in examples 4 and 5. The various models are evaluated for multiple instances of each problem with different time horizons and demand distributions and with respect to two dissimilar objective functions, maximization of profit and minimization of makespan. The latter objective of makespan minimization is considered to be more rigorous for assessing the performance of different models, as most of the models proposed in the literature have difficulties in closing the integrality gaps for this case.

(b) Completion to global optimality: While solving an optimization problem, completion to optimality in order to close the integrality gap is important for a fair comparison of different models. Unlike some of the comparisons reported in the literature, in this paper, all the problems are solved to zero integrality gap, except in some cases when one or more models take excessive computational time to solve to the reported global optimal solution, compared to the other models. For each model and for each instance of the various examples, we study parametrically the increase of the number of events or slots until there is no further improvement in the objective function, as suggested by Ierapetritou & Floudas[29].

(c) Computer hardware and software: The computer hardware and software used for comparing different models also has a noteworthy influence on the computational time taken to solve to zero integrality gap (as is also noted recently by Sundaramoorthy and Karimi[10]). The performance of the same model would be different on computers with different hardware (speed, RAM, etc.). Also, the computational performance would be different with a different version of the optimization software used (for instance, different versions of GAMS and its solvers). Hence, for a valid comparison, all the models are implemented on the same computer (3 GHz Pentium 4 with 2 GB RAM) and under similar conditions (GAMS distribution 21.1, CPLEX 8.1.0). For the solvers, only the default option values are used.

(d) Model implementation: In contrast to most of the comparisons reported in the literature, in this paper, the various formulations are compared based on our own implementation of the above-mentioned models. Before applying each model to the benchmark problems considered in this paper, each of the models is first reproduced against the examples presented in the original paper, to match the reported model statistics (number of variables and constraints) as best as possible. Sometimes, the reported statistics do not match with our implementation, possibly because of the usage of additional constraints not reported in the relevant paper. The model formulations we implemented are reported in the appendices.

(e) Model statistics: The number of binary variables and constraints resulting from a model has a significant impact on its computational performance. The different models are compared with respect to the resulting number of binary variables, the total number of continuous variables and constraints used, the total number of nodes explored to reach zero integrality gap, the computational times (CPU seconds) on the same computer, the objective function value at the relaxed node, and the number of nonzero elements in the resulting coefficient matrix. Although the number of nodes explored does not depend on the computer hardware, is often found to be mildly dependent on the order in which the constraints are written, for instance, while implementing the models in GAMS.[39] Also, the *M* value used in the big-M constraints may affect the value of the objective function at the relaxed node, and also may affect the computational time. However, we used a common value of *M* for all the models instead of exploring the best value of *M* for each model that requires big-M constraints.

3. Descriptions of Different Continuous-Time Models for Batch Plants

Six different continuous-time models for batch plants are considered in this comparative study. They have been selected on the basis of representing all possible classes: slot-based, global event-based, and unit-specific event-based models. Also, the papers of Castro and co-workers.,[21,22] Giannelos and Georgiadis,[37] Maravelias and Grossmann,[28] and Sundaramoorthy and Karimi[10] each provided comparisons with other approaches. The key features and the differences among the various continuous-time models compared in this paper are briefly discussed below, in chronological order.

3.1. Unit-Specific Event-Based Model of Ierapetritou and Floudas[29] (I&F). The authors presented the original concept of event points which correspond to a sequence of time instances located along the time axis of each unit, each representing the beginning of a task or the utilization of the unit. The location of event points is different for each unit, allowing different tasks to start at different times in each unit for the same event point. The timings of tasks are accounted through special sequencing constraints involving big-M constraints. No resources other than materials and equipment are considered. Although the model originally claimed its superiority due to both decoupling of task and unit events and nonuniform-time grid, later it became evident that it is primarily the introduction of the unit-specific events that gives the model the resulting cutting edge and makes it a class apart from all other models proposed in the literature. The resulting model requires less event points compared to the corresponding global-event or slot-based models, thus yielding better computational results, although big-M constraints are used. This model was later extended by Janak et al.,[35, 36] allowing tasks to spread over multiple events to accurately account for the utilization of different resources and storage policies. For the comparative study, in this paper, we use a slightly modified version of Ierapetritou and Floudas,[29] as presented in Appendix A.

3.2. Global Event-Based Model of Castro and co-workers.[21, 22] (CBM, CBMN). Castro et al.[21] (CBM) proposed a formulation using RTN representation for short-term scheduling of batch plants. The time horizon is divided into several global events that are common across all units. Binary variables are defined for assigning both start and end times of different tasks to the corresponding global events. Because of the unified treatment of various resources in the RTN framework, no special sequencing constraints are required. All the balances are written in terms of a single excess resource constraint, which implicitly includes the balances on the status and batch amounts of each unit. This model has no big-M constraints except for those that relate the extents of each task to the corresponding binary variables. Because of the provision for end times of tasks to be before the end times of the corresponding time slots, the processing time of each task on a given unit is not exactly represented but has an additional waiting period. Although the authors claimed superiority over the STN based event-driven formulation of Ierapetritou and Floudas,[29] it was established later (Ierapetritou and Floudas[31]) that the claims were based on incorrect data obtained from rounding off the parameter values used. Later, Castro et al.[22] (CBMN) proposed an improved model by eliminating some of the redundant binary and continuous variables and proposed new timing constraints that result in compact problem statistics and improved relaxed solutions. They compared the results for two different models (MN and MO) with the new and old timing constraints, respectively. On the basis of the request of a reviewer to compare with the new model of Castro et al.,[22] we choose the MN model for comparison because it has fewer constraints and better LP relaxed solution over the MO model. We also compare the performance of the MN model with that of Castro et al.[21] The models we implemented are reported in Appendix B. It should be noted that, in the model of Castro et al.,[22] there is an additional parameter (Δt) that defines a limit on the maximum number of events over which a task can occur, and it has a significant impact on the solution obtained, the computational time, and the problem statistics. At each event point, we need to iterate over this parameter to get the global optimal solution.

3.3. Unit-Specific Event-Based Model of Giannelos and Georgiadis[37] (G&G). The authors proposed an STN represented, unit-specific event-based formulation for short-term scheduling of multipurpose batch plants. This is a slight variation of the model proposed by Ierapetritou and Floudas,[29] wherein the authors relaxed the task durations using buffer times and implicitly eliminated the various big-M constraints of Ierapetritou and Floudas.[29] However, the authors introduced special duration and sequencing constraints that effectively transform the nonuniform time grid to a uniform one (global events) for the purposes of material balance and storage constraints. The start times (end times) of the tasks producing/consuming the same state were, respectively, forced to be the same, leading to suboptimal solutions, as observed by Sundaramoorthy and Karimi[10] and also as demonstrated later in this paper. The model we implemented is reported in Appendix C.

3.4. Global Event-Based Model of Maravelias and Grossmann[28] (M&G). This is a recent global event-based model using STN process representation. The model accounts for resource constraints other than equipment (utilities), various storage policies (unlimited, finite, zero wait, and no intermediate storage), and sequence-dependent changeover times and allows for batch mixing/splitting. This model reduces to the case of no resources, and it was used as such for comparison to other approaches (see Maravelias and Grossmann[28]). Global event points are used that are common across all units, and tasks are allowed to be processed over multiple events. Different binary variables are used to denote if a task starts, or continues over multiple events, or if it finishes processing a batch at a given event point. Also, a new class of tightening inequalities is proposed for tightening the relaxed LP solutions. The model we implemented for the computational studies in Section 5 is based on the reduction to no resources (refer to Appendix D), and it is included in this comparative study on the grounds that Maravelias and Grossmann[28] compared it to other continuous-time models without resources. Janak et al.[35,36] extended the basic event-based formulation of Ierapetritou and Floudas[29] to allow tasks to extend over multiple events in order to accurately account for different resource constraints and storage policies and provided a comparison for the case of resource constraints. At the request of a reviewer, we employ the models of Maravelias and Grossmann,[28] along with the models of Castro et al.[22] and Janak et al.[35,36] in a comparative study with resource constraints described in Section 6.

3.5. Slot-Based Model of Sundaramoorthy and Karimi[10] (S&K). Among the various slot-based formulations proposed in the literature, this is a recent model available for the short-term scheduling of multipurpose batch plants. The authors claim superior performance for the models they compared with, including those based on global events and that of Giannelos and Georgiadis.[37] They use generalized recipe diagrams for process representation, wherein a storage task is used to model the mixing and splitting of the same material streams. No resources other than materials and equipment are considered, and transfer and setup times are lumped into the batch processing times of tasks. The time horizon is divided into multiple time slots of varying lengths, and tasks are allowed to continue processing over multiple time slots. For each unit, binary variables are used to assign the beginning of each task to various time slots, and [0,1] continuous variables are used to denote tasks that continue over multiple slots and to denote tasks that release their batch amount at the end of a slot. An additional zero task is defined for modeling idling of units and to occupy extra redundant slots. Even though this model is categorized as slot-based, tasks are allowed to finish before the end of the time slot, making the model inherently similar to the global event-based models, except for the differences in accounting the various balances. Several balances are proposed based on status of each unit, material and storage constraints, and a new way of writing the balance on the time remaining on each unit, leading to a compact model. Some of the examples reported in their paper are not solved to zero integrality gap. In contrast to the authors' claim of 'absolutely' no big-M constraints, the readers can easily verify that there are typographical mistakes in the balances for the batch amount in a unit (constraints 11 and 12 of the original paper; see also Appendix E) which, if corrected, are similar to big-M constraints. Except for these constraints, the resulting model has no other big-M constraints. The model we implemented is reported in Appendix E.

4. Description of the Benchmark Examples

In this section, examples without resource constraints such as utility requirements are considered, and additional examples that include resource constraints are discussed later in Section 6. The following three benchmark examples, which have been studied by many authors, are considered from the short-term scheduling literature[10] for multipurpose batch plants with variable batch processing times. For simplicity, the process representations and the data for all three examples are shown using the STN representation. The processing time of task i on unit j is assumed to be a linear function, $\alpha_{ij} + \beta_{ij}B$, of its batch size, B.

4.1. Example 1. This is a simple motivating example from Sundaramoorthy and Karimi[10] involving a multipurpose batch plant that requires one raw material and produces two intermediates and one final product. The raw material is processed in three sequential tasks, where the first task is suitable in two units (J1 and J2), the second task is suitable in one unit (J3), and the third task is suitable in two units (J4 and J5). The STN for this motivating example is shown in Figure 2.

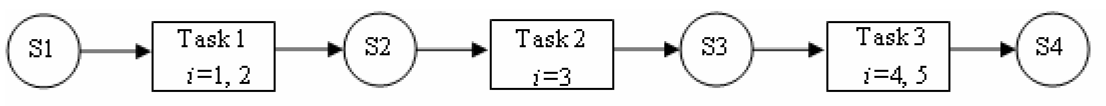


Figure 2. State-task network representation for example 1.

A task which can be performed in different units is considered as multiple, separate tasks, thus leading to five separate tasks (i=1,…,5), each suitable in one unit. The relevant data[10] of the constant (α_{ij}) and linear (β_{ij}) coefficients for processing times of different tasks (i), the suitable units (j), and their minimum ($B_{ij}^{\min}$) and maximum ($B_{ij}^{\max}$) batch sizes for all three examples considered are shown in Table 1. The storage capacities, initial stock levels, and prices of each state for all three examples are given in Table 2. The initial stock level for all intermediates is assumed to be zero and unlimited storage capacity is assumed for all states.

4. 2. Example 2. This is a standard example for short-term scheduling of multipurpose batch plants and has been studied comprehensively by several authors. Two different products are produced through five processing stages: heating, reactions 1, 2, and 3, and separation, as shown in the STN representation of the plant flow sheet in Figure 3. Since each of the reaction tasks can take place in two different reactors, each reaction is represented by two separate tasks. The relevant data[10, 28] is shown in Tables 1 and 2. The initial stock level for all intermediates is assumed to be zero and unlimited storage capacity is assumed for all states.

Table 1. Data of Coefficients of Processing Times of Tasks, Limits on Batch Sizes of Units for Examples 1-3

Task(i)	Unit(j)	α_{ij}	β_{ij}	$B_{ij}^{\min}$ (mu)	$B_{ij}^{\max}$ (mu)
		Example 1			
Task1 (i = 1)	Unit1	1.333	0.01333	---	100
(i = 2)	Unit2	1.333	0.01333	---	150
Task2 (i = 3)	Unit3	1.000	0.00500	---	200
Task3 (i = 4)	Unit4	0.667	0.00445	---	150
(i = 5)	Unit5	0.667	0.00445	---	150
		Example 2			
Heating (i = 1)	Heater	0.667	0.00667	---	100
Reaction1 (i = 2)	Reactor1	1.334	0.02664	---	50
(i = 3)	Reactor2	1.334	0.01665	---	80
Reaction2 (i = 4)	Reactor1	1.334	0.02664	---	50
(i = 5)	Reactor2	1.334	0.01665	---	80
Reaction3 (i = 6)	Reactor1	0.667	0.01332	---	50
(i = 7)	Reactor2	0.667	0.008325	---	80
Separation(i = 8)	Separator	1.3342	0.00666	---	200
		Example 3			
Heating1 (i = 1)	Heater	0.667	0.00667	---	100
Heating2 (i = 2)	Heater	1.000	0.01000	---	100
Reaction1(i = 3)	Reactor1	1.333	0.01333	---	100
(i = 4)	Reactor2	1.333	0.00889	---	150
Reaction2(i = 5)	Reactor1	0.667	0.00667	---	100
(i = 6)	Reactor2	0.667	0.00445	---	150
Reaction3(i = 7)	Reactor1	1.333	0.01330	---	100
(i = 8)	Reactor2	1.333	0.00889	---	150
Separation(i = 9)	Separator	2.000	0.00667	---	300
Mixing (i = 10)	Mixer1	1.333	0.00667	20	200
(i = 11)	Mixer2	1.333	0.00667	20	200

4.3. Example 3. This is a relatively complex example from Sundaramoorthy and Karimi[10] involving 11 tasks that can be performed in 6 units producing 13 states. The STN for this example is shown in Figure 4. This problem has several common characteristics of a multipurpose batch plant (i.e., a unit can perform either a single task or multiple tasks, a task can be performed in multiple units, etc.). Additionally, some of the intermediates have nonzero initial stock levels and unlimited storage capacity is assumed for all states. The relevant data[10] is shown in Tables 1 and 2.

Table 2. Data of Storage Capacities, Initial Stock Levels, and Prices of Various States for Examples 1-3[a]

	example 1			example 2			example 3		
state	storage capacity (mu)	initial stock (mu)	price ($/mu)	storage capacity (mu)	initial stock (mu)	price ($/mu)	storage capacity (mu)	initial stock (mu)	price ($/mu)
S1	UL	AA	0	UL	AA	0	UL	AA	0
S2	UL	0	0	UL	AA	0	UL	AA	0
S3	UL	0	0	UL	AA	0	UL	0	0
S4	UL	0	5	UL	0	0	UL	0	0
S5				UL	0	0	UL	0	0
S6				UL	0	0	UL	50	0
S7				UL	0	0	UL	50	0
S8				UL	0	10	UL	AA	0
S9				UL	0	10	UL	0	0
S10							UL	0	0
S11							UL	AA	0
S12							UL	0	5
S13							UL	0	5

[a] UL = Unlimited; AA = available as and when required.

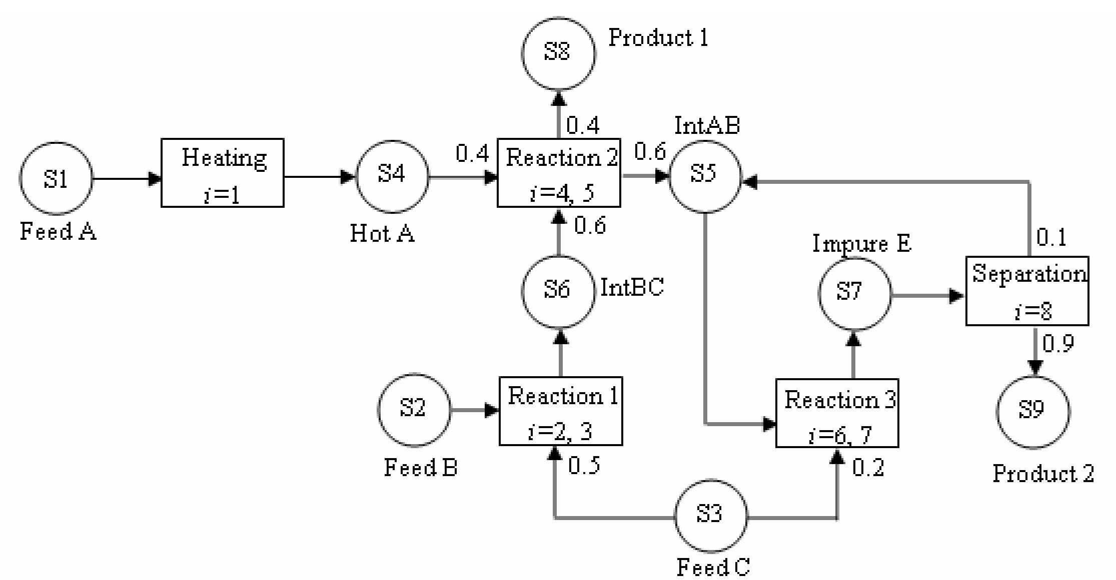


Figure 3. State-task network representation for example 2.

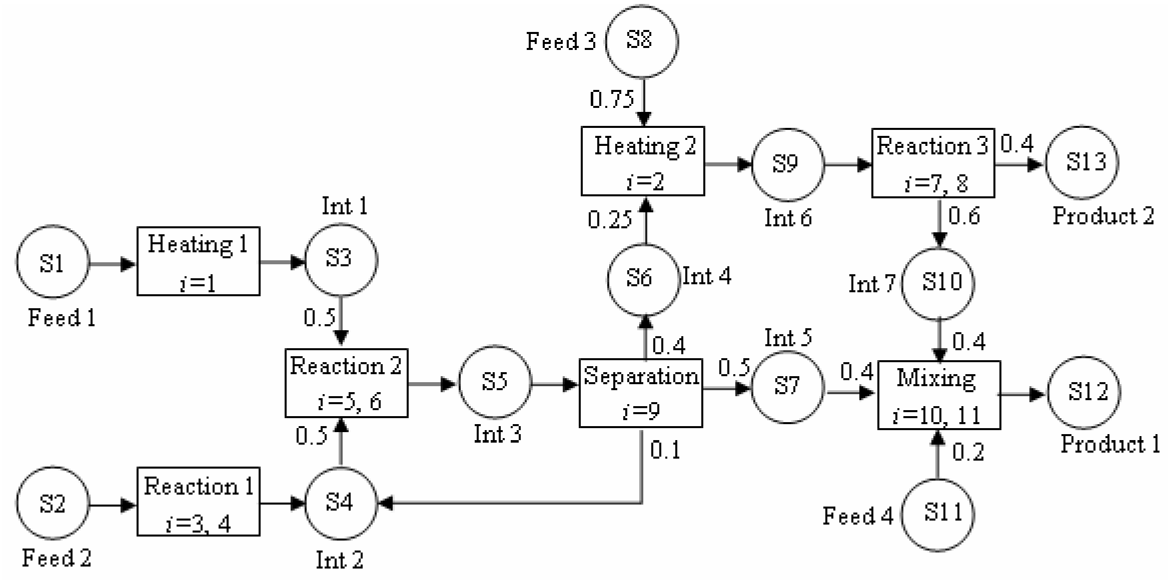


Figure 4. State-task network representation for example 3.

5. Computational Studies without Resource Constraints

Each of the six short-term scheduling models that are compared for examples without resource constraints are implemented and solved for all the above three examples with respect to two diverse objective functions: maximization of profit and minimization of makespan. The results using maximization of profit are discussed first in Section 5.1, followed by the results for minimization of makespan in Section 5.2. Under each category, several instances of demand distributions and different time horizons are considered for each example as done by Sundaramoorthy and Karimi.[10] All the resulting MILP models are solved in GAMS distribution 21.1 using CPLEX 8.1.0 on the same computer (3 GHz Pentium 4 with 2 GB RAM). The best performing model in each of the instances of different examples is shown on bold face in all the subsequent tables. It should be noted that, in the results by Sundaramoorthy and Karimi[10] the number of events reported in all the examples is misleading in the comparison tables. They need an additional event (k = 0) at the beginning of the time horizon as their numbering of events is from k = 0 to K, although they do not allow any tasks to start at k = K. Hence, what they appear to have reported is the number of slots, which is always one less than the total number of global events. In all the subsequent results reported in this study for the model of Sundaramoorthy and Karimi,[10] we show n event points to represent n-1 slots for a valid comparison with the other global-event and unit-specific event-based models.

5.1. Maximization of Profit. The computational results for each of the three examples are discussed below for the case of maximization of profit.

5.1.1. Example 1. This motivating example is solved for three different time horizons. The model statistics and computational results for all three cases are shown in Table 3.

Table 3. Model Statistics and Computational Results for Example 1 under Maximization of Profit

model	events	cpu time (s)	nodes	RMILP ($)	MILP ($)	binary variables	continuous variables	constraints	nonzeros
				Example 1a (H = 8)					
S&K	5	0.05	13	2000.0	1840.2	40	215	192	642
M&G	5	0.03	2	2000.0	1840.2	40	195	520	1425
CBM	5	0.04	0	2000.0	1840.2	70	115	201	655
CBMN(Δt=1)	5	0.01	0	2000.0	1840.2	20	70	86	274
(Δt=2)	5	0.02	7	2000.0	1840.2	35	85	116	414
G&G	4	0.01	0	2000.0	1840.2	20	76	122	355
I&F	**4**	**0.01**	**1**	**2000.0**	**1840.2**	**10**	**48**	**69**	**176**
	5	0.05	160	2804.6	1840.2	15	62	92	245
				Example 1b (H = 12)					
S&K	9	26.83	27176	4481.0	3463.6	80	415	408	1358
M&G	9	29.52	26514	4563.8	3463.6	80	375	1000	3415
CBM	9	26.93	23485	5237.6	3463.6	220	301	553	2099
CBMN(Δt=1)	9	0.23	606	4419.9	3301.6[a]	40	130	162	546
(Δt=2)	9	10.32	21874	5237.6	3463.6	75	165	232	886
G&G	6	0.03	22	3890.0	3301.6[a]	30	114	182	541
I&F	**6**	**0.03**	**24**	**4000.0**	**3463.6**	**20**	**76**	**115**	**314**
	7	0.19	589	4857.6	3463.6	25	90	138	383
				Example 1c (H = 16)					
S&K	12	5328.22	3408476	6312.6	5038.1	110	565	570	1895
	13	>67000[b]	36297619	6381.9	5038.1	120	615	624	2074
M&G	12	37675.13	17465450	6332.8	5038.1	110	510	1360	5275
	13	>67000[c]	20693001	6391.4	5038.1	120	555	1480	5965
CBM	12	32456.61	14385711	7737.6	5038.1	385	493	922	3707
	13	>67000[d]	22948021	8237.6	5038.1	450	567	1065	4343
CBMN(Δt=2)	12	1086.08	1642027	7737.6	5000.0[a]	105	225	319	1240
(Δt=3)	12	3911.14	4087336	7737.6	5038.1	150	270	409	1680
(Δt=3)	13	40466.83	44252075	8237.6	5038.1	165	295	448	1848
G&G	11	3.40	9533	6236.0	4840.9[a]	55	209	332	1006
I&F	**9**	**1.76**	**6596**	**6601.5**	**5038.1**	**35**	**118**	**184**	**521**
	10	20.60	89748	6601.5	5038.1	40	132	207	590

[a] Suboptimal solution. [b] Relative Gap = 1.24 %. [c] Relative Gap = 6.54%. [d] Relative Gap = 2.92%.

In the first scenario, for H = 8 h (example 1a), both the slot-based and global event-based models require 5 events, while the unit-specific event-based models require only 4 events. All the models perform equally well with respect to the computational time for this simple case. The model of Castro et al.[22] (CBMN), solved for $\Delta t = 1$, has the best model statistics among the slot-based/global event-based models, while the modified model of Ierapetritou and Floudas[29] presented in this paper performs the best with respect to the model statistics among all models. In all the results, the models are solved with additional event points until there is no further improvement in the objective function in order to ensure the optimal objective is obtained.

In the second scenario, we consider $H = 12$ h (example 1b). In this case, both the slot-based and global event-based models require 9 events, where only 6 events are required by the unit-specific event-based models. However, the model of Giannelos and Georgiadis[37] gives a suboptimal solution ($3301.6). This occurs not only for this example but also for several other instances of the other examples considered later, as is also shown recently by Sundaramoorthy and Karimi.[10] Note that we increased the number of events considered from 6 to 10, and their model is not able to find the global optimal solution ($3463.6), which is obtained by all the other models. When we remove the special sequencing constraints posed by Giannelos and Georgiadis,[37] then their model is similar to that of Ierapetritou and Floudas[29] and gives the global optimal solution. This indicates that the special sequencing constraints for enforcing the same start and finish times for all tasks consuming/producing the same state that were used by Giannelos and Georgiadis[37] are incorrect and lead to suboptimal solutions. The model of Castro et al.[22] ($\Delta t = 2$) performs better among the slot-based/global event-based models with respect to the computational time and problem statistics. However, it should be noted that their model[22] gives a suboptimal solution for $\Delta t = 1$, and hence, for a fair comparison with other models, we should add the CPU times and the number of nodes for all instances of Δt that need to be tested at each event point. The model of Castro et al.[21] (CBM) has the largest number of binary variables among all the models. The model of Ierapetritou and Floudas[29] performs the best among all the models, not only with respect to the model statistics but also with respect to the computational time. This indicates that, although there are big-M constraints in this model, the requirement of fewer events enables this model to outperform all other models.

Similar conclusions hold true for the third scenario of this example, which considers $H = 16$ h (example 1c), as is seen in Table 3. For this case, the slot-based/global event-based models require 12 event points compared to the model of Ierapetritou and Floudas,[29] which requires only 9 events and, hence, is computationally superior to all of the other models. The model of Giannelos and Georgiadis[37] gives a suboptimal solution ($4840.9), while all the other models are able to find the global optimal solution ($5038.1). The model of Castro et al.[22] (CBMN) gives a suboptimal solution ($5000) for Δt = 2, and hence, for fair comparison, we consider the total CPU time of 4997.22 s for Δt = 2 and Δt = 3 at 12 events. So, the model of Castro et al.[22] performs better among slot-based/global event-based models with respect to the computational time. Among all the models, the model of Maravelias and Grossmann[28] has the largest number of constraints and nonzeros while the model of Sundaramoorthy and Karimi[10] has the largest number of continuous variables. The Gantt charts for this case are shown in Figures 5 and 6 for the models of Ierapetritou and Floudas[29] and Castro et al.,[22] respectively.

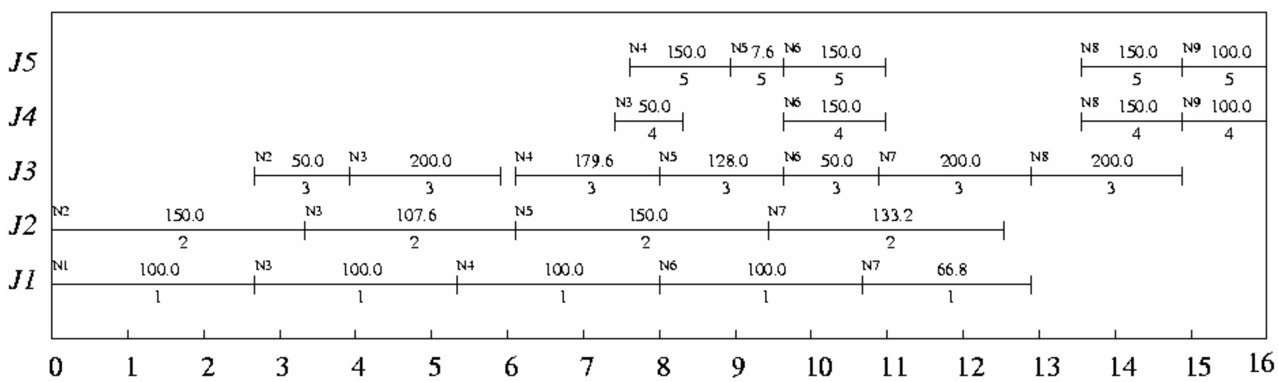


Figure 5. Gantt chart for example 1c (9 events) using I&F model under maximization of profit.

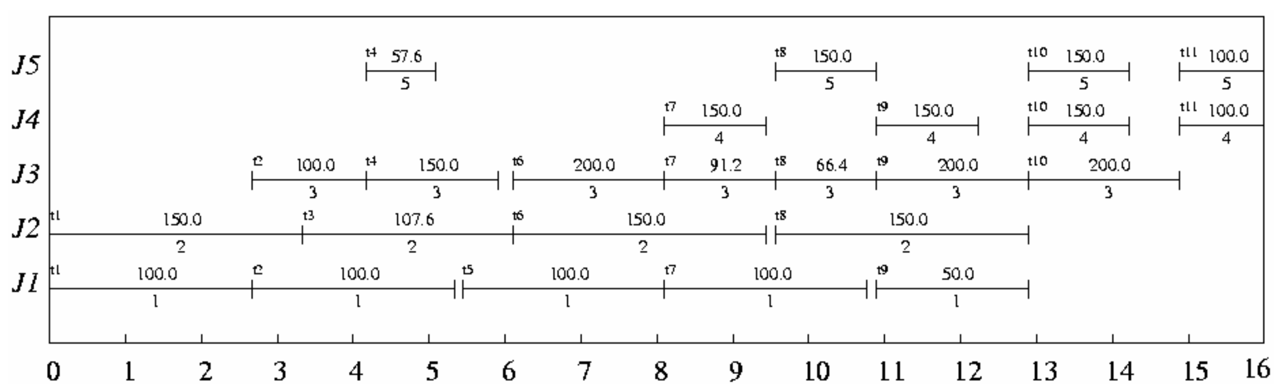


Figure 6. Gantt chart for example 1c (12 events) using CBMN model under maximization of profit.

Note that, for the model of Castro et al.[22] in Figure 6, there is an additional event at the end at which no task occurs (so a total of 12 events). When we consider an additional slot/event point, the slot-based/global event-based models take excessive CPU times (shown in Table 3), while the unit-specific event-based model of Ierapetritou and Floudas[29] takes only 21 s to find the same global optimal solution.

5.1.2. Example 2. This example problem is also solved for two different time horizons. The model statistics and computational results for both the cases are shown in Table 4.

Table 4. Model Statistics and Computational Results for Example 2 under Maximization of Profit

model	events	CPU time (s)	nodes	RMILP ($)	MILP ($)	binary variables	continuous variables	constraints	nonzeros
				Example 2a (H = 8)					
S&K	5	0.07	4	1730.9	1498.6	48	235	249	859
M&G	5	0.16	26	1730.9	1498.6	64	360	826	2457
CBM	5	0.07	8	1812.1	1498.6	112	184	322	1105
CBMN(Δt=1)	5	0.01	4	1730.9	1498.6	32	104	114	439
G&G	4	0.03	14	1812.1	1498.6	32	142	234	820
I&F	**4**	**0.03**	**13**	**1812.1**	**1498.6**	**18**	**90**	**165**	**485**
	5	0.28	883	2305.3	1498.6	26	115	216	672
				Example 2b (H = 12)					
S&K	7	1.93	1234	3002.5	2610.1	72	367	387	1363
	8	29.63	16678	3167.8	2610.3	84	433	456	1615
	9	561.58	288574	3265.2	2646.8	96	499	525	1867
	10	10889.61	3438353	3315.8	2646.8	108	565	594	2119
	11	>67000[b]	17270000	3343.4	2646.8[a]	120	631	663	2371
M&G	7	2.15	814	3002.5	2610.1	96	526	1210	4019
	8	58.31	17679	3167.8	2610.3	112	609	1402	4884
	9	2317.38	611206	3265.2	2646.8	128	692	1594	5805
	10	>67000[c]	10737753	3315.8	2646.8	144	775	1786	6782
	11	>67000[d]	9060850	3343.4	2658.5	160	858	1978	7815
CBM	7	1.38	1421	3190.5	2610.1	216	316	572	2146
	8	35.81	30202	3788.3	2610.3	280	394	721	2791
	9	1090.53	680222	4297.9	2646.8	352	480	886	3519
	10	40355.97	19225950	4770.8	2646.8	432	574	1067	4330
	11	>67000[e]	13393455	5228.7	2627.9[a]	520	676	1264	5224
CBMN(Δt=2)	7	0.63	1039	3045.0	2610.1	88	188	224	1050
	8	14.39	32463	3391.0	2610.3	104	218	261	1238
	9	331.72	593182	3730.5	2646.8	120	248	298	1426
	10	4366.09	6018234	4070.0	2646.8	136	278	335	1614
	11	>67000[f]	80602289	4409.5	2646.8[a]	152	308	372	1802
G&G	6(to 11)	0.33	701	3190.5	2564.6[a]	48	208	348	1238
I&F	**7**	**6.19**	**14962**	**3788.3**	**2658.5**	**42**	**165**	**318**	**1046**
	8	105.64	211617	4297.9	2658.5	50	190	369	1233

[a] Suboptimal solution. [b] Relative Gap = 1.59%. [c] Relative Gap = 3.16%. [d] Relative Gap = 5.12%. [e] Relative Gap = 28.16%. [f] Relative Gap = 2.58%.

In the first scenario, for H = 8 h (example 2a), both the slot-based and global event-based models require 5 events, while the unit-specific event-based models require only 4 events. All the models perform equally well with respect to the computational time for this simple case. The model of Castro et al.[22] (CBMN) for $\Delta t = 1$ has better problem statistics among slot-based/global event-based models, while the model of Ierapetritou and Floudas[29] performs the best among all models with respect to the model statistics. The model of Castro et al.[21] (CBM) requires the largest number of binary variables, while the model of Maravelias and Grossmann[28] requires the largest number of continuous variables, constraints, and nonzeros for this case.

In the second scenario, we consider H = 12 h (example 2b). In the paper by Sundaramoorthy and Karimi,[10] they report the results for this case for the first feasible solution of 7 events only using finite intermediate storage for the intermediate states. In this work, as already mentioned, we solve the problem to its global optimal solution and for zero integrality gap, assuming unlimited intermediate storage for all states. For this case, the slot-based/global event-based models require at least 11 event points compared to the model of Ierapetritou and Floudas,[29] which requires only 7 events. The slot-based/global event-based models are not solved until zero integrality gap as they take excessive computational time and because the model of Ierapetritou and Floudas[29] solves to the global optimal solution in just 6.19 s. The model of Giannelos and Georgiadis[37] gives a suboptimal solution ($2564.6). The slot-based/global event-based models take excessive computational time, and only the model of Maravelias and Grossmann[28] is able to solve to the global optimal solution. The model of Castro et al.[21] (CBM) has poor LP relaxation and requires more binary variables. The Gantt charts for this case are shown in Figures 7 and 8 for the models of Ierapetritou and Floudas[29] and Maravelias and Grossmann,[28] respectively.

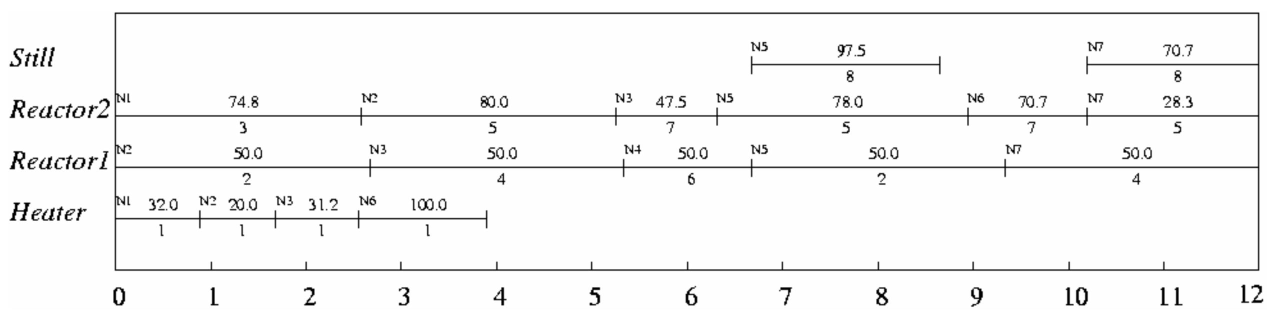


Figure 7. Gantt chart for example 2b (7 events) using I&F model under maximization of profit.

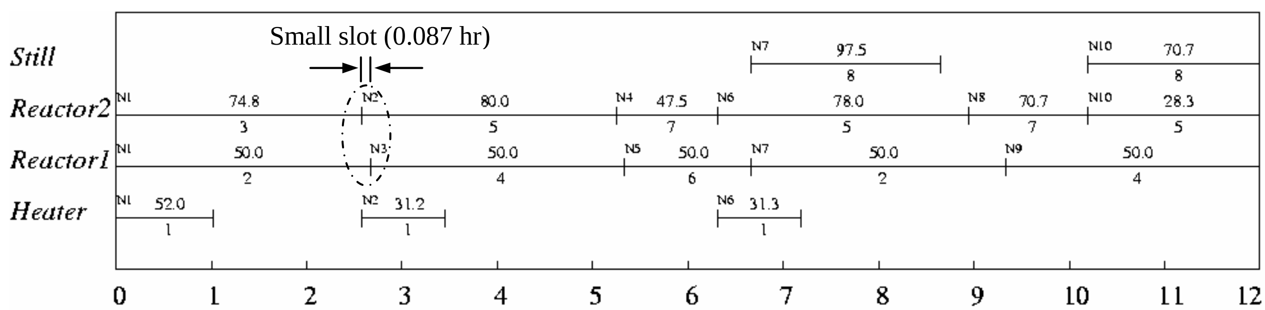


Figure 8. Gantt chart for example 2b (11 events) using M&G model under maximization of profit.

Interestingly, in the Gantt chart of Figure 8 for the model of Maravelias and Grossmann,[28] it can be observed that, it corresponds to the requirement of a very tiny slot of duration 0.087 h (in the second slot) for the slot-based/global event-based models to find the reported global optimal solution. This is evidenced by the excessive CPU time taken by the slot-based model of Sundaramoorthy and Karimi[10] for which the global optimal solution is not obtained. However, in the Gantt chart of Figure 7 for the model of Ierapetritou and Floudas,[29] it is evident that such a slot would not be necessary as they use a unit-specific event-based model. Hence, this case emphasizes the important difference between the slot-based/global event-based models and the unit-specific event-based models. Because of the different alignment of the start times of different units, sometimes the slot-based/global event-based models may require very small slots which can result in a very large number of event points, and may prohibit the realization of global optimal solution in reasonable CPU time compared to the unit-specific event-based models. The unit-specific event-based models consider the time horizon in a "true" continuous-time form without requiring such tiny time slots and lead to the requirement of a relatively lower number of event points. Thus, this example clearly demonstrates the distinct advantages of the unit-specific event-based models over the slot-based/global event-based models, despite the presence of big-M constraints in the former. The model of Ierapetritou and Floudas[29] takes only 6.19 s compared to the model of Maravelias and Grossmann,[28] which takes >67 000 s for obtaining the same global optimal solution. The model of Sundaramoorthy and Karimi,[10] although has no big-M constraints, is not able to find the global optimal solution in reasonable CPU time.

5.1.3. Example 3. This relatively complex example is solved for two different time horizons. The model statistics and computational results for both the cases are shown in Table 5. In the first scenario, for H = 8 h (example 3a), both the slot-based and global event-based models require 7 events, while the unit-specific event-based models require only 5 events. The model of Giannelos and Georgiadis[37] gives a suboptimal solution ($1274.5), while all the other models are able to find the global optimal solution ($1583.4). The model of Castro et al.[22] (CBMN) for Δt = 2 performs best with respect to the computational time among the slot-based/global event-based models, although it requires more binary variables compared to the model of Sundaramoorthy and Karimi.[10] The model of Maravelias and Grossmann[28] has the largest number of continuous variables, constraints, and nonzeros for this case. The model of Ierapetritou and Floudas[29] performs the best with respect to both the model statistics and the computational time when compared to all of the other models.

Table 5. Model Statistics and Computational Results for Example 3 under Maximization of Profit

model	events	CPU time (s)	nodes	RMILP ($)	MILP ($)	binary variables	continuous variables	constraints	nonzeros
				Example 3a (H = 8)					
S&K	7	184.46	145888	2513.8	1583.4	102	597	584	2061
M&G	7	1012.68	429949	2560.6	1583.4	132	717	1667	5601
CBM	7	19.82	13130	2809.4	1583.4	297	433	841	3049
CBMN(Δt=2)	7	6.90	10361	2606.5	1583.4	121	264	343	1495
G&G	5	0.35	807	2100.0	1274.5[a]	55	244	392	1335
I&F	**5**	**0.38**	**1176**	**2100.0**	**1583.4**	**30**	**155**	**303**	**875**
	6	25.92	57346	2847.8	1583.4	41	190	377	1139
				Example 3b (H = 12)					
S&K	9	372.92	94640	3867.3	3041.3	136	783	792	2789
	10	>71000[b]	12781125	4029.7	3041.3	153	876	896	3153
M&G	9	19708.33	2254227	3867.3	3041.3	176	943	2195	8114
	10	>71000[c]	5857914	4029.7	2981.7[a]	198	1056	2459	9492
CBM	9	290.84	80123	4059.4	3041.3	484	658	1307	5001
	10	16416.31	3194816	4615.6	3041.3	594	787	1576	6154
CBMN(Δt=2)	9	107.97	47798	3864.3	3041.3	165	348	457	2031
	10	1173.82	751686	4189.8	3041.3	187	390	514	2299
G&G	6	1.18	2750	2871.9	2443.2[a]	66	290	469	1608
I&F	**7**	**18.33**	**15871**	**3465.6**	**3041.3**	**52**	**225**	**451**	**1403**
	8	50.48	41925	4059.4	3041.3	63	260	525	1667

[a] Suboptimal solution. [b] Relative Gap = 3.76%. [c] Relative Gap = 12.85%.

Similar conclusions can be drawn for the second scenario where H = 12 h (example 3b). In this case, both the slot-based and global event-based models require 9 events, while only 7 events are required by the unit-specific event-based models. The model of Giannelos and Georgiadis[37] again gives a suboptimal solution ($2443.2), while all the other models are able to find the global optimal solution ($3041.3). The model of Castro et al.[22] (CBMN) for $\Delta t = 2$ performs best for this case with respect to the computational time among the slot-based/global event-based models, although it requires more binary variables compared to the model of Sundaramoorthy and Karimi[10]. The model of Maravelias and Grossmann[28] has the largest number of continuous variables, constraints, and nonzeros for this case as well. The model of Ierapetritou and Floudas[29] again performs the best both with respect to the model statistics and the computational time when compared to all the other models. The Gantt charts for this case are shown in Figures 9 and 10 for the models of Ierapetritou and Floudas[29] and Castro et al.,[22] respectively. When we consider an additional slot/event point, the models of Sundaramoorthy and Karimi[10] and Maravelias and Grossmann[28] take excessive CPU times (> 71000 s, as shown in Table 5), while the models of Castro and co-workers.[21,22] perform better. The model of Ierapetritou and Floudas[29] again performs the best at higher event points as well.

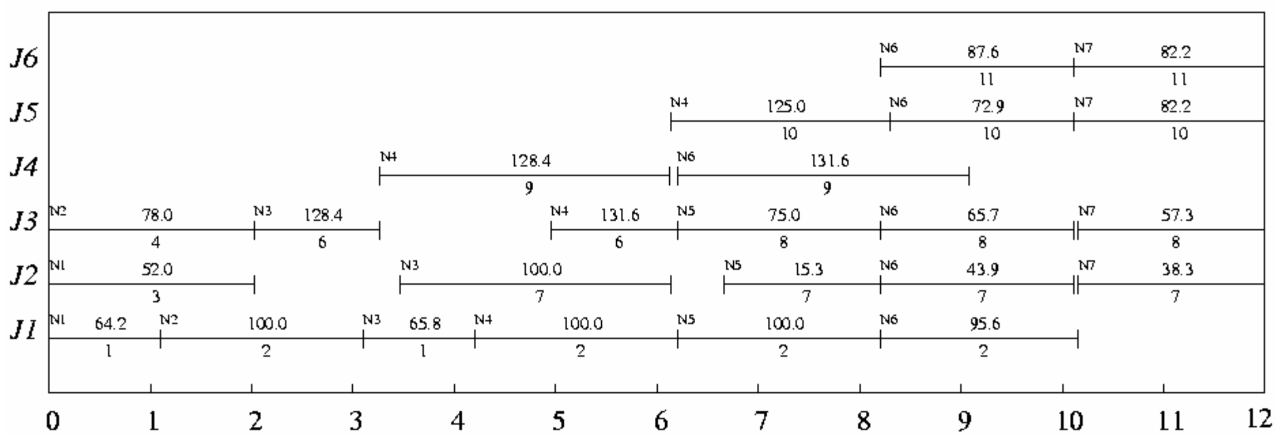


Figure 9. Gantt chart for example 3b (7 events) using I&F model under maximization of profit.

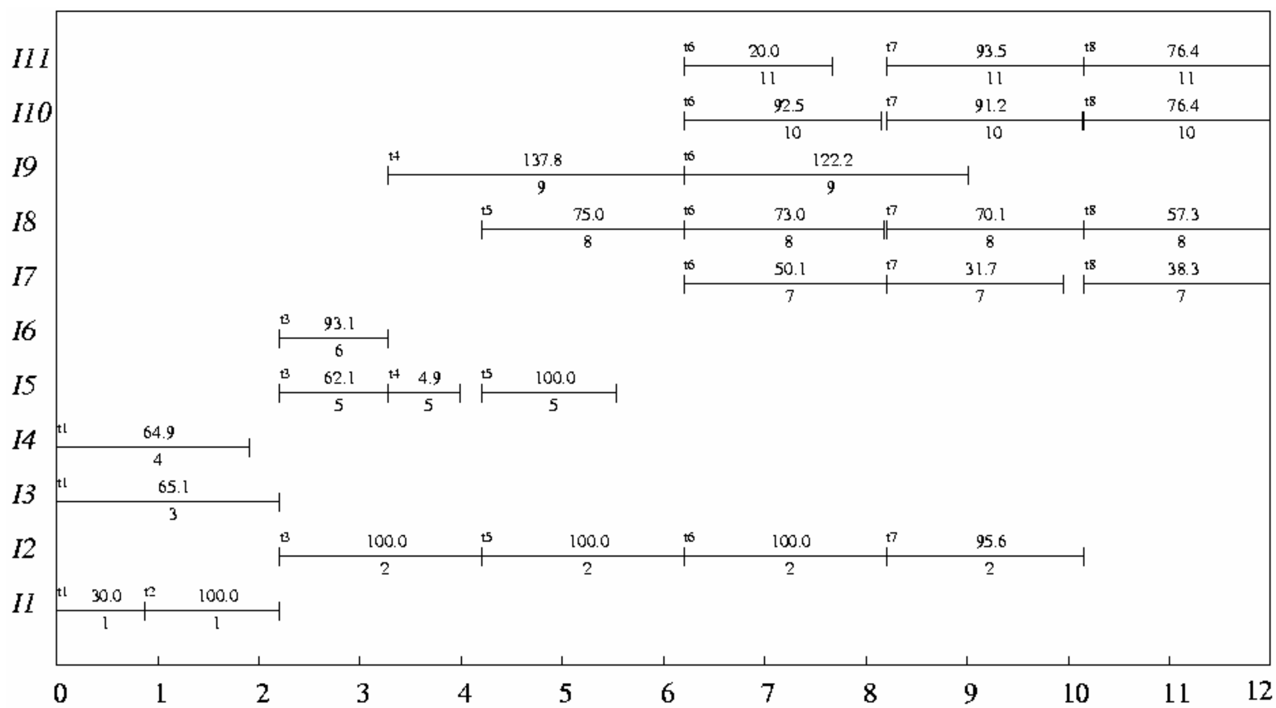
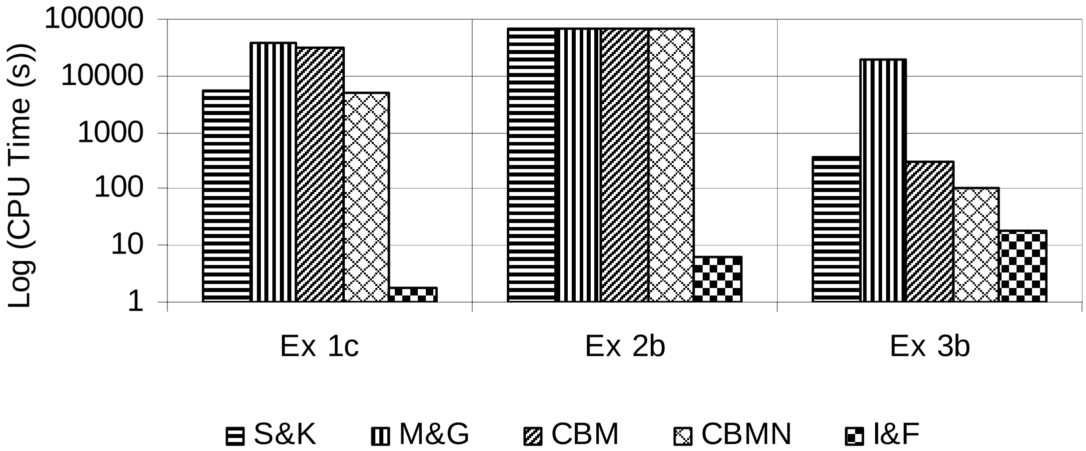


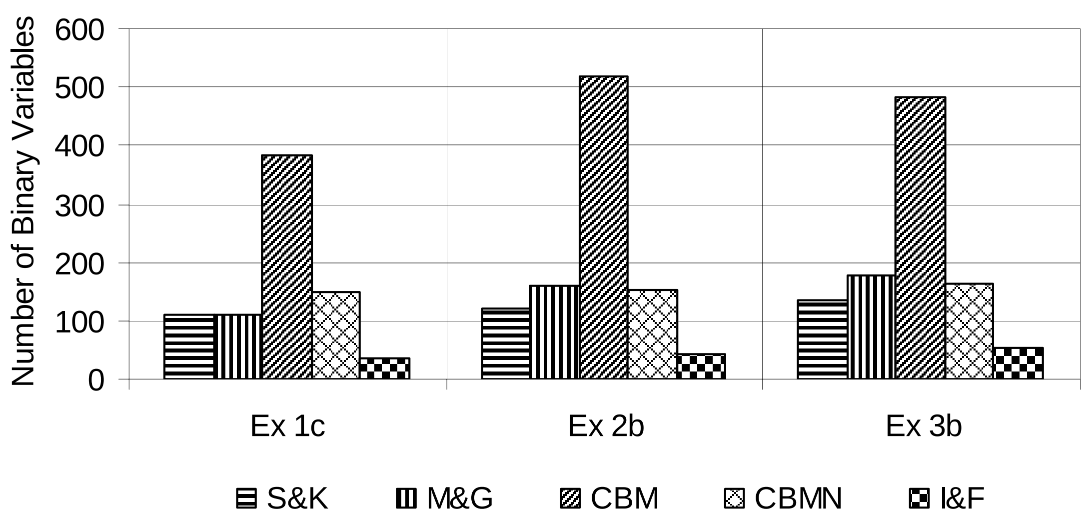
Figure 10. Gantt chart for example 3b (9 events) using CBMN model under maximization of profit.

The CPU times for representative examples of all the models (except Giannelos and Georgiadis,[37] as it gives suboptimal solutions) for the objective of maximization of profit are depicted in Figure 11. It should be noted that, the models of Sundaramoorthy and Karimi[10] and Castro and co-workers.[21,22] yield suboptimal solutions for example 2b. The number of binary variables for each model is shown in Figure 12. Thus, with respect to the objective of maximization of profit, in all the instances of the three examples considered, it can be seen that the unit-specific event-based model of Ierapetritou and Floudas[29] performs the best in terms of both problem size and computational performance and is orders of magnitude better than the other models. If we consider the cumulative CPU time of increasing events until the same optimal solution is found for each model, then it is evident from Tables 3–5 that, the solution statistics for both the slot-based and global event-based models would be even more inferior compared to the unit-specific event-based model.



Note: S&K, CBM, and CBMN yield suboptimal solutions for Ex2b

Figure 11. CPU times of different models for maximization of profit.



Note: S&K, CBM, and CBMN yield suboptimal solutions for Ex2b

Figure 12. Number of binary variables in different models for maximization of profit.

5.2. Minimization of Makespan. Finding optimal solutions for problems where the minimization of the makespan is the objective function is reported in the literature to be the most difficult scheduling problem to solve, even for simple examples. Although Sundaramoorthy and

Karimi[10] claim that their model performs very well for this objective function, it can be seen later that, for some instances, their model also does not find the global optimal solutions (even at higher event points) which are obtained by the unit-specific event-based model of Ierapetritou and Floudas.[29] In the results reported in Sundaramoorthy and Karimi,[10] for most of the examples using makespan minimization, each problem is not solved to zero integrality gap, using finite storage for the intermediate states. However, in this work, we solved all the models for all the examples to zero integrality gap assuming unlimited intermediate storage for all states, and by considering an increasing number of event points in order to find the global optimal solutions.

The computational results for each of the three examples are discussed below for the objective of minimization of makespan. The data for all three examples is the same except we consider fixed demands and the time horizon (*H*) is varied. For all the models that have big-M constraints (Maravelias and Grossmann[28] and Ierapetritou and Floudas[29]), we need to specify the horizon time as well. For fair comparison, we consider the same values of *H* used by Sundaramoorthy and Karimi.[10] Note that there is no need to specify *H* for the model of Giannelos and Georgiadis[37] when solving makespan minimization problems.

5.2.1. Example 1. This motivating example is solved for two different demand scenarios. The model statistics and computational results for both cases are shown in Table 6. In the first scenario (example 1a), we consider a demand for state S4 (D_4 = 2000 mu), and H = 50 h is used for the models involving big-M constraints. It can be observed that the optimal solution obtained by the model of Ierapetritou and Floudas[29] for 12 events (28.439 h) is better than the best solutions obtained by all other models, even at higher event points. For the models of Sundaramoorthy and Karimi[10], Maravelias and Grossmann,[28] and Castro et al.,[21] using 16 events, it takes relatively longer time compared to Castro et al.,[22] even though better solutions are not obtained. The model of Giannelos and Georgiadis[37] results in a suboptimal solution using 12–15 event points. For 17 events, the models of Sundaramoorthy and Karimi[10] and Castro et al.[22] (CBMN using Δt = 2) the optimal solution found is 28.773 h, for which both the models take more than 80000 s. The model of Ierapetritou and Floudas[29] outperforms all the other models in terms of computational time and problem size and finds the global optimal solution of 27.881 h using 14 events with a CPU time of just 41.89 s. The Gantt chart for this case is shown in Figure 13 for the model of Ierapetritou and Floudas.[29]

Table 6. Model Statistics and Computational Results for Example 1 under Minimization of Makespan

model	events	H	CPU time (s)	nodes	RMILP (h)	MILP (h)	binary variables	continuous variables	constraints	nonzeros
Example 1a (D_4=2000 mu)										
S&K	13	--	1.18	362	27.126	29.772	120	615	624	2074
	14	--	31.54	15622	25.702	29.772	130	665	678	2253
	15	--	728.05	400789	25.142	29.439	140	715	732	2432
	16	--	37877.69	12064418	24.871	29.106	150	765	786	2611
	17	--	>80000[b]	17279722	24.716	28.773[a]	160	815	840	2790
M&G	13	50	2.19	394	27.126	29.772	120	556	1485	6056
	14		645.06	139488	25.335	29.772	130	601	1605	6786
	15		25253.81	5273904	25.024	29.439	140	646	1725	7551
	16		>90000[c]	11258561	24.834	29.106[a]	150	691	1845	8351
CBM	13	--	0.50	6	23.313	29.772	450	568	1066	4404
	14	--	14.90	4262	21.049	29.772	520	647	1219	5095
	15	--	2163.55	454549	19.049	29.439	595	731	1382	5836
	16	--	64850.69	9852772	17.049	29.106[a]	675	820	1555	6627
CBMN(Δt=1)	13	--	0.02	0	27.126	29.772	60	191	239	824
(Δt=1)	14	--	0.11	65	25.824	29.772	65	206	258	892
(Δt=1)	15	--	0.28	417	25.358	29.772	70	221	277	960
(Δt=2)	15	--	235.90	236250	19.049	29.439	135	286	407	1605
(Δt=2)	16	--	27994.64	23426601	17.049	29.106	145	306	436	1723
(Δt=2)	17	--	>80000[d]	80105289	15.049	28.773[a]	155	326	465	1841
G&G	12	--	0.03	0	27.126	29.772	60	228	367	1108
	15	--	1.87	3529	25.064	29.772[a]	75	285	457	1387
I&F	12	50	0.12	208	24.236	28.439	50	160	253	732
	13		2.26	7863	24.236	27.903	55	174	276	801
	14		**41.89**	**134961**	**24.236**	**27.881**	**60**	**188**	**299**	**870**
	15		950.64	2693556	24.236	27.881	65	202	322	939
Example 1b (D_4= 4000 mu)										
S&K	23	--	101.03	34598	51.362	56.432	220	1115	1164	3864
	24	--	15814.03	4164921	49.939	56.432[a]	230	1165	1218	4043
M&G	23	100	21974.42	2525960	51.362	56.432	220	1006	2685	14931
	24		>90000[e]	5129168	49.572	57.765[a]	230	1051	2805	16011
CBM	23	--	6.09	185	43.313	56.432	1375	1583	3046	13564
	24	--	2016.50	136348	41.049	56.432[a]	1495	1712	3299	14755
CBMN(Δt=1)	23	--	0.05	0	51.362	56.432	110	341	429	1504
(Δt=2)	24	--	0.20	72	50.061	56.432	115	356	448	1572
(Δt=2)	25	--	>80000[f]	34358380	39.049	56.099[a]	235	486	697	2785
G&G	22	--	0.07	0	51.362	56.432	110	418	667	2038
	24	--	1.53	1707	49.594	56.432[a]	120	456	727	2224
I&F	22	100	6.48	19019	48.473	52.433	100	300	483	1422
	23		**384.12**	**832372**	**48.473**	**52.072**	**105**	**314**	**506**	**1491**

[a] Suboptimal solution. [b] Relative Gap = 4.22%. [c] Relative Gap = 7.38%. [d] Relative Gap = 0.12%. [e] Relative Gap = 11.01%. [f] Relative Gap = 2.18%.

Similar conclusions hold true for the second scenario (example 1b) where the demand is D_4 = 4000 mu, and H = 100 h is used for the models involving big-M constraints. The model of Ierapetritou and Floudas[29] outperforms the other models and finds the global optimal solution of 52.072 h using 23 events with a CPU time of 384.12 s.

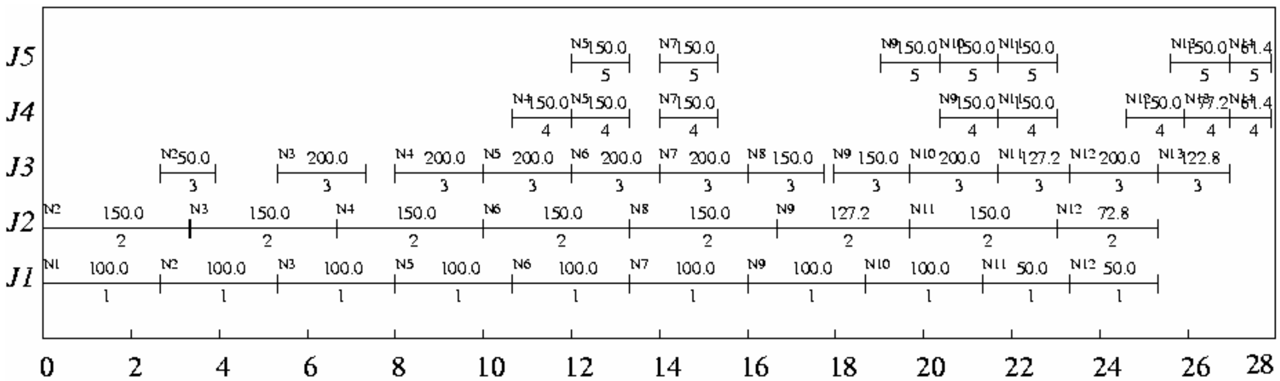


Figure 13. Gantt chart for example 1a (14 events) using I&F model under minimization of makespan.

5.2.2. Example 2. This problem is solved with demands for states S8 and S9 ($D_8 = D_9 = 200$ mu) and $H = 50$ h is used for the models involving big-M constraints. The model statistics and computational results are shown in Table 7.

Table 7. Model Statistics and Computational Results for Example 2 under Minimization of Makespan

model	events	H	CPU time (s)	nodes	RMILP (h)	MILP (h)	binary variables	continuous variables	constraints	nonzeros
				Example 2 ($D_8 = D_9 = 200$ mu)						
S&K	9	--	10.98	5378	18.685	19.789	96	556	528	1936
	10	--	519.35	142108	18.685	19.340	108	622	597	2188
	11	--	11853.03	2840768	18.685	19.340	120	688	666	2440
M&G	9	50	66.55	15674	18.685	19.789	128	693	1598	5869
	10		5693.53	1066939	18.685	19.340	144	776	1790	6850
	11		>80000[b]	5019315	18.685	19.340	160	859	1982	7887
CBM	9	--	7.75	6426	12.555	19.789	352	481	888	3584
	10	--	727.23	441130	9.889	19.340	432	575	1069	4403
	11	--	32258.74	13776145	7.223	19.340	520	677	1266	5305
CBMN(Δt=1)	9	--	0.71	1809	18.685	19.789	64	193	216	872
(Δt=1)	10	--	50.49	134189	18.685	19.789	72	215	241	979
(Δt=2)	10	--	56.11	109917	15.654	19.340	136	279	337	1623
(Δt=2)	11	--	5535.27	8389012	12.988	19.340	152	309	374	1811
G&G	8	--	1.97	3804	12.555	19.789	64	274	475	1675
	10	--	1614.35	1182082	10.475	19.789[a]	80	340	589	2093
I&F	8	50	0.78	1008	12.738	19.764	45	190	367	1211
	9		**74.26**	**111907**	**12.477**	**19.340**	**53**	**215**	**418**	**1398**
	10		1672.11	2079454	12.435	19.340	61	240	469	1585

[a] Suboptimal solution. [b] Relative Gap = 2.03%.

For this case, all the models except one are able to find the optimal solution of 19.34 h; however, the model of Ierpateritou & Floudas[29] requires 1 less event point. The model of Giannelos and Georgiadis[37] did not find the optimal solution using 8–10 event points. The model of Maravelias and Grossmann[28] has the largest computational time and the largest number of continuous variables, constraints, and nonzeros. The model of Castro et al.[22] (CBMN) performs better among slot-based/global event-based models as it takes an overall of 106.6 s using 10 events (for Δt = 1 and Δt = 2). The model of Ierapetritou and Floudas[29] has the least number of binary and continuous variables and performs the best with respect to the computational time as well using only 9 events. The Gantt charts for this case are shown in Figures 14 and 15 for the models of Ierapetritou and Floudas[29] and Castro et al.,[22] respectively.

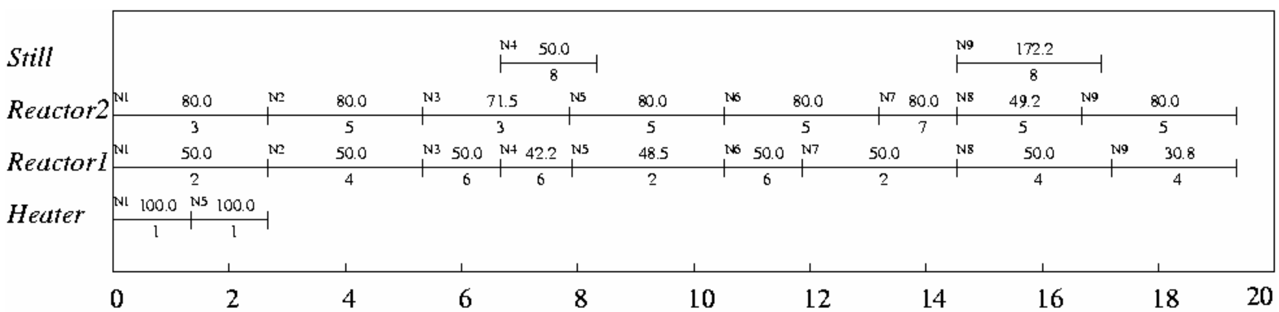


Figure 14. Gantt chart for example 2 (9 events) using I&F model under minimization of makespan.

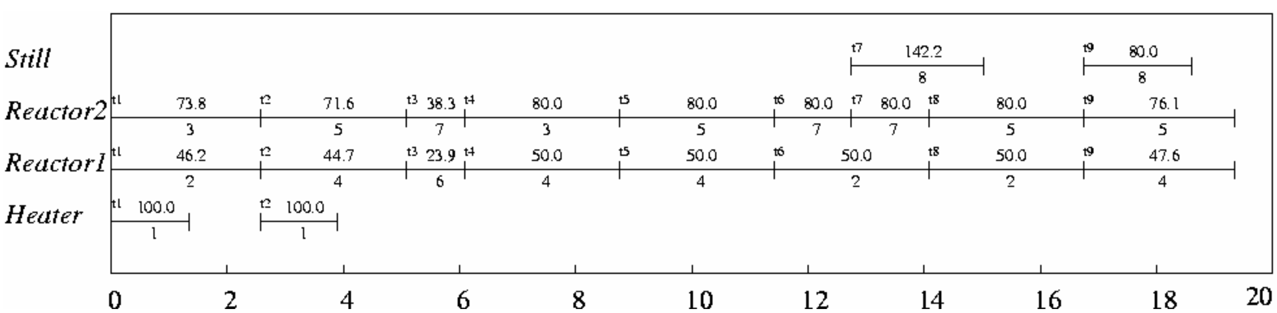


Figure 15. Gantt chart for example 2 (10 events) using CBMN model under minimization of makespan.

5.2.3. Example 3. This relatively complex example is solved for two different demand scenarios. The model statistics and computational results for both the cases are shown in Table 8. In the first scenario (example 3a), we consider demands for state S12 and S13 (D_{12} = 100mu, D_{13} = 200 mu), and H = 50 h is used for the models involving big-M constraints. All the models are solved for an increasing number of event points. The slot-based/global event-based models require 11 events to find the optimal solution of 13.367 h. The model of Maravelias and Grossmann[28] takes excessive CPU time (>80 000 s with 3.1% gap) and has the largest number of continuous variables, constraints, and nonzeros. The model of Castro et al.[21] solves faster among the slot-based/global event-based models using 11 events, although it has very poor LP relaxation. There is no improvement in the model of Giannelos and Georgiadis[37] from 7–9 events, and each yields a suboptimal solution. The model of Ierapetritou & Floudas[29] requires just 7 events and outperforms the other models in terms of both exceptional computational performance (0.38 s vs 2514.97 s taken by Castro et al.[21]) and least problem size. The Gantt charts for this case are shown in Figures 16 and 17 for the models of Ierapetritou and Floudas[29] and Castro et al.,[21] respectively.

Table 8. Model Statistics and Computational Results for Example 3 under Minimization of Makespan

model	events	H	CPU time (s)	nodes	RMILP (h)	MILP (h)	binary variables	continuous variables	costraints	nonzeros
Example 3a (D_{12} = 100 mu, D_{13} = 200 mu)										
S&K	8	--	0.28	36	12.317	14.366	119	690	689	2425
	9	--	13.32	5156	11.621	13.589	136	783	793	2789
	10	--	226.83	53789	11.417	13.532	153	876	897	3153
	11	--	4340.65	821194	11.335	13.367	170	969	1001	3517
	12	--	>80000[1]	11858901	11.295	13.367	187	1062	1105	3881
M&G	8	50	1.15	316	12.317	14.366	154	831	1937	6905
	9		126.77	21366	11.621	13.589	176	944	2201	8208
	10		3949.36	605450	11.417	13.532	198	1057	2465	9592
	11		>80000[2]	7481387	11.335	13.367	220	1170	2729	11057
CBM	8	--	0.62	68	10.941	14.366	385	541	1064	4044
	9	--	5.89	2762	8.941	13.589	484	659	1309	5090
	10	--	53.42	22452	6.941	13.532	594	788	1578	6254
	11	--	2514.97	673460	4.941	13.367	715	928	1871	7536
	12	--	>80000[3]	20380858	3.825	13.367	847	1079	2188	8936
CBMN(Δt=2)	8	--	0.23	67	12.192	14.366	143	307	402	1776
(Δt=2)	9	--	2.23	2566	10.192	13.589	165	349	459	2044
(Δt=2)	10	--	14.73	17426	8.192	13.532	187	391	516	2312
(Δt=2)	11	--	312.07	326752	6.192	13.532	209	433	573	2580
(Δt=3)	11	--	20230.89	16842943	6.192	13.367	297	521	725	3494
(Δt=3)	12	--	11547.29	5054232	4.635	13.367	330	574	801	3877
G&G	7	--	0.25	338	11.066	14.616	77	336	558	1902
	9	--	3.36	3960	10.167	14.616[a]	99	428	712	2448
I&F	**7**	**50**	**0.38**	**458**	**11.066**	**13.367**	**52**	**225**	**452**	**1413**
	8		2.89	3506	10.000	13.367	63	260	526	1677
Example 3b (D_{12} = D_{13} = 250 mu)										
S&K	11	--	981.01	226238	14.535	17.357[a]	170	969	1001	3517
M&G	11	100	62724.36	5802875	14.535	17.357[a]	220	1170	2729	11057
CBM	11	--	38.14	9627	10.722	17.357[a]	715	928	1871	7536
CBMN(Δt=2)	11	--	31.57	28079	12.494	17.357[a]	209	433	573	2580
G&G	10	--	59.26	84970	12.763	18.978[a]	110	474	789	2721
I&F	**10**	**100**	**2.50**	**2668**	**12.500**	**17.025**	**85**	**330**	**674**	**2205**
	11		396.58	424617	12.500	17.025	96	365	748	2469

[a] Suboptimal solution; Relative Gap: 1.01 %[1], 3.055 %[2], 3.29 %[3]

When we consider an additional slot/event point, except the model of Castro et al.,[22] the slot-based/global event-based models take excessive CPU times (>80 000 s, as shown in Table 8) while the unit-specific event-based model of Ierapetritou and Floudas[29] takes only 2.89 s to find the same global optimal solution.

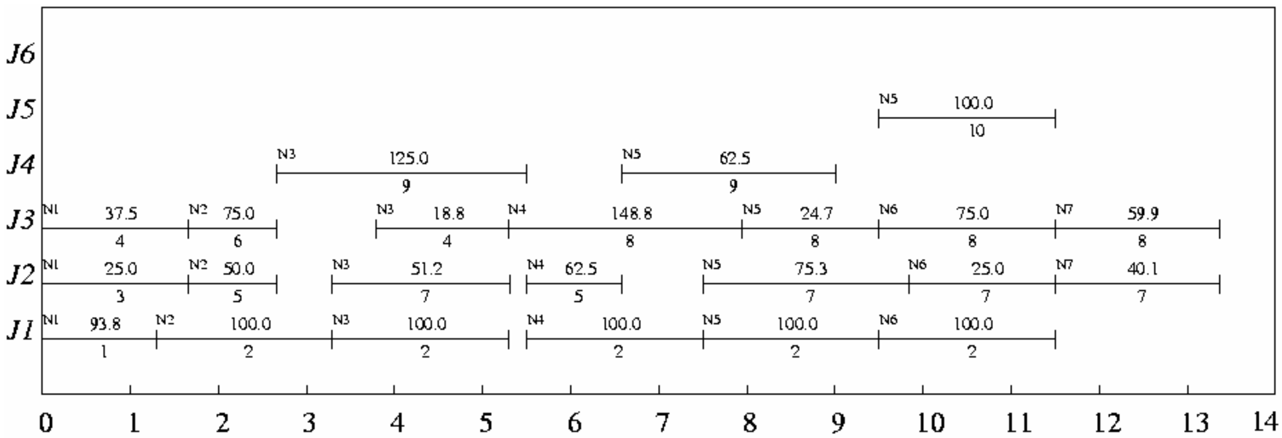


Figure 16. Gantt chart for example 3a (7 events) using I&F model under minimization of makespan.

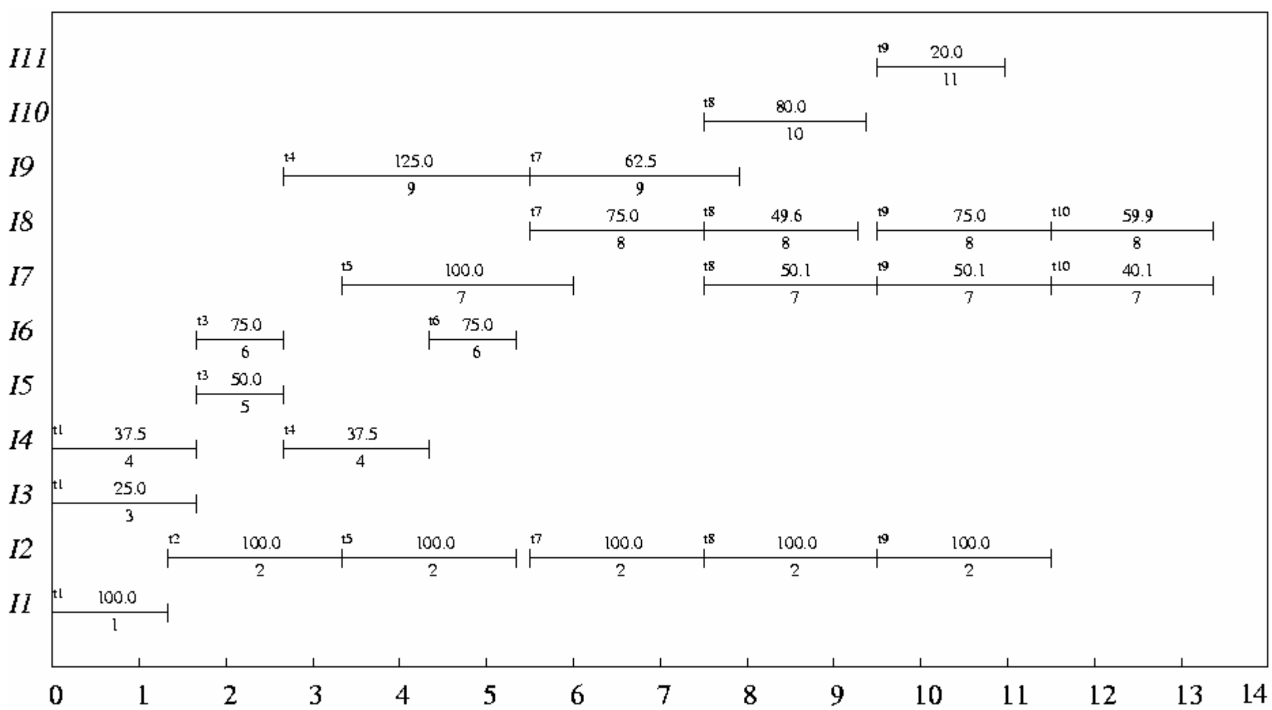
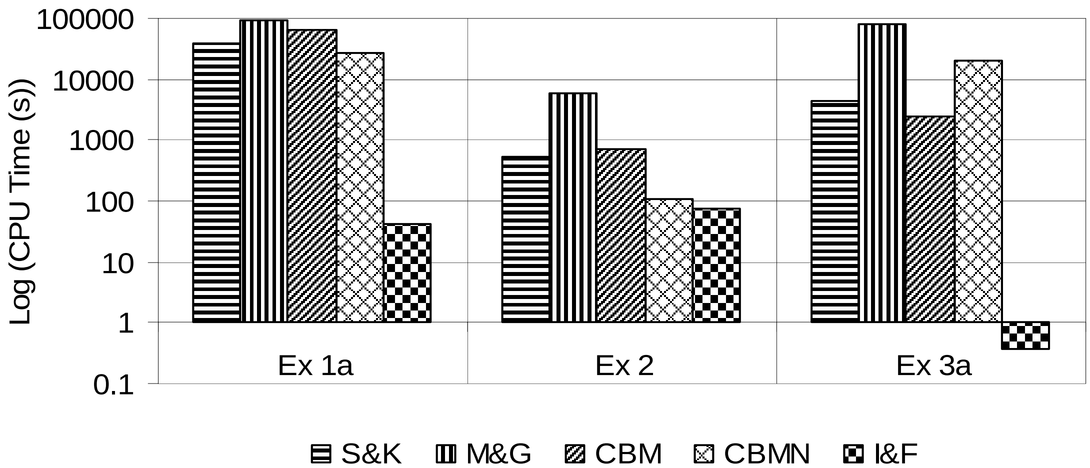


Figure 17. Gantt chart for example 3a (11 events) using CBM model under minimization of makespan.

Similar conclusions hold true for the second scenario (example 3b), where the demands are D_{12} = D_{13} = 250 mu and H = 100 h is used for the models involving big-M constraints. All the slot-based/global event-based models require 11 events to find the suboptimal solution of 17.357 h. The model of Giannelos and Georgiadis[37] gives a suboptimal solution (18.978 h) using 10 events.

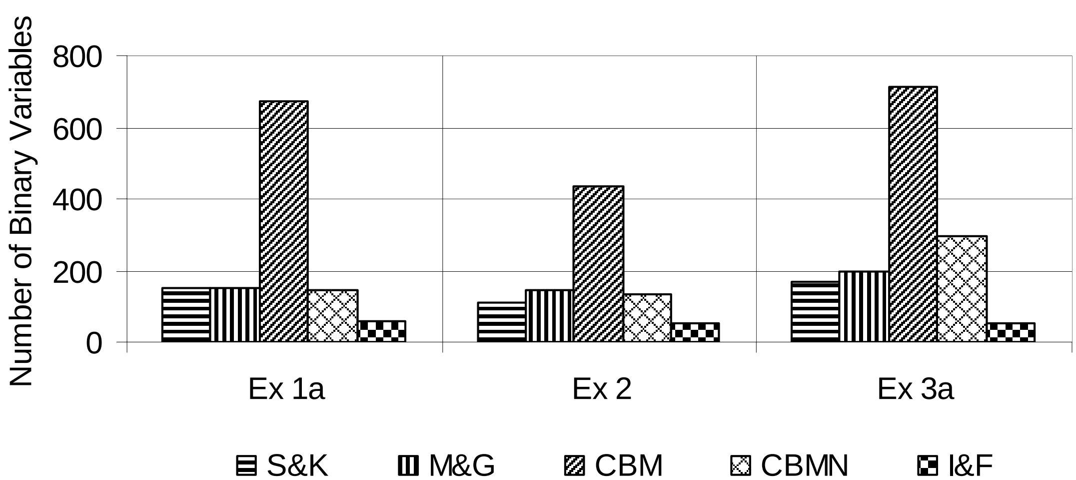
The model of Castro et al.[22] using 11 events (for $\Delta t = 2$) solves faster among the slot-based/global event-based models, but it provides a suboptimal solution. However, the model of Ierapetritou & Floudas[29] finds the global optimal solution of 17.025 h in 2.5 s and, hence, outperforms the other models.

The CPU times for representative examples of all the models (except Giannelos and Georgiadis[37] as it gives suboptimal solutions) for the objective of minimization of makespan are depicted in Figure 18. The number of binary variables for each model is shown in Figure 19.



Note: S&K, M&G, CBM, and CBMN yield suboptimal solutions for Ex1a

Figure 18. CPU times of different models for minimization of makespan.



Note: S&K, M&G, CBM, and CBMN yield suboptimal solutions for Ex1a

Figure 19. Number of binary variables in different models for minimization of makespan.

It should be noted that, all the slot-based/global event-based models of Sundaramoorthy and Karimi[10], Maravelias and Grossmann,[28] and Castro and co-workers[21,22] yield suboptimal solutions for example 1a, example 1b, and example 3b. It can be observed that, for the objective of minimization of makespan as well, the unit-specific event-based model of Ierapetritou & Floudas[29] outperforms all the other models by orders of magnitude and is able to find global optimal solutions in all cases. If we consider the cumulative CPU time of increasing events until the global optimal solution is found for each model, then it is evident from Tables 6–8 that the solution statistics for both the slot-based and global event-based models would be even more inferior compared to the unit-specific event-based model.

6. Computational Studies with Resource Constraints

Even though a comparative study of approaches with resource constraints was provided in Janak et al.,[35,36] at the request of a reviewer, we consider here additional examples that include resource constraints such as utility requirements and mixed storage policies. For the global event-based formulations, the models of Maravelias and Grossmann[28] and Castro et al.,[22] and for the unit-specific event-based formulations, the model of Janak et al.,[35,36] are considered in the comparative study.

6.1. Example 4. This example, which includes resource constraints, variable batch sizes and processing times, and utility requirements, was solved by Maravelias and Grossmann[28] and Janak et al.[35] The STN for this example is shown in Figure 20, and the corresponding data[28,35] is given in Tables 9 and 10.

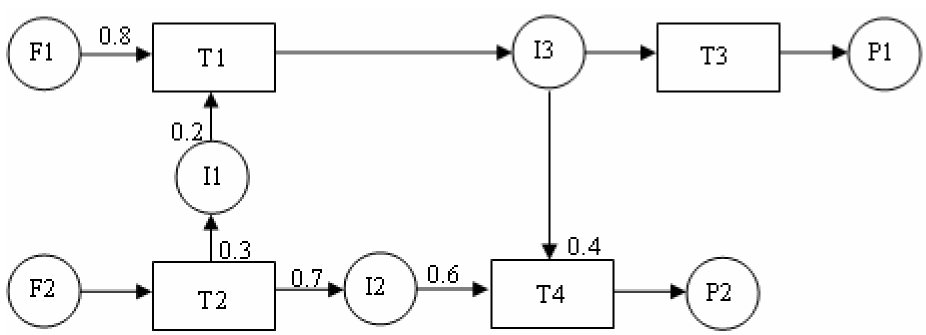


Figure 20. STN for example 4

Table 9. State Related Data for Example 4

	F1	F2	I1	I2	I3	P1	P2
ST_s^{max} (kg)	1000	1000	200	100	500	1000	1000
ST_s^0 (kg)	400	400	0	0	0	0	0
$price_s$($/kg)	0	0	0	0	0	30	40

Table 10. Task Related Data for Example 4[a]

			T1		T2		T3		T4		T1		T2		T3		T4	
	cap^{min}	cap^{max}	α	β	α	β	α	β	α	β	γ_{iHS}	δ_{iHS}	γ_{iCW}	δ_{iCW}	γ_{iHS}	δ_{iHS}	γ_{iCW}	δ_{iCW}
R1	40	80	0.5	0.025	0.75	0.0375					6	0.25	4	0.3				
R2	25	50	0.5	0.4	0.75	0.06					4	0.25	3	0.3				
R3	40	80					0.25	0.0125	0.5	0.025					8	0.4	4	0.5

[a] cap^{min}/cap^{max} in kg, α in h, β in h/kg, γ in kg/min, and δ in kg/min per kg of batch.

There are two types of reactors available for the process (types I and II), with two reactors of type I (R1 and R2) and one reactor of type II (R3) with four reactions suitable in them. Reactions T1 and T2 require a type I reactor, whereas reactions T3 and T4 require a type II reactor. Additionally, reactions T1 and T3 are endothermic, where the required heat is provided by steam (HS) available in limited amounts. Reactions T2 and T4 are exothermic, and the required cooling water (CW) is also available in limited amounts. Each reactor allows variable batch sizes, where the minimum batch size is half the capacity of the reactor. The processing times and the utility requirements include a fixed time and a variable term that is proportional to the batch size. The processing times are set so that the minimum batch size is processed in 60% of the time needed for the maximum batch size. For the raw materials and final products, unlimited storage is available, while for the intermediates, finite storage is available. Two different cases of this example studied in the literature[28,35] are considered that differ in the resource availability. In the first case (example 4a), we assume that the availability of both HS and CW is 40 kg/min, and in the second case (example 4b), it is 30 kg/min. Also, two different objective functions, maximization of profit and minimization of makespan, are considered.

6.1.1. Maximization of Profit. For the objective of maximization of profit and a time horizon of 8 h, the optimal solution is $5904.0 in the first case (example 4a) and $5227.778 in the second case (example 4b). The computational results in terms of the model statistics and the CPU times are reported in Table 11 for the models of Maravelias and Grossmann[28] (M&G), Janak et al.[35] (JLF), and Castro et al.[22] (CBMN).

6.1.2. Minimization of Makespan. For the objective of minimization of makespan, the optimal solution is 8.5 h in the first case (example 4a) and 9.025 h in the second case (example 4b). The computational results in terms of the model statistics and the CPU times are reported in Table 12 for the models of Maravelias and Grossmann[28] (M&G), Janak et al.[35] (JLF), and Castro et al.[22] (CBMN). For the models involving big-M constraints,[28,35] a common value of M =10 is used.

Table 11. Model Statistics and Computational Results for Example 4 under Maximization of Profit

model		events	CPU time (s)	nodes	RMILP ($)	MILP ($)	binary variables	continuous variables	constraints	nonzeros
					Example 4a					
M&G		7	1.22	680	8870.5	5904.0	72	545	1082	4184
CBMN	(Δt=1)	7	0.30	376	8875.4	5482.04[a]	36	140	175	741
	(Δt=2)	7	1.07	1312	10396.7	5904.0	66	170	250	1216
JLF		6	1.03	294	10981.8	5904.0	45	273	1304	4606
					Example 4b					
M&G		6	0.27	67	7267.1	5227.8	60	470	925	3411
CBMN	(Δt=1)	6	0.09	96	7685.7	5000.0[a]	30	121	148	622
	(Δt=2)	6	0.16	112	8360.3	5227.8	54	145	208	1002
JLF		5	0.15	26	6414.7	5227.8	33	220	1028	3265

[a] Suboptimal solution

Table 12. Model Statistics and Computational Results for Example 4 under Minimization of Makespan

Model		Events	CPU time (s)	Nodes	RMILP (h)	MILP (h)	Binary variables	Continuous variables	Constraints	Nonzeros
					Example 4a					
M&G		8	9.20	3331	5.48	8.5	84	620	1241	5039
CBMN	(Δt=1)	8	0.77	995	5.47	9.25[a]	42	159	204	861
	(Δt=2)	8	5.89	5641	2.37	8.5	78	195	294	1431
JLF		7	1.95	180	6.27	8.5	57	326	1601	6234
					Example 4b					
M&G		7	0.74	197	5.85	9.025	72	545	1084	4209
CBMN	(Δt=1)	7	0.24	229	5.81	9.25[a]	36	140	177	742
	(Δt=2)	7	0.72	491	3.05	9.025	66	170	252	1217
JLF		6	0.59	10	6.35	9.025	45	273	1317	4693

[a] Suboptimal solution

For both the objective functions, the unit-specific event-based model of Janak et al.[35] requires 1 event point less and has the least number of binary variables compared to the global event-based models of Maravelias and Grossmann[28] and Castro et al.[22]. The model of Castro et al.[22] (using Δt = 2) requires the least number of continuous variables, constraints and nonzeros. It should be noted that the model of Castro et al.[22] (CBMN) yields suboptimal solution for Δt = 1 in both the cases and hence, the overall CPU time and the number of nodes (for both Δt = 1 and Δt = 2) should be considered.

6. 2. Example 5. This example comprises of resource constraints, mixed storage policies, variable batch sizes and processing times, and utility requirements that was solved by Maravelias and Grossmann,[28] Janak et al.,[35] and Castro et al.[22]. The STN for this example is shown in Figure 21, and the relevant data is given in Tables 13 and 14.

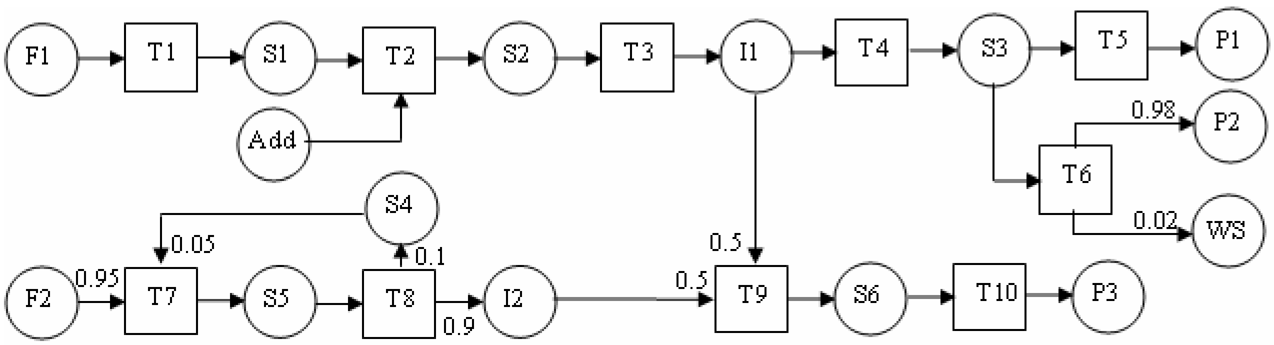


Figure 21. STN for example 5

Table 13. State Related Data for Example 5

	F1	F2	S1	S2	S3	S4	S5	S6	I1	I2	P1	P2	P3
ST_s^{max} (kg)	∞	∞	0	0	15	40	0	0	∞	∞	∞	∞	∞
ST_s^0 (kg)	100	100	0	0	0	10	0	0	0	0	0	0	0
$price_s$($/kg)											1	1	1

Table 14. Task Related Data for Example 5[a]

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
unit	U1	U2	U3	U1	U4	U4	U5	U6	U5	U6
cap^{max}	5	8	6	5	8	8	3	4	3	4
α	2	1	1	2	2	2	4	2	2	3
utility	LPS	CW	LPS	HPS	LPS	HPS	CW	LPS	CW	CW
γ	3	4	4	3	8	4	5	5	5	3
δ	2	2	3	2	4	3	4	3	3	3

[a] cap^{max} in kg, α in h, γ in kg/min, and δ in kg/min per kg of batch.

The plant consists of 6 units involving 10 processing tasks and 14 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 (ZW) policy 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 maximization of profit, and two instances of time horizons of 12 h (example 5a) and 14 h (example 5b) are considered.

For the objective of maximization of profit and a time horizon of 12 h, the optimal solution is \$13000 in the first case (example 5a), and for a time horizon of 14 h, the optimal solution is \$16350 in the second case (example 5b). The computational results in terms of the model statistics and the CPU times are reported in Table 15 for the models of Maravelias and Grossmann[28] (M&G), Janak et al.[35] (JLF), and Castro et al.[22] (CBMN).

Table 15. Model Statistics and Computational Results for Example 5 under Maximization of Profit

model	events	CPU time (s)	nodes	RMILP (\$)	MILP (\$)	binary variables	continuous variables	constraints	nonzeros
				Example 5a (H = 12)					
M&G	9	63.63	18150	18423.5	13000	160	993	2184	7282
CBMN(Δt=1)	9	0.26	291	18388.7	10000[a]	80	320	464	1618
(Δt=2)	9	4.35	3767	21063.3	13000	150	390	688	2668
JLF	8	1.33	115	24000	13000	109	743	3100	11309
				Example 5b (H = 14)					
M&G	8	0.89	94	18648.6	16350	140	875	1923	6163
CBMN(Δt=1)	8	0.06	1	18473.7	15000[a]	70	286	409	1422
(Δt=2)	8	0.15	10	18696.4	16350	130	346	601	2322
JLF	7	0.42	48	18960.4	16350	91	643	2690	9258

[a] Suboptimal solution

It should be noted that, for the model of Janak et al.,[35] two additional storage tasks are defined explicitly, while for the models of Maravelias and Grossmann[28] and Castro et al.,[22] no storage tasks are required. The unit-specific event-based model of Janak et al.[35] requires 1 event point less and has the least number of binary variables compared to the global event-based models of Maravelias and Grossmann[28] and Castro et al.[22] The model of Castro et al.[22] (using Δt = 2) requires the least number of continuous variables, constraints, and nonzeros. It should be noted that the model of Castro et al.[22] (CBMN) yields a suboptimal solution for Δt = 1 in both cases, and hence, the overall CPU time and the number of nodes (for both Δt = 1 and Δt = 2) need to be considered.

7. Conclusion

In this paper, we compare and assess the performance of various continuous-time models proposed in the literature for short-term scheduling of multipurpose batch plants. These models are broadly classified into three distinct categories: slot-based, global event-based, and unit-specific event-based formulations. On the basis of our implementation, the models are compared using several benchmark example problems from the literature. Two different objective functions, maximization of profit and minimization of makespan, are considered, and the models are compared with respect to different metrics such as problem size (in terms of the number of variables and constraints), computational times (on the same computer), and number of nodes taken to reach zero integrality gap. It is observed that, both the slot-based and global event-based models always require the same number of event points, while the unit-specific event-based models require less event points to solve a problem to global optimality. Thus, the unit-specific event-based models result in smaller problem sizes compared to both slot-based and global event-based models and are computationally superior. In all the examples considered for the objective of maximization of profit, the model of Castro et al.[22] performs better among the slot-based/global event-based models, and it usually requires the smaller number of continuous variables and constraints, while the model of Maravelias and Grossmann[28] generally has the largest number of constraints and nonzeros. For example 2b, the models of Sundaramoorthy and Karimi,[10] and Castro and co-workers[21,22] yield suboptimal solutions. In constrast, for the objective of minimization of makespan, all the slot-based/global event-based models perform weakly in most of the instances of the examples compared to the unit-specific event-based model of Ierapetritou and Floudas.[29] For examples 1a, 1b, and 3b Sundaramoorthy and Karimi,[10] Maravelias and Grossmann,[28] and Castro and co-workers.[21,22] result in suboptimal solutions. The model of Giannelos and Georgiadis[37] yields suboptimal solutions most of the time because of the special sequencing constraints enforced in their model. The unit-specific event-based model of Ierapetritou and Floudas[29] attains the global optimal solution in all examples, and performs the best in terms of both computational performance and problem size. When resource constraints such as utility requirements are considered in the additional two examples it is observed that the unit-specific event-based model of Janak et al.[35] requires 1 less event point and the minimum number of binary variables compared to the global event-based models of Maravelias and Grossmann[28] and Castro et al.[22]

Appendix A: Unit-Specific Event-Based Model of Ierapetritou and Floudas[29] (I&F)

The following is the model used in this paper for the unit-specific event-based formulation of Ierapetritou and Floudas.[29]

For the objective of maximization of profit:

$$\text{Max Profit } = \sum_s price_s \left(ST(s,N) + \sum_{i \in \rho_{si}>0} \rho_{si} \sum_{j \in suit_{ij}} b(i,j,N) \right) \tag{A.1}$$

$$\sum_{i \in suit_{ij}} w(i,j,n) \le 1 \qquad \forall j,n \tag{A.2}$$

$$w(i,j,n)B_{ij}^{\min} \le b(i,j,n) \le w(i,j,n)B_{ij}^{\max} \qquad \forall i,j \in suit_{ij}, \forall n \tag{A.3}$$

$$ST(s,n) = ST(s,n-1) + \sum_{i \in \rho_{si}>0} \rho_{si} \sum_{j \in suit_{ij}} b(i,j,n-1) + \sum_{i \in \rho_{si}<0} \rho_{si} \sum_{j \in suit_{ij}} b(i,j,n) \qquad \forall s,n \tag{A.4}$$

$$ts(i,j,n+1) \ge ts(i',j,n) + \alpha_{i'j} w(i',j,n) + \beta_{i'j} b(i',j,n) \qquad \forall i,i',j \in suit_{ij}, suit_{i'j}, \forall n < N \tag{A.5}$$

$$ts(i,j,n+1) \ge ts(i',j',n) + \alpha_{i'j'} w(i',j',n) + \beta_{i'j'} b(i',j',n) - H(1 - w(i',j',n))$$
$$\forall s,i,i',j,j' \in suit_{ij}, suit_{i'j'}, i \ne i', j \ne j', \rho_{si} < 0, \rho_{si'} > 0, \forall n < N \tag{A.6}$$

$$ts(i,j,N) + \alpha_{ij} w(i,j,N) + \beta_{ij} b(i,j,N) \le H \qquad \forall i,j \in suit_{ij} \tag{A.7}$$

$$ts(i,j,n) \le H \qquad \forall i,j \in suit_{ij}, \forall n \tag{A.8}$$

$$ST(s,n) \le ST_s^{\max} \qquad \forall s \in FIS, \forall n \tag{A.9}$$

$$w(i,j,n) = b(i,j,n) = ts(i,j,n) = 0 \qquad \forall i,j \in suit_{ij} = 0 \tag{A.10}$$

For the objective of minimization of makespan:

$$Min \quad MS \tag{A.11}$$

$$\sum_s \left(ST(s,N) + \sum_{i \in \rho_{si}>0} \rho_{si} \sum_{j \in suit_{ij}} b(i,j,N) \right) \ge Demand_s \tag{A.12}$$

$$ts(i,j,N) + \alpha_{ij} w(i,j,N) + \beta_{ij} b(i,j,N) \le MS \qquad \forall i,j \in suit_{ij} \tag{A.13}$$

The model for makespan minimization is composed of constraints A.2-A.6 and A.9-A.13. The original model of Ierapetritou and Floudas[29] is slightly modified here with some of the dependent variables being eliminated, and the constraints for the same task in the same unit and different tasks in the same unit are combined into one equation in eq. A.5. Also, in contrast to the original model, the only big-M constraints are in constraint A.6. This led to improved LP relaxations in some of the examples. If the problem involves sequence-dependent changeovers, then the constraint A.5 will also have big-M terms. Additionally, tasks that cannot occur at certain events are identified and the corresponding variables are fixed to zero in our implementation.

Appendix B: Global Event-Based Models of Castro and co-workers [21, 22] (CBM, CBMN)

The following is the model used in this paper for the global event-based formulation of Castro et al.[21] (CBM).

For the objective of maximization of profit:

$$\text{Max Profit} = \sum_r price_r R(r, t = |T|) \tag{B.1}$$

$$T(t') - T(t) \geq \alpha_i \overline{N}(i,t,t') + \beta_i \overline{\xi}(i,t,t') \qquad \forall i,t,t' \in t' > t \tag{B.2}$$

$$N(i,t) = \sum_{t'>t} \overline{N}(i,t,t') \qquad \forall i,t \in t < |T| \tag{B.3}$$

$$\xi(i,t) = \sum_{t'>t} \overline{\xi}(i,t,t') \qquad \forall i,t \in t < |T| \tag{B.4}$$

$$V_i^{\min} N(i,t) \leq \xi(i,t) \leq V_i^{\max} N(i,t) \qquad \forall i,t \in t < |T| \tag{B.5}$$

$$V_i^{\min} \overline{N}(i,t,t') \leq \overline{\xi}(i,t,t') \leq V_i^{\max} \overline{N}(i,t,t') \qquad \forall i,t,t' \in t' > t, t < |T| \tag{B.6}$$

$$R(r,t) = R_r^0\big|_{t=1} + R(r,t-1)\big|_{t>1} + \sum_i \left(\mu_{ri} N(i,t) + \nu_{ri}\xi(i,t)\right) + \sum_i \sum_{t'<t} \left(\overline{\mu_{ri}}\,\overline{N}(i,t',t) + \overline{\nu_{ri}}\,\overline{\xi}(i,t',t)\right) \quad \forall r,t \tag{B.7}$$

$$R_r^{\min} \leq R(r,t) \leq R_i^{\max} \qquad \forall r,t \tag{B.8}$$

$$T(t) \leq H \qquad \forall t \tag{B.9}$$

$$T(t) = 0 \qquad \forall t = 1 \tag{B.10}$$

$$\overline{N}(i,t,t') = \overline{\xi}(i,t,t') = 0 \qquad \forall t' = 1 \;\; or \;\; t' \leq t \tag{B.11}$$

$$N(i,t) = \xi(i,t) = \overline{N}(i,t,t') = \overline{\xi}(i,t,t') = 0 \qquad \forall t = |T| \tag{B.12}$$

For the objective of minimization of makespan:

$$Min \quad MS \tag{B.13}$$

$$R(r,t) \geq Demand_r \qquad t = |T| \tag{B.14}$$

$$T(t) \leq MS \qquad t = |T| \tag{B.15}$$

The model for makespan minimization is composed of constraints B.2-B.8 and B.10-B.15.

The following is the model used in this paper for the global event-based formulation of Castro et al.[22] (CBMN). The variables $N(i,t)$ and $\xi(i,t)$ are eliminated from the model of Castro et al.,[21] and for each event we have an additional iteration over a parameter Δt.

For the objective of maximization of profit:

$$\text{Max Profit} = \sum_r price_r R(r, t = |T|) \tag{B.16}$$

$$T(t') - T(t) \geq \sum_i \overline{\mu_{ri}} \left(\alpha_i \overline{N}(i,t,t') + \beta_i \overline{\xi}(i,t,t')\right) \quad \forall r \in R^{EQ}, t, t', t < t' \leq \Delta t + t, t \neq |T| \tag{B.17}$$

$$V_i^{\min} \overline{N}(i,t,t') \leq \overline{\xi}(i,t,t') \leq V_i^{\max} \overline{N}(i,t,t') \quad \forall i,t,t', t < t' \leq \Delta t + t, t \neq |T| \tag{B.18}$$

$$R(r,t) = R_r^0\Big|_{t=1} + R(r,t-1)\Big|_{t>1} + \sum_i \sum_{t<t'\le \Delta t+t} \left(\mu_{ri}\overline{N}(i,t,t') + \nu_{ri}\overline{\xi}(i,t,t') \right) + \sum_i \sum_{t-\Delta t \le t' < t} \left(\overline{\overline{\mu_{ri}}\overline{N}}(i,t',t) + \overline{\overline{\nu_{ri}}\overline{\xi}}(i,t',t) \right) \quad \forall r,t \tag{B.19}$$

$$R_r^{\min} \le R(r,t) \le R_i^{\max} \qquad \forall r,t \tag{B.20}$$

$$T(t) \le H \qquad \forall t \tag{B.21}$$

$$T(t) = 0 \qquad \forall t = 1 \tag{B.22}$$

$$\overline{N}(i,t,t') = \overline{\xi}(i,t,t') = 0 \qquad \forall t' = 1 \;\; or \;\; t' \le t \;\; or \;\; \forall t = |T| \tag{B.23}$$

The model for makespan minimization is composed of constraints B.13-B.15, B.17-B.20, and B.22-B.23.

For zero-wait tasks, the following constraint needs to be added:

$$T(t') - T(t) \le H\left(1 - \sum_{i \in I^{ZW}} \overline{\overline{\mu_{ri}}\overline{N}}(i,t,t') \right) + \sum_{i \in I^{ZW}} \overline{\mu_{ri}} \left(\alpha_i \overline{N}(i,t,t') + \beta_i \overline{\xi}(i,t,t') \right) \quad \forall r \in R^{EQ}, t, t', t < t' \le \Delta t + t, t \ne |T| \tag{B.24}$$

Appendix C: Unit-Specific Event-Based Model of Giannelos and Georgiadis[37] (G&G)

The following is the model used in this paper for the unit-specific event-based formulation of Giannelos and Georgiadis.[37]

For the objective of maximization of profit:

$$\text{Max profit} = \sum_s price_s STF(s) \tag{C.1}$$

$$\sum_{i \in suit_{ij}} x(i,n) \le 1 \qquad \forall j,n \tag{C.2}$$

$$x(i,n)B_i^{\min} \le b(i,n) \le x(i,n)B_i^{\max} \qquad \forall i,n \tag{C.3}$$

$$ST(s,n) = ST(s,n-1) + \sum_{i \in \rho_{si}>0} \rho_{si} b(i,n-1) + \sum_{i \in \rho_{si}<0} \rho_{si} b(i,n) \qquad \forall s,n \tag{C.4}$$

$$STF(s) = ST(s,N) + \sum_{i \in \rho_{si}>0} \rho_{si} b(i,N) \qquad \forall s \tag{C.5}$$

$$\tau(i,n) \ge \tau(i,n-1) + \theta(i,n) + \alpha_i x(i,n) + \beta_i b(i,n) \qquad \forall i,n \tag{C.6}$$

$$\tau(i,n) = \tau(i',n) \qquad \forall s,i,i',n \in i \ne i', \rho_{si} > 0, \rho_{si'} > 0, i = HEAD(I_s^p) \tag{C.7}$$

$$\tau(i,n) - \theta(i,n) - (\alpha_i x(i,n) + \beta_i b(i,n)) = \tau(i',n) - \theta(i',n) - (\alpha_i x(i,n) + \beta_i b(i,n)) \quad \forall s,i,i',n \in i \ne i', \rho_{si} < 0, \rho_{si'} < 0, i = HEAD(I_s^c) \tag{C.8}$$

$$\tau(i,n-1) = \tau(i',n) - \theta(i',n) - (\alpha_{i'} x(i',n) + \beta_{i'} b(i',n) \quad \forall s,i,i',n \in i = HEAD(I_s^p), i' = HEAD(I_s^c) \tag{C.9}$$

$$\tau(i,n) = \tau(i',n) \qquad \forall j,i,i',n \in i \ne i', suit_{ij}, suit_{i'j}, i = HEAD(j) \tag{C.10}$$

$$\tau(i,N) \le H \qquad \forall i \tag{C.11}$$

$$ST(s,n) \le ST_s^{\max} \qquad \forall s \in FIS, \forall n \tag{C.12}$$

For the objective of minimization of makespan:

$$Min \quad MS \tag{C.13}$$

$$STF(s) \geq Demand_s \qquad \forall s \tag{C.14}$$

$$\tau(i,N) \leq MS \qquad \forall i \tag{C.15}$$

The model for makespan minimization consists of constraints C.2-C.10 and C.12-C.15.

Appendix D: Global Event-Based Model of Maravelias and Grossmann[28] (M&G)

The following is the model used in this paper for the global event-based formulation of Maravelias and Grossmann.[28]

For the objective of maximization of profit:

$$\text{Max Profit} = \sum_s price_s ST(s,N) \tag{D.1}$$

$$\sum_{i \in suit_{ij}} Ws(i,n) \leq 1 \qquad \forall j,n \tag{D.2}$$

$$\sum_{i \in suit_{ij}} Wf(i,n) \leq 1 \qquad \forall j,n \tag{D.3}$$

$$\sum_n Ws(i,n) = \sum_n Wf(i,n) \qquad \forall i \tag{D.4}$$

$$\sum_{i \in suit_{ij}} \sum_{n' \leq n} (Ws(i,n') - Wf(i,n')) \leq 1 \qquad \forall j,n \tag{D.5}$$

$$D(i,n) = \alpha_i Ws(i,n) + \beta_i Bs(i,n) \qquad \forall i,n \tag{D.6}$$

$$Tf(i,n) \leq Ts(i,n) + D(i,n) + H(1 - Ws(i,n)) \qquad \forall i,n \tag{D.7}$$

$$Tf(i,n) \geq Ts(i,n) + D(i,n) - H(1 - Ws(i,n)) \qquad \forall i,n \tag{D.8}$$

$$Tf(i,n) - Tf(i,n-1) \leq H \ Ws(i,n) \qquad \forall i, n > 1 \tag{D.9}$$

$$Tf(i,n) - Tf(i,n-1) \geq D(i,n) \qquad \forall i, n > 1 \tag{D.10}$$

$$Ts(i,n) = T(n) \qquad \forall i,n \tag{D.11}$$

$$Tf(i,n-1) \leq T(n) + H(1 - Wf(i,n)) \qquad \forall i, n > 1 \tag{D.12}$$

$$Ws(i,n) B_i^{\min} \leq Bs(i,n) \leq Ws(i,n) B_i^{\max} \qquad \forall i,n \tag{D.13}$$

$$Wf(i,n) B_i^{\min} \leq Bf(i,n) \leq Wf(i,n) B_i^{\max} \qquad \forall i,n \tag{D.14}$$

$$B_i^{\min} \left(\sum_{n' < n} Ws(i,n') - \sum_{n' \leq n} Wf(i,n') \right) \leq Bp(i,n) \leq B_i^{\max} \left(\sum_{n' < n} Ws(i,n') - \sum_{n' \leq n} Wf(i,n') \right) \quad \forall i,n \tag{D.15}$$

$$Bs(i,n-1) + Bp(i,n-1) = Bp(i,n) + Bf(i,n) \qquad \forall i, n > 1 \tag{D.16}$$

$$B^I(i,s,n) = \rho_{si} Bs(i,n) \qquad \forall i,n, \forall s \in SI(i) \tag{D.17}$$

$$B^I(i,s,n) \leq B_i^{max} \rho_{si} Ws(i,n) \qquad \forall i,n, \forall s \in SI(i) \tag{D.18}$$

$$B^O(i,s,n) = \rho_{si} Bf(i,n) \qquad \forall i,n,\forall s \in SO(i) \tag{D.19}$$

$$B^O(i,s,n) \le B_i^{max} \rho_{si} Wf(i,n) \qquad \forall i,n,\forall s \in SO(i) \tag{D.20}$$

$$ST(s,n) = ST(s,n-1) + \sum_{i \in O(s)} B^O(i,s,n) - \sum_{i \in I(s)} B^I(i,s,n) \quad \forall s,n>1 \tag{D.21}$$

$$T(n+1) \ge T(n) \qquad \forall n < N \tag{D.22}$$

$$\sum_{i \in suit_{ij}} \sum_{n} D(i,n) \le H \qquad \forall j \tag{D.23}$$

$$\sum_{i \in suit_{ij}} \sum_{n' \ge n} D(i,n') \le H - T(n) \qquad \forall j,n \tag{D.24}$$

$$\sum_{i \in suit_{ij}} \sum_{n' \le n} \left(\alpha_i Wf(i,n') + \beta_i Bf(i,n') \right) \le T(n) \qquad \forall j,n \tag{D.25}$$

$$Ts(i,n) \le H \qquad \forall i,n \tag{D.26}$$

$$Tf(i,n) \le H \qquad \forall i,n \tag{D.27}$$

$$ST(s,n) \le ST_s^{\max} \qquad \forall s \in FIS, \forall n \tag{D.28}$$

$$T(n) = Wf(i,n) = Bf(i,n) = B^O(i,s,n) = 0 \qquad \forall n = 1 \tag{D.29}$$

$$Ws(i,n) = Bs(i,n) = D(i,n) = Bp(i,n) = B^I(i,s,n) = 0 \qquad \forall n = N \tag{D.30}$$

$$T(N) = H \tag{D.31}$$

For the objective of minimization of makespan:

$$Min \quad MS \tag{D.32}$$

$$ST(s,N) \ge Demand_s \qquad \forall s \tag{D.33}$$

$$T(N) = MS \tag{D.34}$$

$$\sum_{i \in suit_{ij}} \sum_{n} D(i,n) \le MS \qquad \forall j \tag{D.35}$$

$$\sum_{i \in suit_{ij}} \sum_{n' \ge n} D(i,n') \le MS - T(n) \qquad \forall j,n \tag{D.36}$$

The model for makespan minimization uses constraints D.2-D.22, D.25-D.30, and D.32-D.36.

For zero-wait tasks, the following constraints are added:

$$Tf(i,n-1) \ge T(n) - H(1 - Wf(i,n)) \qquad \forall i \in I^{ZW}, n > 1 \tag{D.37}$$

When utility requirements are considered, the following constraints are added:

$$R^I(i,r,n) = \gamma_{ir} Ws(i,n) + \delta_{ir} Bs(i,n) \qquad \forall i,r,n \tag{D.38}$$

$$R^O(i,r,n) = \gamma_{ir} Wf(i,n) + \delta_{ir} Bf(i,n) \qquad \forall i,r,n \tag{D.39}$$

$$R(r,n) = R(r,n-1) - \sum_i R^O(i,r,n-1) + \sum_i R^I(i,r,n) \qquad \forall r,n \tag{D.40}$$

$$R(r,n) \le R_r^{\max} \qquad \forall r,n \tag{D.41}$$

Appendix E: Slot-Based Model of Sundaramoorthy and Karimi[10] (S&K)

The following is the model used in this paper for the slot-based formulation of Sundaramoorthy and Karimi.[10] Here, the set of tasks (I) also includes an idle task '*i*0'that is suitable on all units.

For the objective of maximization of profit:

$$\text{Max Profit} = \sum_s price_s ST(s,K) \quad \text{(E.1)}$$

$$\sum_k SL(k) \le H \quad \text{(E.2)}$$

$$Z(j,k) = \sum_{i \in suit_{ij}} Y(i,j,k) \qquad \forall j, 0 \le k < K \quad \text{(E.3)}$$

$$Y(i,j,k)B_{ij}^{\min} \le B(i,j,k) \le Y(i,j,k)B_{ij}^{\max} \quad \forall i > 0, j \in suit_{ij}, 0 \le k < K \quad \text{(E.4)}$$

$$y(i,j,k) = y(i,j,k-1) + Y(i,j,k-1) - YE(i,j,k) \quad \forall i, j \in suit_{ij}, 0 < k < K \quad \text{(E.5)}$$

$$Z(j,k) = \sum_{i \in suit_{ij}} YE(i,j,k) \qquad \forall j, 0 < k < K \quad \text{(E.6)}$$

$$t(j,k+1) \ge t(j,k) + \sum_{i \in suit_{ij}} \left(\alpha_{ij} Y(i,j,k) + \beta_{ij} B(i,j,k) \right) - SL(k+1) \qquad \forall j, k < K \quad \text{(E.7)}$$

$$b(i,j,k) = b(i,j,k-1) + B(i,j,k-1) - BE(i,j,k) \quad \forall i > 0, j \in suit_{ij}, k > 0 \quad \text{(E.8)}$$

$$y(i,j,k)B_{ij}^{\min} \le b(i,j,k) \le y(i,j,k)B_{ij}^{\max} \quad \forall i > 0, j \in suit_{ij}, 0 < k < K \quad \text{(E.9)}$$

$$YE(i,j,k)B_{ij}^{\min} \le BE(i,j,k) \le YE(i,j,k)B_{ij}^{\max} \quad \forall i > 0, j \in suit_{ij}, 0 < k \le K \quad \text{(E.10)}$$

$$t(j,k) \le \sum_{i \in suit_{ij}} \alpha_{ij} y(i,j,k) + \beta_{ij} b(i,j,k) \qquad \forall j, 0 < k < K \quad \text{(E.11)}$$

$$ST(s,k) = ST(s,k-1) + \sum_j \sum_{i \in suit_{ij}, i \ne 0, \rho_{si} > 0} \rho_{si} BE(i,j,k) + \sum_j \sum_{i \in suit_{ij}, i \ne 0, \rho_{si} < 0} \rho_{si} B(i,j,k) \quad \forall s,k \quad \text{(E.12)}$$

$$ST(s,k) \le ST_s^{\max} \qquad \forall s \in FIS, \forall k \quad \text{(E.13)}$$

$$SL(k) \le \max_j \left[\max_{i \in suit_{ij}} (\alpha_{ij} + \beta_{ij} B_{ij}^{\max}) \right] \qquad \forall k > 0 \quad \text{(E.14)}$$

$$t(j,k) \le \max_{i \in suit_{ij}} (\alpha_{ij} + \beta_{ij} B_{ij}^{\max}) \qquad \forall j,k \quad \text{(E.15)}$$

$$Y(i,j,k) = y(i,j,k) = b(i,j,k) = B(i,j,k) = 0 \qquad \forall i, j \in suit_{ij} = 0 \text{ or } k = K \quad \text{(E.16)}$$

$$YE(i,j,k) = y(i,j,k) = b(i,j,k) = BE(i,j,k) = 0 \qquad \forall i, j \in suit_{ij} = 0 \text{ or } k = 0 \quad \text{(E.17)}$$

$$Z(j,k) = t(j,k) = 0 \qquad \forall j, k = K \quad \text{(E.18)}$$

$$t(j,k) = 0; SL(k) = 0 \qquad \forall k = 0 \quad \text{(E.19)}$$

$$0 \le y(i,j,k), YE(i,j,k), Z(j,k) \le 1 \quad \text{(E.20)}$$

For the objective of minimization of makespan:

$$Min \quad MS = \sum_{k=1}^{K} SL(k) \quad \text{(E.21)}$$

$$ST(s,K) \ge Demand_s \qquad \forall s \quad \text{(E.22)}$$

The model for makespan minimization consists of constraints E.3-E.22. The constraints E.9 and E.10 are misprinted in the original paper (constraints 11 and 12 of Sundaramoorthy and Karimi[10]), in which they were written as follows:

$$B_{ij}^{\min} \le b(i,j,k) \le B_{ij}^{\max} - y(i,j,k) \quad \forall i>0, j \in suit_{ij}, 0<k<K \qquad \text{(E.23)}$$

$$YE(i,j,k)B_{ij}^{\min} \le BE(i,j,k) \le B_{ij}^{\max} - YE(i,j,k) \quad \forall i>0, j \in suit_{ij}, 0<k\le K \qquad \text{(E.24)}$$

Acknowledgements: The authors gratefully acknowledge support from the National Science Foundation and BASF Aktiengesellschaft, Ludwigshafen, Germany.

References

(1) Floudas, C. A.; Lin, X. Continuous-time versus discrete-time approaches for scheduling of chemical processes: A review. *Comput. Chem. Eng.* **2004**, *28*, 2109.

(2) Floudas, C. A.; Lin, X. Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications. *Ann. Oper. Res.* **2005**, *139*, 131.

(3) Pinto, J. M.; Grossmann, I. E. Optimal cyclic scheduling of multistage continuous multiproduct plants. *Comput. Chem. Eng.* **1994**, *18*, 797.

(4) Pinto, J. M.; Grossmann, I. E. A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants. *Ind. Eng. Chem. Res.* **1995**, *34*, 3037.

(5) Pinto, J. M.; Grossmann, I. E. An alternate MILP model for short-term scheduling of batch plants with preordering constraints. *Ind. Eng. Chem. Res.* **1996**, *35*, 338.

(6) Pinto, J. M.; Grossmann, I. E. A logic-based approach to scheduling problems with resource constraints. *Comput. Chem. Eng.* **1997**, *21*, 801.

(7) Karimi, I. A.; McDonald, C. M. Planning and scheduling of parallel semi-continuous processes. 2. Short-term scheduling. *Ind. Eng. Chem. Res.* **1997**, *36*, 2701.

(8) Lamba, N.; Karimi, I. A. Scheduling parallel production lines with resource constraints. 1. Model formulation. *Ind. Eng. Chem. Res.* **2002**, *41*, 779.

(9) Lamba, N.; Karimi, I. A. Scheduling parallel production lines with resource constraints. 2. Decomposition algorithm. *Ind. Eng. Chem. Res.* **2002**, *41*, 790.

(10) Sundaramoorthy, A.; Karimi, I. A. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. *Chem. Eng. Sci.* **2005**, *60*, 2679.

(11) Zhang, X.; Sargent, R. W. H. The optimal operation of mixed production facilities - A general formulation and some approaches for the solution. *Comput. Chem. Eng.* **1996**, *20*, 897.

(12) Zhang, X.; Sargent, R. W. H. The optimal operation of mixed production facilities - Extensions and improvements. *Comput. Chem. Eng.* **1998**, *22*, 1287.

(13) Mockus, L.; Reklaitis, G. V. Mathematical programming formulation for scheduling of batch operations based on nonuniform time discretization. *Comput. Chem. Eng.* **1997**, *21*, 1147.

(14) Mockus, L.; Reklaitis, G. V. Continuous time representation approach to batch and continuous process scheduling. 1. MINLP formulation. *Ind. Eng. Chem. Res.* **1999**, *38*, 197.

(15) Mockus, L.; Reklaitis, G. V. Continuous time representation approach to batch and continuous process scheduling. 2. Computational issues. *Ind. Eng. Chem. Res.* **1999**, *38*, 204.

(16) Schilling, G.; Pantelides, C. C. A simple continuous-time process scheduling formulation and a novel solution algorithm. *Comput. Chem. Eng.* **1996**, *20*, S1221.

(17) Schilling, G.; Pantelides, C. C. Optimal periodic scheduling of multipurpose plants. *Comput. Chem. Eng.* **1999**, *23*, 635.

(18) Mendez, C. A.; Henning, G. P.; Cerda, J. Optimal scheduling of batch plants satisfying multiple product orders with different due-dates. *Comput. Chem. Eng.* **2000**, *24*, 2223.

(19) Mendez, C. A.; Cerda, J. An MILP continuous-time framework for short-term scheduling of multipurpose batch processes under different operation strategies. *Optim. Eng.* **2003**, *4*, 7.

(20) Mendez, C. A.; Henning, G. P.; Cerda, J. An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities. *Comput. Chem. Eng.* **2001**, *25*, 701.

(21) Castro, P.; Barbosa-Povoa, A. P. F. D.; Matos, H. An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants. *Ind. Eng. Chem. Res.* **2001**, *40*, 2059.

(22) Castro, P. M.; Barbosa-Povoa, A. P.; Matos, H. A.; Novais, A. Q. Simple continuous-time formulation for short-term scheduling of batch and continuous processes. *Ind. Eng. Chem. Res.,* **2004**, *43*, 105.

(23) Mendez, C. A.; Cerda, J. Optimal scheduling of a resource-constrained multiproduct batch plant supplying intermediates to nearby end-product facilities. *Comput. Chem. Eng.* **2000**, *24*, 369.

(24) Majozi, T.; Zhu, X. X. Novel continuous-time MILP formulation for multipurpose batch plants. 1. Short-term scheduling. *Ind. Eng. Chem. Res.* **2001**, *40*, 5935.

(25) Lee, K.; Park, H. I.; Lee, I. A novel nonuniform discrete time formulation for short-term scheduling of batch and continuous processes. *Ind. Eng. Chem. Res.* **2001**, *40*, 4902.

(26) Burkard, R. E.; Fortuna, T.; Hurkens, C. A. J. Makespan minimization for chemical batch processes using nonuniform time grids. *Comput. Chem. Eng.* **2002**, *26*, 1321.

(27) Wang, S.; Guignard, M. Redefining event variables for efficient modeling of continuous-time batch processing. *Ann. Oper. Res.* **2002**, *116*, 113.

(28) Maravelias, C. T.; Grossmann, I. E. New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants. *Ind. Eng. Chem. Res.* **2003**, *42*, 3056.

(29) Ierapetritou, M. G.; Floudas, C. A. Effective continuous-time formulation for short-term scheduling: 1. Multipurpose batch processes. *Ind. Eng. Chem. Res.* **1998**, *37*, 4341.

(30) Ierapetritou, M.G.; Floudas, C.A. Effective continuous-time formulation for short-term scheduling: 2. Continuous and semi-continuous processes. *Ind. Eng. Chem. Res.* **1998**, *37*, 4360.

(31) Ierapetritou, M. G.; Floudas, C. A. Comments on “An Improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants.” *Ind. Eng. Chem. Res.* **2001**, *40*, 5040.

(32) Ierapetritou, M. G.; Hene, T. S.; Floudas, C. A. Effective continuous-time formulation for short-term scheduling: 3. Multiple intermediate due dates. *Ind. Eng. Chem. Res.* **1999**, *38*, 3446.

(33) Lin, X.; Floudas, C. A. Design, synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation. *Comput. Chem. Eng.* **2001**, *25*, 665.

(34) Lin, X.; Chajakis, E. D.; Floudas, C. A. Scheduling of tanker lightering via a novel continuous-time optimization framework. *Ind. Eng. Chem. Res.* **2003**, *42*, 4441.

(35) Janak, S. L.; Lin, X.; Floudas, C. A. 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.* **2004**, *42*, 2516.

(36) Janak, S. L.; Lin, X.; Floudas, C. A. Comments on “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.* **2005**, *44*, 426.

(37) Giannelos, N. F.; Georgiadis, M. C. A simple new continuous-time formulation for short-term scheduling of multipurpose batch processes. *Ind. Eng. Chem. Res.* **2002**, *41*, 2178.

(38) Maravelias, C. T.; Grossmann, I. E. A hybrid MILP/CP decomposition approach for the continuous time scheduling of multipurpose batch plants. *Comput. Chem. Eng.* **2004**, *28*, 1921.

(39) Brooke, A.; Kendrick, D.; Meeraus, A.; Raman, R. *GAMS: A user's guide*; GAMS Development Corporation: South San Francisco, CA, 2003.