IMPROVED TIGHTENED MILP FORMULATIONS FOR SINGLE-STAGE BATCH SCHEDULING PROBLEMS

Pablo A. Marchetti and Jaime Cerdá*

INTEC (UNL - CONICET)
Güemes 3450 - 3000 Santa Fe - ARGENTINA
** E-mail: jcerda@intec.unl.edu.ar*

Abstract: This work presents a set of improved MILP mathematical formulations for the scheduling of single-stage batch plants with parallel production lines. Minimization of the average weighted earliness and the makespan, i.e. the time needed to complete all processing tasks, are considered as alternative problem goals. For each objective function an enhanced model that incorporates specific tightening constraints is presented. These constraints improve each model's efficiency by increasing the corresponding objective function lower bound, thus accelerating the branch and bound node pruning process. Several problem instances with different number of batches demonstrate that the proposed approach reduces the computational effort by orders of magnitude. Sequence dependent setup times can also be effectively accommodated.

Keywords: Scheduling algorithms, Optimization, Manufacturing processes, Computer-aided engineering, Integer programming

1. INTRODUCTION

The problem of finding an efficient short-term schedule for a multiproduct batch plant is of major interest for most manufacturing companies. Several solution methodologies have been proposed for different kinds of scheduling problems. An extensive review of the state of the art can be found in Floudas and Lin (2004). Overall, exact solution methods have received most of the researchers attention. Among the different MILP models proposed in the literature, continuous-time models have shown a better computational performance for the scheduling of batch processes with sequence-dependent changeovers. In particular, the continuous time batch scheduling problem model of Méndez *et. al.* (2001) shows a good computational behaviour compared with other continuous approaches, like the time-slot formulation of Pinto and Grossmann (1995) or the unit-specific time event model of Ierapetritou and Floudas (1998). However, its performance is somewhat deteriorated by the big-M batch sequencing constraints, especially when the makespan objective function is considered. Big-M constraints produce an increase in the integrality gap (the difference between optimal values for the relaxed and MILP problems), which in turn makes the optimal solution harder to find by the MILP solver through a branch-and-bound algorithm.

This work presents a pair of improved formulations for single-stage batch plant scheduling problems that incorporate tightening constraints in order to reduce the integrality gap and enhance the branch-and-bound node pruning process. Based on the MILP approach by Méndez *et. al.* (2001), the proposed models account for release times, ready times, due dates, and sequence dependent setup times between batches. Minimization of the average weighted earliness and the makespan, i.e. the time needed to execute all processing tasks, are both considered as problem objectives. Specific tightening constraints are presented in order to improve the lower bound on each objective function, leading to different formulations for each problem.

This work is organized as follows. In section 2 the scheduling problem under consideration is properly defined. In section 3 the different mathematical models are presented. At first, the single-stage version of the MILP model by Méndez *et. al.* (2001) is reviewed. Common constraints related to assignment and sequencing decisions are included. Afterwards, specific constraints and binary variables that improve the model's efficiency for each objective function are introduced. These additional constraints are incorporated or just replace previous ones, and their main purpose is to increase the objective function lower bound. Section 4 shows the effectiveness of the proposed approach by tackling several problem instances with an increasing number of batches and different sequencing conditions for both objective functions. Conclusions are discussed in section 5.

2. PROBLEM STATEMENT

The problem of short-term scheduling of single-stage multiproduct batch plants with parallel production lines can be stated as follows. Given: (a) a single-stage multiproduct batch plant with multiple parallel units $j \in J$, (b) a set of single-batch orders $i \in I$ to be completed within the scheduling horizon, (c) the order release times rt_i and due dates dd_i for each $i \in I$, (d) the set of available processing units $J_i \subset J$ for each batch i, and the constant processing times pt_{ij} required at each unit $j \in J_i$, (e) the sequence-dependent setup times $\tau_{ii'j}$, (f) the equipment unit ready times ru_j, and (g) the specified time horizon H. The problem goal is to find a production schedule that completes all batch orders within their time limits, meeting assignment and sequencing constraints and optimizing a given schedule criterion, like the overage weighted earliness or the makespan.

3. IMPROVED MATHEMATICAL FORMULATIONS

3.1 The MILP approach of Méndez et. al. (2001)

PROBLEM CONSTRAINTS:

Assignment of batches to processing units. A single equipment item should be allocated to every batch. The binary variable Y_{ij} stands for the decision of allocating batch i to unit j.

$$\sum_{j \in J_i} Y_{ij} = 1 \qquad \forall i \in I \qquad (1)$$

Batch sequencing. If two batches $i, i' \in I$ can be assigned to the same processing unit $j \in J_i \cap J_{i'}$, then sequencing constraints dealing with setup or changeover times that prevent from task overlapping should be included. These sequencing decisions are handled through fewer binary variables by using the general precedence concept. Let the binary variable $X_{ii'}$ stand for the relative ordering of batches $i,i' \in I$, where $i<i'$, if both batches are assigned to the same unit j ($Y_{ij} = Y_{i'j} = 1$). Specifically, $X_{ii'} = 1$ if batch i is processed before batch i' in the common equipment unit j, or $X_{ii'} = 0$ if batch i takes place afterwards. Notice that this binary variable becomes meaningless if i and i' are assigned to different equipment units.

$$C_i + \tau_{ii'j} + su_{i'j} \le S_{i'} + H\left(1 - X_{ii'}\right) + H\left(2 - Y_{ij} - Y_{i'j}\right)$$
$$\forall i,i' \in I,\ j \in J_{ii'}:\ i < i' \qquad (2)$$

$$C_{i'} + \tau_{i'ij} + su_{ij} \le S_i + H\, X_{ii'} + H\left(2 - Y_{ij} - Y_{i'j}\right)$$
$$\forall i,i' \in I,\ j \in J_{ii'}:\ i < i' \qquad (3)$$

where: $J_{ii'} = J_i \cap J_{i'}$

Timing of batches. The starting time of batch i can be computed from its completion time by subtracting its processing time in the assigned unit.

$$S_i = C_i - \sum_{j \in J_i} pt_{ij} Y_{ij} \qquad \forall i \in I \qquad (4)$$

In addition, the starting time of a batch must be higher than either its release time or the sum of both the unit ready time and the batch setup time in the assigned equipment unit.

$$S_i \ge \sum_{j \in J_i} Max\left[rt_i,\ ru_j + su_{ij}\right] Y_{ij} \qquad \forall i \in I \qquad (5)$$

OBJECTIVE FUNCTIONS:

Makespan. The makespan represents how much time is required to complete all processing tasks. The definition of the makespan variable MK is incorporated in the model by the following inequality constraint:

$$C_i \le MK \qquad \forall i \in I \qquad (6)$$

Thus, the problem goal will be:

$$\text{Minimize } MK \qquad (7)$$

Average weighted earliness. An alternative problem goal is to minimize the average weighted earliness:

$$\text{Minimize } \frac{1}{|I|} \sum_{i \in I} \alpha_i \left(dd_i - C_i\right) \qquad (8)$$

where the parameter α_i stands for the weight of the earliness of batch i. Notice that minimizing the average or the overall weighted earliness are equivalent problems as long as $n = |I|$ is a constant parameter. Since due dates are also constant, this objective function involves maximizing the average completion time of the batches.

The previous objective function will be useful only if the completion time of each batch never exceeds its specified due date.

$$C_i \le dd_i \qquad \forall i \in I \qquad (9)$$

3.2 *Tightening the Makespan lower bound*

A simple analysis of the above equations (2)-(6) leads to the following conclusions: The makespan lower bound is defined as the maximum completion time of any batch by constraint (6). At the same time, constraint (4) defines the completion time of each batch based on its processing time and start time. Therefore, it can be expected that the optimal solution will minimize processing times and start times simultaneously, at least for the last batch completed, since both (4) and (5) depend on the unique processing unit allocated by (1). If the equipment availability is much higher than the batch requirements (i.e. more equipment units than batches to schedule are available), the above constraints will lower bound the objective function and the MILP solver will easily choose the best possible assignment solution.

Unfortunately, this is not the ordinary situation. In general, several batches will be allocated to each processing unit and the sequencing equations (2) and (3) will need to be taken into account. Since these constraints are of big-M type, they do not provide a good lower bound estimation on the objective function. Even if the assignment variables Y_{ij} and $Y_{i'j}$ were set to 1 for a given equipment j, the sequencing variable $X_{ii'}$ can take a fractional value during the branch-and-bound search causing that neither equation (2) nor (3) have any effect on the starting or completion times of the batches.

However, assignment variables partially or completely allocating units to batches, i.e. equal to 1 or a positive fractional value, constitute a valuable information to estimate a tight lower bound for the makespan. Usually, the processing time of a task is frequently larger than its setup time (either sequence dependent or independent). Based on this assumption, the overall workload assigned to each equipment unit can be estimated using the summation of processing times of all the tasks allocated to it. When sequence independent setup times are considered, they can be included in the summation, which happens to be a good estimation for the schedule makespan:

$$ru_j + \sum_{i \in I_j} \left(su_{ij} + pt_{ij} \right) Y_{ij} \leq MK \qquad \forall j \in J \quad (10)$$

Notice that constraint (10) determines a lower bound for the makespan based on the total processing time at each unit, whatever is the sequence of tasks selected. Consequently, the summation term is a valid estimation for the makespan, based only on assignment variables. Sequencing decisions neither appear in this equation nor influence its tightening effect.

If, instead, sequence dependent setup times are significant, they can still be included in the previous equation very easily. The summation in constraint (11) now includes the lowest possible sequence dependent setup for each batch, as defined by equation (12). Since τ_{ij}^{Min} is included for every batch allocated on the processing unit, and the first batch to be processed does not need any prior setup, the highest possible sequence dependent setup is subtracted in order to ensure optimality:

$$ru_j - \underset{i \in I_j}{Max}\left[\tau_{ij}^{Min}\right] + \sum_{i \in I_j} \left(\tau_{ij}^{Min} + su_{ij} + pt_{ij} \right) Y_{ij} \leq MK \qquad \forall j \in J \quad (11)$$

where:

$$\tau_{ij}^{Min} = \underset{i' \in I_j : i' \neq i}{Min}\left[\tau_{i'ij}\right] \qquad \forall i \in I,\, j \in J_i \quad (12)$$

Constraint (11) provides a good lower bound on the value of *MK* because the model will try to optimise the batch sequencing at each unit in order to minimize the setup times. However, only if setup times between batches at the same unit are of similar order of magnitude this inclusion will be useful as a tight estimation. If $\tau_{ij}^{Min} = 0$ for a given batch, no improvement on the lower bound is possible.

Although the above estimations are rather straightforward, they are quite useful to get a good lower bound estimation on the makespan using a formulation that allows sequence dependent setup times. It is desirable to also get simple estimations for other objective functions.

3.3 *Tightening the lower bound on the Average Earliness*

As mentioned before, the objective function defined by (8) maximizes the summation of batch completion times. In turn, constraint (9) gives an upper bound on the completion time of each batch, also defining a preliminary lower bound on the value of this objective function. Neither (4) nor (5) have any influence on it. Since start times are not primarily affected by the objective function, there is no direct model trend to choose any particular equipment unit for a given batch. In the makespan case it was likely to choose an equipment with minimum processing time. Thus, for a given batch $i \in I$ allocated to unit $j \in J_i$, it is clear that its completion time will not be deteriorated unless another batch i' is assigned to the same processing unit. But, as mentioned before, sequencing decisions must be made in order to change start or completion times, and these decisions are defined by constraints (2) and (3), which are of big-M type. Since changes on the objective function value during the branch-and-bound process depend on the relative ordering of the batches allocated to the same processing unit, and because batches are not assumed to be previously assigned to any equipment unit, estimating a lower bound on the average earliness will be a more complex task than before.

Nonetheless, valid estimations of batch earliness can still be inferred. Let us suppose that two processing tasks i and i' are both allocated to the same unit j with batch i preceding i' (i.e., $Y_{ij} = Y_{i'j} = 1$ and $X_{ii'} = 1$ if

$i<i'$). Therefore, let us define the parameter $\beta_{ii'j}$ to estimate the earliness deterioration caused by batch i' over batch i if both are allocated to the same unit j, and batch i is executed before i'. Notice that the latest start time (*LST*) of batch i' will be an upper bound on the completion time of batch i. Figure 1 shows three possible scenarios for the temporal relation between due dates and processing times of both batches. In Figure 1(a), batch i' is due before batch i. Since it was assumed that i precedes i', the completion time of batch i must be deteriorated by at least the sum of the processing time and the setup time of i'. Alternatively, if batch i' is due after batch i, the completion time of batch i will be bounded by the difference shown in Figure 1(b), only if such a difference is positive. Otherwise, the estimation will be null as in Figure 1(c). Therefore, the earliness of batch i is deteriorated for each batch i' executed on unit j after i by the amount $\beta_{ii'j}$. Summing up these individual deteriorations, it is possible to estimate a lower bound on the earliness of each batch i, as will be next shown.

To derive the proposed estimation, a different set of sequencing binary variables must be defined:

$$\Theta_{ii'j} = \begin{cases} 1 & \text{if batch } i \text{ is processed before batch } i' \text{ on unit } j \\ 0 & \text{otherwise} \end{cases} \qquad \forall i, i' \in I,\ j \in J_{ii'}:\ i \neq i'$$

Since these binary variables are defined for every possible permutation of two distinct batches, and for each eligible equipment unit for both batches, an immediate conclusion is that this new approach will significantly increase the number of decision variables. With the model of Méndez *et. al.* (2001), a smaller set of sequencing variables is required because sequencing and assignment decisions were independent. Variable $X_{ii'}$ do not includes index j and is defined for every ordered pair of batches $i, i' \in I$, such that $i<i'$. As the number of variables $\Theta_{ii'j}$ is larger, it can be expected that the computational performance will not improve. However, as it will be shown, this is not true since a better lower bound for the objective function and, consequently, a lower CPU requirement is achieved.

Constraints (13) and (14) replace constraints (2) and (3) for the proposed relationships among sequencing decisions:

$$Y_{ij} + Y_{i'j} \leq 1 + \Theta_{ii'j} + \Theta_{i'ij} \qquad \forall i, i' \in I,\ j \in J_{ii'}: i < i' \qquad (13)$$

$$C_i + \sum_{j \in J_{ii'}} \left(\tau_{ii'j} + su_{i'j}\right) Y_{i'j} \leq S_{i'} + H\left(1 - \sum_{j \in J_{ii'}} \Theta_{ii'j}\right) \qquad \forall i, i' \in I:\ i \neq i' \qquad (14)$$

Constraint (13) produces that either $\Theta_{ii'j}$ or $\Theta_{i'ij}$ are set to 1 if batches i and i' are both allocated to unit j.

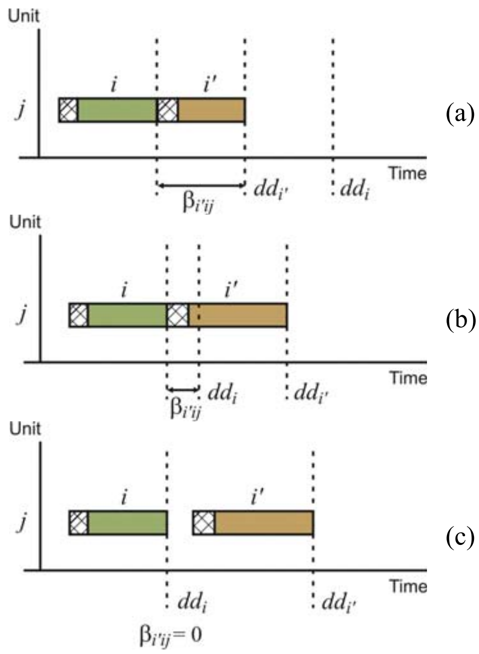

Fig. 1. Earliness deterioration caused by batch i' over batch i if both are allocated to the same unit j.

This constraint relates model's assignment decisions and model's sequencing decisions, which is the main difference with the previous approach. In turn, constraint (14) has the same effect that constraints (2) and (3), but here a summation over the available equipment units reduces the number of constraints by one order of magnitude.

Finally, the proposed constraint to tighten the lower bound on the overall earliness objective function is:

$$C_i + \sum_{i' \in I:\, i' \neq i} \ \sum_{j \in J_{ii'}} \beta_{ii'j}\, \Theta_{ii'j} \leq dd_i \qquad \forall i \in I \qquad (15)$$

Here constraint (15) replaces previous constraint (6), where $\beta_{ii'j}$ is the earliness estimation parameter defined as:

$$\beta_{ii'j} = \begin{cases} \tau_{i'j}^{Min} + su_{i'j} + pt_{i'j} & , \text{if } dd_{i'} \leq dd_i \\ dd_i + \tau_{i'j}^{Min} + su_{i'j} + pt_{i'j} - dd_{i'} & , \text{if } (dd_i < dd_{i'}) \text{ and } (dd_{i'} - \tau_{i'j}^{Min} - su_{i'j} - pt_{i'j} < dd_i) \\ 0 & , \text{otherwise} \end{cases} \qquad (16)$$

Constraint (15) causes that the upper bounds on the completion times are affected by the assignment decisions while batch-unit allocations are made. To reach this conclusion it is necessary to understand how constraints (13) and (15) work together. Constraint (13) assures that, once a batch i is assigned to equipment j ($Y_{ij} = 1$), any other batch i' assigned or partially assigned to the same unit ($0 < Y_{i'j} \leq 1$) will automatically increase binaries variables $\Theta_{ii'j}$ or $\Theta_{i'ij}$. Since these sequencing variables appear on the summation of constraint (15), it is expected that a variable $\Theta_{i'ij}$ related to the ordered pair (i, i') on unit j and featuring a lower $\beta_{ii'j}$ will take a nonzero value. Therefore, the lowest possible deterioration of both completion times will be chosen.

Tightening constraint (15) will have no effect until assignment decisions are at least partially made. If two batches have the same or almost the same due date, it is expected that both $\beta_{ii'j}$ and $\beta_{i'ij}$ will be nonzero. In this case, if i is already allocated to unit j ($Y_{ij} = 1$), the optimal relaxed solution will avoid the assignment of i' to the same unit j, if a deterioration of the earliness of one of the batches will happen. In this way, the model will avoid the assignment of new tasks to a unit if it is overloaded. In general, the lower bound proposed on the value of the objective function is useful during the node pruning process whenever $1 < Y_{ij}+Y_{i'j} \leq 2$, and consequently, i and/or i' are assigned or partially assigned to the same processing unit j.

4. COMPUTATIONAL RESULTS

The effectiveness of the above tightening constraints will be illustrated by finding the minimum-makespan (Example 1) and the minimum-earliness (Example 2) schedules for a single-stage multiproduct batch plant. Both sequence dependent and independent changeovers are considered, since sequence dependent setups have a significant influence on the model performance and they cannot be efficiently considered with other formulations.

The problem to be tackled involves a plastic compounding plant with a single stage and four extruders running in parallel. This problem was first studied by Pinto and Grossmann (1995) and Ierapetritou, Hené, and Floudas (1999) with up to 29 batch orders. Méndez and Cerdá (2003) expanded the number of batches to 40, in order to undertake an appropriate dynamic scheduling scenario. Order due dates and unit-dependent processing and setup times used on this section can be found in Méndez and Cerdá (2003).

Table 1. Product families

Family	Batches
F_1	$O_1, O_2, O_3, O_5, O_{10}, O_{16}, O_{20}, O_{22}$
F_2	$O_4, O_8, O_9, O_{14}, O_{18}, O_{26}, O_{31}$
F_3	$O_7, O_{23}, O_{24}, O_{30}, O_{33}, O_{34}, O_{36}, O_{37}, O_{38}, O_{40}$
F_4	$O_6, O_{11}, O_{15}, O_{17}, O_{19}, O_{32}, O_{35}$
F_5	$O_{12}, O_{13}, O_{21}, O_{25}, O_{27}, O_{28}, O_{29}, O_{39}$

Table 2. Sequence dependent setup times between families

	F_1	F_2	F_3	F_4	F_5
F_1	0.104	0.127	0.178	0.192	0.217
F_2	0.122	0.115	0.266	0.229	0.291
F_3	0.191	0.214	0.175	0.304	0.424
F_4	0.350	0.205	0.328	0.184	0.400
F_5	0.357	0.423	0.348	0.284	0.205

In order to address sequence dependent changeovers, batch orders are grouped into five product families F_1-F_5 as shown in Table 1, and sequence-dependent setup times between families are listed in Table 2. Hence, two versions of Examples 1 and 2 have been studied, one assuming sequence-independent changeovers and the other considering changeovers as sequence dependent. Both versions were solved using the approach of Méndez *et. al.* (2001) and the corresponding improved formulation, for an increasing number of batches ranging form 12 to 25/40, in order to reach the computational limit of each model.

Table 3. Makespan minimization results with the model of Méndez *et. al.* (2001)

n	Binary vars, Continuous vars, Constraints	Sequence independent setup times				Sequence dependent setup times			
		Objective Function	Relative Gap (%)	CPU time (sec.)	Nodes	Objective Function	Relative Gap (%)	CPU time (sec.)	Nodes
12	82, 25, 214	8.428	-	19.03	94365	8.645	-	8.36	39350
16	140, 33, 382	12.353	2.43	>3600	8893218	12.854	-	1188.50	3421982
18	161, 37, 444	13.985	-	2872.81	7166701	14.633	27.07	>3600	8708577
20	201, 41, 558	15.268	22.62	>3600	6282059	15.998	21.95	>3600	6570231

Table 4. Makespan minimization results with the proposed model

n	Binary vars, Continuous vars, Constraints	Sequence independent setup times				Sequence dependent setup times			
		Objective Function	Relative Gap (%)	CPU time (sec.)	Nodes	Objective Function	Relative Gap (%)	CPU time (sec.)	Nodes
12	82, 25, 218	8.428	-	0.05	12	8.645	-	0.05	15
16	140, 33, 386	12.353	-	0.03	1	12.854	-	0.09	44
18	161, 37, 448	13.985	-	0.11	27	14.611	-	40.36	116413
20	201, 41, 562	15.268	-	0.14	21	15.998	-	183.56	417067
22	228, 45, 622	15.794	-	0.20	49	16.396	-	167.09	359804
25	286, 51, 792	18.218	-	0.42	110	19.064 *	-	79.25	109259
29	382, 59, 1064	23.302	-	0.61	82	24.723 *	-	5.92	5385
35	532, 71, 1430	26.683	-	0.97	90				
40	625, 81, 1656	28.250	-	0.91	34				

* For a Relative Gap Tolerance of 0.01

Table 5. Overall earliness minimization results with the model of Méndez *et. al.* (2001)

n	Binary vars, Continuous vars, Constraints	Sequence independent setup times				Sequence dependent setup times			
		Objective Function	Relative Gap (%)	CPU time (sec.)	Nodes	Objective Function	Relative Gap (%)	CPU time (sec.)	Nodes
12	82, 24, 214	1.026	-	0.03	22	1.376	-	0.01	12
16	140, 32, 382	9.204	-	1.30	3668	11.647	-	2.70	8301
18	161, 36, 444	16.496	-	38.48	84843	18.773	-	55.77	123666
20	201, 40, 558	17.073	-	77.78	148101	19.131	-	81.38	159388
22	228, 44, 618	22.815	1.68	>3600	4385294	27.754	8.33	>3600	3616973
25	286, 50, 788	29.430	49.63	>3600	2720801	40.541	57.68	>3600	3435213

Table 6. Overall earliness minimization results with the proposed model

n	Binary vars, Continuous vars, Constraints	Sequence independent setup times				Sequence dependent setup times			
		Objective Function	Relative Gap (%)	CPU time (sec.)	Nodes	Objective Function	Relative Gap (%)	CPU time (sec.)	Nodes
12	191, 24, 245	1.026	-	0.02	1	1.376	-	0.03	1
16	351, 32, 437	9.204	-	0.30	78	11.647	-	0.56	404
18	408, 36, 508	16.496	-	1.19	785	18.773	-	1.20	853
20	519, 40, 639	17.073	-	1.63	807	19.131	-	3.27	2178
22	574, 44, 721	22.815	-	9.11	6598	27.754	-	90.75	75569
25	738, 50, 916	29.430	-	91.14	57741	37.216	15.25	>3600	2006319

All results were found on a Pentium IV PC (2.8 GHz) with ILOG OPL Studio 3.7, using the embedded CPLEX v. 9.0 mixed-integer optimizer. CPU time limit was defined on 1 hr. Except for the two cases indicated in Table 4, the solver default relative gap tolerance equal to 0.0001 was used. The time horizon limit H = 30 was used as the big-M parameter.

The results for the makespan minimization problem (Example 1) are shown in Table 3 for the model of Mendez *et. al.* (2001), and in Table 4 for the improved formulation. The proposed model is always faster for each problem instance being tackled. For sequence independent problems, 40 batches are scheduled in less than a second. For sequence-dependent problems almost optimal solutions are found in few CPU seconds, since the relative gap decreases significantly faster because of the tightening constraints.

For Example 2 the corresponding results are shown in Tables 5 and 6. Direct comparison for the 20 batches problem shows an improvement on the computational time of 47:1 for the sequence dependent and of 24:1 for the sequence independent cases. Since tightening constraints for this example do not have a notorious effect until assignments are made, the lower bound for the proposed formulation increases slower than before.

For both examples, the model of Méndez *et. al.* (2001) reaches good (most optimal) solutions in few seconds, but needs larger computational effort to prove their optimality, since the lower bound on the value of the objective function increases very slowly.

5. CONCLUSIONS

A pair of improved MILP formulations for single-stage batch scheduling problems that efficiently handle sequence dependent setup times has been proposed. Better computational results are achieved by incorporating tightening constraints that increase the lower bound on the objective function value. Problems of up to 40 batches have been solved in a very low CPU time.

REFERENCES

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

Ierapetritou, M.G. and C.A. Floudas (1998). Effective Continuous-Time Formulation for Short-Term Scheduling. 1 Multipurpose Batch Processes. *Industrial and Engineering Chemistry Research* **37**, 4341-4359.

Ierapetritou, M.G., T.S. Hené and C.A. Floudas (1999). Effective continuous-time formulation for short-term scheduling. 3 Multiple intermediate due dates. *Industrial and Engineering Chemistry Research* **38**, 3445-3461.

Méndez, C.A., G.P. Henning and J. Cerdá (2001). An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities. *Computers and Chemical Engineering* **25**, 701-711

Méndez, C.A. and J. Cerdá (2003). Dynamic scheduling in multiproduct batch plants. *Computers and Chemical Engineering* **27**, 1247-1259.

Pinto, J.M. and I.E. Grossmann (1995). A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants. *Industrial and Engineering Chemistry Research* **34**, 3037-3051.