A New Continuous-Time Formulation for the Scheduling of Single Stage Batch Plants with Sequence Dependent Changeovers

Pedro Castro[1], Ignacio Grossmann[2] and Augusto Novais[1]
[1]DMS/INETI, 1649-038 Lisboa, Portugal
[2]Dep. Chemical Engineering, Carnegie Mellon University, Pittsburgh PA 15213, USA

Abstract

This paper presents a new multiple-time grid, continuous-time mixed integer linear program (MILP) model for the short-term scheduling of single stage, multiproduct plants featuring equipment units with sequence dependent changeovers. It considers combined processing and changeover tasks as opposed to handling them separately and is very versatile in terms of type of objective function that can be efficiently handled (minimization of total cost, total earliness and makespan). The performance of the formulation is compared to other MILP, constraint programming (CP) and hybrid MILP/CP models with the results showing that the new formulation is overall the best approach.

1. Introduction

Scheduling problems can be tackled by a variety of optimization approaches as well as other solution methods (Méndez et al., 2006). For instance, mathematical programming (MP) models, usually leading to MILP problems, have received considerable attention in the literature. The focus has ranged essentially from specific to general types of network configurations, from pure batch to pure continuous type of processes, from short-term to periodic modes of operation and from discrete to continuous representations of time. While some are more robust than others, small changes in the characteristics of the problem can make some MP models non-applicable, while others become highly inefficient. Constraint programming (CP), Hentenryck (1989), originally developed to solve feasibility problems, has also been extended to solve optimization problems, particularly scheduling problems. CP and MP approaches have complementary strengths (Hooker, 2002) and some researchers (Jain and Grossmann, 2001, Maravelias, 2006) have already taken full advantage of this, by developing hybrid methods that are considerable more efficient than the standalone approaches.

Most of the recent MP scheduling models are based on a continuous-time representation. Those employing one or more time grids focus on general multipurpose plants and on the development of increasingly efficient models. An important recent advance was the introduction by Sundaramoorthy and Karimi (2005) of a formulation without big-M constraints that proved more efficient than other competing methods. Discrete-time formulations for scheduling of multipurpose plants go further back in time starting with the seminal paper of Kondili et al. (1993) that also had the merit of introducing the state-task network (STN) process representation, and which was followed shortly after by the resource-task network (RTN) based model of Pantelides (1994). Discrete and continuous-time approaches have complementary strengths and a mixed-time representation model has recently been presented by Maravelias (2005) for the simultaneously optimization of scheduling and supply chain management problems.

This paper follows another by the same authors (Castro and Grossmann, 2006), which has focused on single stage problems featuring either sequence independent changeovers or none at all. In it, different optimization approaches that are capable of solving single stage problems with sequence dependent changeovers are compared, and a new multiple time grid continuous formulation is proposed. In the new model, processing and changeover tasks are combined into a single set of tasks, which contributes to both reduction in the number of model variables and solution degeneracy, and this is done also for the discrete-time model (DT). The other approaches involved in the comparison are the continuous-time model with global precedence sequencing variables (SV) of Harjunkoski and Grossmann (2002), a CP model based on the same work that uses the OPL Studio modeling language, and a hybrid MILP/CP model, first proposed by Jain and Grossmann (2001) and more recently improved by Maravelias (2006). Not included is the single grid, continuous-time model of Castro et al. (2004), because it was shown to be a poor performer in the previous study and due to the fact that consideration of changeover tasks would inevitable lead to even larger MILPs.

2. Problem statement

The short-term scheduling problem of single stage plants with sequence dependent changeovers can be characterized as follows. Given are a set I of product orders to be processed on a set M of dissimilar equipment units, where any given unit m can process all orders belonging to set I_m; the duration of the processing, $p_{i,m}$, and changeover (cleaning) tasks, $cl_{i,i',m}$, as well as the release, r_i and due dates, d_i, all treated as hard constraints; the processing cost is $c_{i,m}$, whenever the objective is the minimization of total cost. Two other objective functions are considered: the minimization of total earliness and the minimization of makespan.

3. Handling of time

The several formulations considered in this paper treat time differently. The new multiple time grid continuous-time formulations as well as the RTN-based discrete-time formulation, divide the time horizon into a fixed number, |T|-1, of time intervals. The number of tasks that can fit into the time horizon is greatly dependent on the number of time points, in time-grid based continuous-time formulations (also sometimes called event points), and less so in discrete-time formulations. Fewer time points are also used by the former type of approach (a dozen is a practical upper bound), while the latter usually rely on tens or even hundreds of them. Another major difference is that while the discrete-time formulation features equal length intervals, meaning that the time corresponding to each time point is known a priori, continuous-time formulations treat such times as model variables.

The new multiple time grid continuous-time formulations uses several time grids to locate the tasks. More specifically, |M| unit specific time grids are employed. It is assumed that all feature the same number of event points although it is straightforward to adapt them to a different number per grid. The rationale behind this option is that the use of a different number of time points per grid increases the number of a priori decisions to make that can affect the quality of the final solution, and also because that option has already been found to be an efficient one (Castro and Grossmann, 2006). The selection of the cardinality of set T involves the following trade-off: too few points makes it impossible to find the global optimum, and too many makes the problem intractable. Usually, a few MILPs need to be solved in sequence, by using single increments in |T|, until no further improvements in the objective function are observed.

The other featured continuous-time formulation does not rely on explicit time grids. Instead of allocating tasks to different time intervals, a totally different concept is exploited that relies on sequencing variables to ensure that every machine only handles one order at a time. An important advantage when compared to the other continuous-time and discrete-time formulations is that no decisions need to be taken before solving the problem that may eventually compromise its solution. That is, the model needs only to be solved once and the resulting solution will always be an exact and global optimal solution (if solved to optimality). The same can be said for the CP formulation, although this is more accurately classified as a discrete-time model since all the activities (i.e. tasks) must have integer durations.

4. New multiple time grid formulation

The formulation (CT) uses 4-index binary variables, $N_{i,i',m,t}$, to assign the execution of the combined processing and changeover task, to a particular machine and also to a certain time point. The other set of binary variables used are the excess resource variables $R_{m,t}$ that identify equipment availability (=1) at a given event point. The remaining variables, all nonnegative, are the timing variables $T_{t,m}$ and also the new variables $C_{i,m,t}$, which are also excess resource variables that, when equal to 1, indicate that equipment *m* is ready to handle order *i* at time point *t*. The initial equipment state is also a model variable, $C_{i,m}^{0}$.

The constraints that compose the multiple time grid formulation are given next, where the set $I_{i',m,t}$ represents the domain of variables $N_{i,i',m,t}$. Eq 1 represents the excess resource balance for the equipment resources. It is a typical multiperiod material balance expression in which the excess amount at point *t* is equal to that at point *t*-1 adjusted by the amounts produced/consumed by all tasks starting or ending at *t*. Eq 2 is the equivalent constraint for the equipment states but is slightly more complex due to the fact that the execution of a given task usually involves production and consumption of different states. Eq 3 ensures that there is only one initial equipment state. To reduce solution degeneracy and to improve the performance of the model, we enforce tasks to be allocated to time points with as low as index as possible, eq 4. All orders need to be processed exactly once, eq 5. Eq 6 states that the difference between the absolute times of any two time points must be greater than the duration of the combined task. Eq 7 ensures that any order only starts to be processed after its release date. Eq 8 is the due date constraint, where $ub_{i,i',m}$ is the highest time at which order *i* can start to be processed on unit *m* and still be followed by order *i'* (see eq 9). The global lower bound on the value of the timing variables is the minimum release date (eq 10), while the global upper bound is the maximum due date (eq 11).

$$R_{m,t} = 1 - \sum_{i' \in I_m} \sum_{i \in I_{i',m,t}} N_{i,i',m,t} \quad \forall m \in M, t \in T \tag{1}$$

$$C_{i,m,t} = C^0_{i,m}\Big|_{t=1} + C_{i,m,t-1}\Big|_{t\neq 1} + \sum_{i' \in I_{i,m,t-1}} N_{i',i,m,t-1} - \sum_{\substack{i' \in I_m \\ i \in I_{i',m,t}}} N_{i,i',m,t} \ \forall i \in I, m \in M_i, t \in T \tag{2}$$

$$\sum_{i \in I_m} C^0_{i,m} = 1 \ \forall m \in M \tag{3}$$

$$R_{m,t} \geq R_{m,t-1} \ \forall m \in M, t \in T, t \neq 1 \tag{4}$$

$$\sum_{\substack{t \in T \\ t \neq |T|}} \sum_{m \in M} \sum_{\substack{i' \in I_m \\ i \in I_{i',m,t}}} N_{i,i',m,t} = 1 \ \forall i \in I \tag{5}$$

$$T_{t+1,m} - T_{t,m} \geq \sum_{i' \in I_m} \sum_{i \in I_{i',m,t}} [N_{i,i',m,t} \cdot (p_{i,m} + cl_{i,i',m})] \ \forall m \in M, t \in T, t \neq |T| \tag{6}$$

$$T_{t,m} \geq \sum_{i' \in I_m} \sum_{i \in I_{i',m,t}} N_{i,i',m,t} r_i \ \forall m \in M, t \in T, t \neq |T| \tag{7}$$

$$T_{t,m} \leq \sum_{i' \in I_m} \sum_{i \in I_{i',m,t}} N_{i,i',m,t} ub_{i,i',m} + H(1 - \sum_{i' \in I_m} \sum_{i \in I_{i',m,t}} N_{i,i',m,t}) \ \forall m \in M, t \in T, t \neq |T| \tag{8}$$

$$ub_{i,i',m} = \min(d_i - p_{i,m}, d_{i'} - p_{i,m} - cl_{i,i',m} - p_{i',m}\Big|_{i \neq i'}) \ \forall i, i' \in I, m \in M_i \cap M_{i'} \tag{9}$$

$$T_{t,m} \geq \min_{i \in I_m} r_i \ \forall m \in M, t \in T \tag{10}$$

$$T_{t,m} \leq H = \max_{i \in I_m} d_i \ \forall m \in M, t \in T \tag{11}$$

Whenever the objective is makespan minimization, a new variable is required (*MS*) that must be greater than the ending time of all tasks. Eq 12 is a constraint that ensures this goal by relating the variable to the starting time of all time points. The second term on the right-hand side represents the duration of all combined tasks starting in unit *m* at or after time point *t*. It is a new term since the objective of makespan minimization was not considered in Castro and Grossmann (2006).

$$MS \geq T_{t,m} + \sum_{\substack{t' \in T \\ t' \geq t \\ t' \neq |T|}} \sum_{i' \in I_m} \sum_{i \in I_{i',m,t'}} [N_{i,i',m,t'} \cdot (p_{i,m} + cl_{i,i',m})] \ \forall m \in M, t \in T, t \neq |T| \tag{12}$$

The mathematical formulation can handle three alternative objective functions. These are total cost minimization, eq 13, total earliness minimization, eq 14, and makespan minimization, eq 15. Note that the second term in eq 14 accounts for the delivery dates of all orders and that the third subtracts the absolute time of all points (the actual number is know *a priori* from the sets cardinalities) that have no tasks starting at.

$$\min \sum_{\substack{t \in T \\ t \neq |T|}} \sum_{m \in M} \sum_{i' \in I_m} \sum_{i \in I_{i',m,t}} N_{i,i',m,t} c_{i,m} \tag{13}$$

$$\min \sum_{i \in I} d_i - \sum_{\substack{t \in T \\ t \neq |T|}} \sum_{m \in M} (T_{t,m} + \sum_{i' \in I_m} \sum_{i \in I_{i',m,t}} N_{i,i',m,t} p_{i,m}) - H[|I| - |M| \cdot (|T| - 1)] \tag{14}$$

$$\min MS \tag{15}$$

5. Computational results

The performance of the five different approaches considered is illustrated trough the solution of 18 example problems. These are identified by a number (1-6) and two additional characters, where the last one identifies the objective function being considered. The problems under consideration range from 12 orders in 3 units to 20 orders in 5 units. We have used GAMS/CPLEX 9.1 for the solution of the MILPs and all problems were solved to optimality (relative tolerance=1E-6), unless otherwise stated. The CP and hybrid MILP/CP models were solved in ILOG OLP Studio 3.7.1. Concerning hardware, we have used a Pentium-4, 3.4 GHz computer with 1 GB of RAM and running Windows XP Professional.

For total cost minimization, the results given in Table 1 clearly show that the new continuous-time formulation (CT) is the best performer by a significant margin to all but the hybrid MILP/CP model. The continuous-time model with global precedence sequencing variables (SV) could always find the global optimal solution but failed to prove optimality in three cases (P3C, P5C and P6C), either because the maximum resource limit was achieved or because the solver ran out of memory. The CP model exhibited a better performance than SV but failed to find the optimal solution for P5C. At the bottom of the list comes the discrete-time formulation (DT), which due to the large number of time points required to handle the exact problem data, could only solve approximate versions of the problems.

For total earliness minimization, the new formulation continues to be the best performer even though it fails to find the optimal solution for P6E. For this, we have to rely on the discrete-time formulation, which contrary to what happens with the other objectives, is also a very good performer. In particular, DT can solve all problems but P5E with the exact problem data. The other two formulations, SV and CP, perform worse than for total cost. For these, P3E is the most interesting problem since CP terminated with an optimal solution (561) that is in fact suboptimal. The use of global precedence sequencing variables either explicitly (SV) or implicitly in CP global constraints, results in an important limitation: the difference between the starting points of any two orders i and i' (with i preceding i') cannot be lower than $p_{i,m}+cl_{i,i',m}$ even if there is a shorter duration order i'' that can fit in between, i.e., $cl_{i,i'',m}+p_{i'',m}+cl_{i'',i',m}<cl_{i,i',m}$. Nevertheless, it is unlikely that such combination of processing data occurs in a real industrial environment. The hybrid MILP/CP model was not used since this objective depends both on the assignment and sequencing variables. In such case, its performance is likely to worsen as then the CP part of the model has to solve an optimization rather than a feasibility problem. The same can be said for makespan minimization.

The objective of makespan minimization is the most difficult for the new formulation. However, even when it fails to find the optimal solution (338 vs. 337) for P5M and (168 vs. 164) for P6M, it can always find good solutions. The other continuous-time formulation (SV) has now a similar performance to CT. The best performer is, however, the constraint programming model, since it is the fastest for P1M-P3M and can also find the best solution for P5M, for which the optimal solution is still unknown. Nevertheless, its performance for P6M is rather weak since the best solution found after more than 15 h of computational time is still far from the best solution (237 vs. 164), which was discovered by the discrete-time formulation.

Table 1. Overview of computational performance

Problem/model	Optimum	CT	SV	DT	CP	MILP/CP
P1C (|I|=12, |M|=3)	101	10.0	118	713[m]	0.75	13.4
P2C (|I|=12, |M|=3)	87	5.42	33.1	1250[n]	0.19	1.11
P3C (|I|=15, |M|=5)	121	23.2	20000[d]	1395[n]	133	37.4
P4C (|I|=15, |M|=5)	106	3.80	510	2602[o]	15.1	6.81
P5C (|I|=20, |M|=5)	163	66.9	12217[e]	551[n]	57000[u]	6935
P6C (|I|=20, |M|=5)	146	27.9	15844[f]	2084[p]	7620	83.3
P1E (|I|=12, |M|=3)	690	4.28	427	451	2.36	-
P2E (|I|=12, |M|=3)	146	4.11	653	190	92.1	-
P3E (|I|=15, |M|=5)	559	17.9	6842[g]	4506	11146[v]	-
P4E (|I|=15, |M|=5)	54	4612	9059[h]	169	3058	-
P5E (|I|=20, |M|=5)	1187	208	9821[i]	3614[q]	83000[w]	-
P6E (|I|=20, |M|=5)	150	62.3[a]	6259[j]	522	142000[x]	-
P1M (|I|=12, |M|=3)	409	15.5	17.5	8073[r]	0.98	-
P2M (|I|=12, |M|=3)	171	3.55	14.9	29509	0.84	-
P3M (|I|=15, |M|=5)	291	201	17255	980[s]	122	-
P4M (|I|=15, |M|=5)	147	1435	2.42	61.5	702	-
P5M (|I|=20, |M|=5)	337?	8782[b]	8011[k]	8311[t]	54000[y]	-
P6M (|I|=20, |M|=5)	164?	6290[c]	16062[l]	13891[n]	55000[z]	-

AS=approximate solution of the problem due to rounding of the problem data (DT). FTP=fewer time points were used than those required to find the optimal solution, |T| value within brackets. BPS=best possible solution at the time of termination. MRL=maximum resource limit exceeded. NS=no solution found. OM=solver ran out of memory. SO= suboptimal solution returned. [a]SO=164, FTP(|T|=5), OM for |T|=6 with worse solution. [b]OM, SO=338, BPS=330. [c]OM, SO=168, BPS=158. [d]MRL, BPS=119.09.[e]OM, BPS=155.42. [f]OM, BPS=143.14. [g]OM, SO=667, BPS=0. [h]OM, BPS=0. [i]OM, SO=1458, BPS=0. [j]OM, SO=190, BPS=0. [k]OM, SO=347, BPS=290. [l]OM, SO=167, BPS=147. [m]AS, SO=105. [n]AS. [o]AS, SO=107. [p]AS, SO=147. [q]AS, SO=1230. [r]AS, SO=410. [s]AS, SO=295. [t]AS, SO=340. [u]MRL, SO=166. [v]SO=561, although solver solved to optimality (special case). [w]MRL, SO=1214. [x]MRL, SO=767. [y]MRL, best solution=337. [z]MRL, SO=237.

8. Conclusions

This paper has presented a new continuous-time formulation for the short-term scheduling of single stage, multiproduct batch plants, where equipment units are subject to sequence dependent changeovers, and product orders to both release and due dates. The formulation relies on the use of multiple time grids, one per equipment resource and it was found to very efficient on a set of 18 example problems. A computational comparison to other optimization approaches, such as a discrete-time, a continuous-time with global precedence sequencing variables, a constraint programming and a hybrid MILP/CP model, highlighted their strengths and weaknesses.

References

Castro, P., Grossmann, I.E., 2006. Comp. Chem. Eng., 30, 1003.
Harjunkoski, I., Grossmann, I.E., 2002. Comp. Chem. Eng., 26, 1533.
Hentenryck, P.V., 1989. Constraint satisfaction in logic programming. MIT Press: Cambridge, MA.
Hooker, J.N., 2002. INFORMS J. Comp., 14, 295.
Jain, V., Grossmann, I.E., 2001. INFORMS J. Comp., 13, 258.
Kondili, E., Pantelides, C., Sargent, R., 1993. Comp. Chem. Eng., 17, 211.
Maravelias, C., 2005. Ind. Eng. Chem. Res., 44, 9129.
Maravelias, C., 2006. Comp. Chem. Eng., 30, 407.
Méndez, C., Cerdá, J., Grossmann, I.E., Harjunkoski, I., Fahl, M., 2006. Comp. Chem. Eng., 30, 913.
Pantelides, C., 1994. In Proc. 2nd FOCAPO. Cache Publications: New York, 253.
Sundaramoorthy, A., Karimi, I.A., 2005. Chem. Eng. Sci., 60, 2679.