CHEMICAL ENGINEERING TRANSACTIONS, Volume 12, 2007
Edited by Jiri Klemeš
Copyright © 2007, AIDIC Servizi S.r.l., ISBN 88-901915-4-6

Throughput Maximization In Multipurpose Batch Plants: S-Graph Vs Time Point Based Methods

[1]T. Holczinger, [1,2]T. Majozi, [1]M. Hegyhati and [1]F. Friedler[‡]

[1]Department of Computer Science, University of Pannonia, Egyetem u. 10, Veszprém, H-8200, Hungary
[2]Department of Chemical Engineering, University of Pretoria, Lynnwood Road, Pretoria, 0002, South Africa

This paper presents a performance comparative study between a graph-theoretic framework based on the S-graph representation and a mathematical programming approach based on uneven discretization of the time horizon using time points. Both techniques consider maximization of throughput in multipurpose batch plants over a predefined time horizon. The major contrast between these techniques is the irrelevance of time discretization in the S-graph approach and the necessity of presupposition of time points in the mathematical programming approach. Moreover, the S-graph provides added unique advantages like readily exploiting the structure of the problem at hand and elimination of infeasible solutions prior to embarking on problem solution. In all the problems considered, the S-graph succeeded in obtaining solutions in relatively short CPU times compared to the mathematical programming approach. In essence, the mathematical programming approach could not provide solutions in some problems of industrial relevance.

1. Introduction

The scheduling problem entails allocation of resources of limited resources to tasks with the objective of achieving the best predefined performance index. In one category of batch scheduling problems, the number of batches of different products to be produced is known a priori and the objective is to produce all of them using the available resources in the shortest possible time, i.e., makespan. In the other category, the time horizon of interest is known beforehand, and the objective is to determine the allocation of tasks to resources that is concomitant with maximum throughput, revenue or a specifically chosen profit function. On the whole, the problem of scheduling in batch operations is not new. It was initially formulated in 1975 by Sparrow and coworkers (Sparrow et al., 1975) as a nonlinear programming (NLP) problem and has received much more attention in the recent past due to the competitive advantage offered by batch processes in prevalent volatile market conditions. The competitive advantage emanates from their inherent flexibility and adaptability to unexpected changes compared to their continuous counterparts.

Over the years, the scheduling problem has converged to the development of a rigorous time representation framework, since the resulting mathematical formulations strongly depend on this aspect. This follows the work of Kondili et al. (1993) which proposed an even

[‡] Corresponding author: friedler@dcs.vein.hu

discretization of the time horizon of interest. In this formulation the time horizon is discretized into even time intervals that coincide with the activities of tasks in different equipment units. The accuracy of the model is highly dependent on the number, hence the length, of the time intervals. The shorter the time intervals, the more accurate the model. The disadvantage of this approach, however, is that the number of binary variables is directly proportional to the number of time intervals. This implies that accuracy can only be attained with significant computational expense. Consequently, most of the subsequent work in the area of scheduling is focused on developing mathematical formulations that are based on uneven discretization of the time horizon (Schilling and Pantelides, 1996; Zhang and Sargent, 1998; Ierapetritou and Floudas, 1998; Majozi and Zhu, 2001; Castro et al., 2001). These have become known as continuous-time mathematical formulations in literature. The major drawback of the latter formulations is that the number of time points that is appropriate to obtain the optimum is obtained through an iterative procedure which involves incrementing the number of time points until the objective value does not improve for two consecutive increments. In industrial scale problems a single iteration might require a very long time, thereby rendering this technique highly impractical. Moreover, recent studies have shown that the iterative method of determining the appropriate number of time points is not rigorous, since it can easily result in implicit suboptimality. This implies that, although the objective might not improve for two consecutive time points, it might still improve with further increase in time points. One can, therefore, never be certain if convergence of the objective value is a sign of true optimality.
The S-graph framework, on the contrary, does not require any specification of time points since no discretization of the time horizon is necessary. This framework allows the structure of the problem as defined by the recipe and allocation of tasks to various resources or equipment to be directly employed in the search for a global optimum. Added to this unique advantage is the ability of this framework to isolate infeasible solutions prior to solving the entire problem, which ultimately reduces the solution space. One of the infeasible solutions that has gone unnoticed in most published techniques is the one that involves cross-transfer of material between two equipment units. In an S-graph this occurrence is detected by the existence of a loop or cycle in the S-graph. The latter represents a single complete schedule or one of possible solutions in the search space. There exists a feasible schedule for every feasible schedule-graph. This framework has proven successful in both of the aforementioned categories of scheduling problems.

2. Problem statement

The problem addressed in this paper can be summarized as follows. Given,

(i) the production recipe for each product,
(ii) the potential assignment of tasks to equipment units,
(iii) relevant cost data and
(iv) the time horizon of interest,

the objective is to determine the schedule that yields the overall maximum throughput or revenue for all the products involved. The problems considered are based on a Zero-Wait

(ZW) operational philosophy. The presented approach is capable of handling situations involving equipment units of unequal capacities for the same task as commonly encountered in practice.

3. Some background on time-point based techniques

The concept of time points was initially proposed by Schilling and Pantelides (1995) and involves discretization of the time horizon into uneven time intervals as shown in Figure 1. Every time point coincides with the beginning of a task in a particular unit. There is also absolute time that corresponds to every time point relative to the start of the time horizon, H. Using Figure 1 as an example $t_{in}(s_{in,j}, p)$ and $t_{out}(s_{out,j}, p)$ represent the absolute time at which a particular state, $s_{in,j}$, is used in a particular unit j and absolute time at which a particular state, $s_{out,j}$, is produced from unit j at time point p. The binary variable $y(s_{in,j}, p)$ takes the value of 1 if state $s_{in,j}$ is used in a unit at time point p and is equal to 0 otherwise. All 4 major operational philosophies, i.e., Zero-wait (ZW), No-intermediate-storage (NIS), Finite-intermediate-storage (FIS) and Unlimited-intermediate-storage (UIS) can be readily formulated in terms of time as shown in Equation (1) for NIS and Equations (2) and (3) for the ZW case. The FIS and UIS cases are implicitly covered in Equation (1), but require additional auxiliary constraints that handle capacity. In a nutshell, Equation (1) states that a particular state can either be used immediately after it has been produced or be used later when required in the subsequent task along the recipe. This implies that the state can be temporarily stored in the same unit in which it was produced until the subsequent unit is available. Equations (2) and (3) state that a state has to be used immediately after it has been produced. Additional to these constraints, a full model will entail other relevant constraints like sequencing, storage, mass balance and duration constraints.

NIS case

$$t_{in}(s_{in,j}, p) \geq t_{out}(s_{out,j'}, p) - H(2 - y(s_{in,j}, p) - y(s_{in,j'}, p-1)),$$
$$\forall s_{in,j} = s_{out,j'} \subset S, p \in P, p > 1$$

(1)

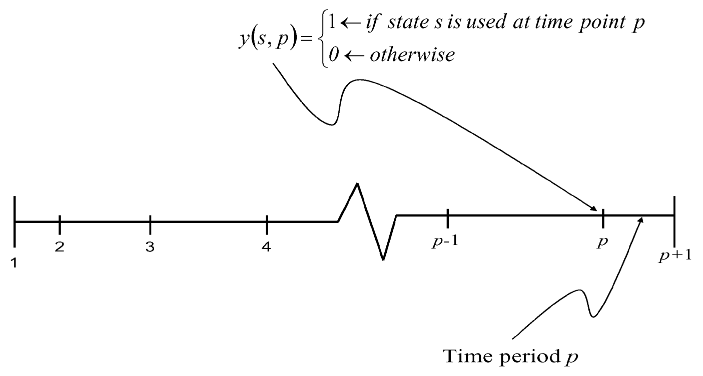
ZW case:

$$t_{in}(s_{in,j}, p) \leq t_{out}(s_{out,j'}, p) + H(2 - y(s_{in,j}, p) - y(s_{in,j'}, p-1)),$$
$$\forall s_{in,j} = s_{out,j'} \subset S, p \in P, p > 1$$

(2)

$$t_{in}(s_{in,j},p) \geq t_{out}(s_{out,j'},p) - H(2 - y(s_{in,j},p) - y(s_{in,j'},p-1)),$$
$$\forall s_{in,j} = s_{out,j'} \subset S, p \in P, p > 1$$

(3)



$y(s,p) = \begin{cases} 1 \leftarrow \textit{if state s is used at time point p} \\ 0 \leftarrow \textit{otherwise} \end{cases}$
1
2
3
4
p-1
p
p+1
Time period p

Figure 1. Description of time points along the time horizon of interest

4. A brief description of the S-graph framework (Sanmartí et al., 1998, 2002)

An S-graph is an advanced graph consisting of two types of arcs, i.e., recipe-arcs and schedule-arcs. Recipe-arcs pertain to the recipes for converting inputs or raw materials to outputs or products. In Figure 2 there are three recipes that correspond to the production of products A, B and C. The recipe for product A involves task nodes 1 and 2 which represent the tasks conducted in units E1 and E2, respectively. It also involves an arc that links task nodes 1 and 2 and an arc linking task node 2 and product node 7. The weight of each of the arcs represents the minimum time it takes for the subsequent task to begin after the preceding task has started. Since these arcs belong to the recipe for product A they are appropriately called recipe-arcs. A similar interpretation applies to recipes for products B and C. On the other hand, arcs (i) and (ii) represent the sequence of tasks in equipment units E1 and E2, respectively. Consequently, they are referred to as schedule-arcs. In the exclusion of arcs (i) and (ii), Figure 2 is referred to as the recipe graph, otherwise it is an S-graph depicting sequences of tasks 1 and 3 and tasks 4 and 5. Arc (i) stipulates that task 3 in equipment E1 can only begin after task 1 in the same equipment is finished and the material is filled to the equipment performing task 2, i.e., at the beginning of task 2 in equipment E2. Arc (ii) stipulates that task 5 in equipment E2 can only begin after task 4 in the same equipment is complete, i.e., once product B is produced. If all the tasks in an S-graph have been scheduled, it is referred to as a Schedule-graph as shown in Figure 3.

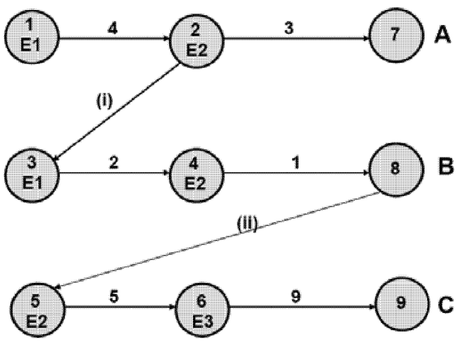


Figure 2. S-graph representation for task sequences 1-3 and 4-5

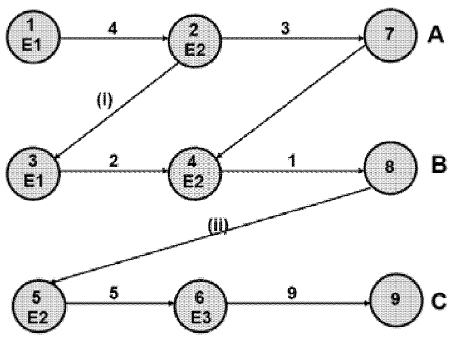


Figure 3. Schedule-graph for a complete schedule

5. Solution procedure using the S-graph approach

The solution procedure using the S-graph approach is based on a guided search within a solution space defined by the structure of the problem under consideration (Majozi and Friedler, 2006). The boundaries of the search space are set by the maximum number of batches of each of the products to be produced within the time horizon of interest. Each of the nodes within the search space represents a unique combination of batches of same or different products. An appropriate analysis of the search space makes it possible to identify a sub-region within which lies an optimum solution. In most instances this analysis excludes a substantial number of nodes that do not have any possibility of providing the optimum solution, thereby drastically reducing the search space. An obvious consequence of this reduction is shorter CPU times for even the most complicated problems. Moreover, not all the nodes within the sub-region of optimality need to be explored in any detail. This implies that some regions within the sub-region of optimality can be further excluded from the search which is an added significant advantage.

6. Case study (Majozi and Zhu, 2001)

This example is for an agrochemical process for the production of an herbicide. The flowsheet for the process is shown in Figure 4. The process considered consists of 5 consecutive steps. The first step involves a reaction which forms an arsenate salt. This reaction requires two raw materials, raw 3 and raw 4, and can be conducted in either reactor R1 or R2. The arsenate salt from the first step is then transferred to either reactor R3 or R4 wherein two reactions take place. The first of these reactions is aimed at converting the arsenate salt to a disodium salt using raw material 1 (raw 1). The disodium salt is then reacted further to form the monosodium salt using raw material 2 (raw 2). The monosodium salt solution is then transferred to the settling step in order to remove the solid byproduct. Settling can be conducted in any of the three settlers, i.e. SE1, SE2 or SE3. The solid byproduct is dispensed with as waste and the remaining monosodium salt solution is transferred to the final step. This step consists of two evaporators, EV1 and EV2, which remove the excess amount of water from the monosodium solution. Evaporated water is removed as effluent and the monosodium salt (product A) is taken to storage. Table 1 shows scheduling data, whilst Table 2 shows stoichiometric data.

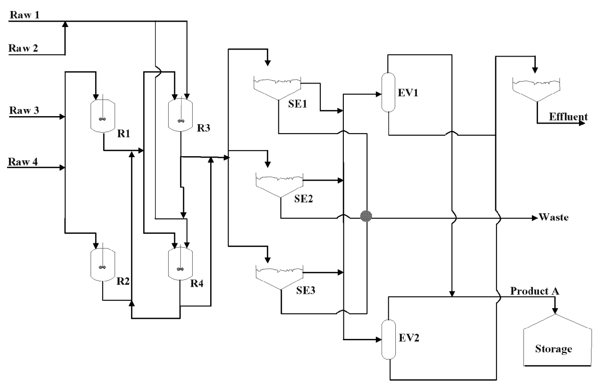


Figure 4. Flowsheet for the case study

Table 1. Scheduling data for the case study

Unit	Capacity	Suitability	Mean processing time (h)
R1	10	Reaction 1 (task 1)	2
R2	10	Reaction 1 (task 1)	2
R3	10	Reaction 2 (task 2), reaction 3 (task3)	3, 1
R4	10	Reaction 2 (task 2), reaction 3 (task 3)	3, 1
SE1	10	Settling (task 4)	1
SE2	10	Settling (task 4)	1
SE3	10	Settling (task 4)	1
EV1	10	Evaporation (task 5)	3
EV2	10	Evaporation (task 5)	3

7. Results and discussion

In this problem the time point based technique involved 1402 constraints, 88 binary variables, 737 continuous variables and was solved in 745.8 CPU seconds using GAMS/CPLEX solver in a 3 GHz Pentium 4 processor. The S-graph technique, on the other hand, obtained the solution within 0.28 CPU seconds in the same processor.

8. Conclusion

The S-graph approach has demonstrated excellent performance in comparison with its time point based counterparts. The S-graph approach can be classified as truly continuous in time since it neither involves any specification of time points nor discretization of the time horizon of interest.

9. References

Sparrow, R.E., Forder, G.J., Rippin, D.W.T., 1975, The choice of equipment sizes for multiproduct batch plants – heuristics vs branch and bound, *Ind. Eng. Chem. Proc. Des. Dev.*, 14: 197 – 203.

Kondili, E.; Pantelides, C. C.; Sargent, R. W. H., 1993, A general algorithm for short-term scheduling of batch operations. I. MILP formulation. *Comp. Chem. Eng.*, *17* (2), 211 - 227.

G. Schilling and C. C. Pantelides, 1996, "A simple continuous-time process scheduling formulation and a novel solution algorithm", Comput. Chem. Eng., 20: S1221-S1226.

X. Zhang and R.W.H. Sargent, 1998, "The optimal operation of mixed production facilities—extensions and improvements", Comput. Chem. Eng., 22: 1287–1295.

Ierapetritou, M. G., & Floudas, C. A. (1998a). Effective continuous-time formulation for short-term scheduling. Part 1. Multipurpose batch processes, *Ind. Eng. Chem. Res, 37*: 4341 – 4359.

Majozi, T., Zhu, X.X., 2001, A novel continuous time MILP formulation for multipurpose batch plants. 1. Short-term scheduling, *Ind. Eng. Chem. Res.*, 40(25): 5935 – 5949.

Castro, P., Barbosa-Póvoa, A. P. F. D., Matos, H., 2001, An Improved RTN Continuous-Time Formulation for the Short-term Scheduling of Multipurpose Batch Plants, *Ind. Eng. Chem. Res.*, 40: 2059 – 2068.

Sanmartí, E., F. Friedler, L. Puigjaner, 1998, Combinatorial technique for short term scheduling of multipurpose batch plants based on schedule-graph representation, *Comp. Chem. Eng.*, 22 (Suppl.): S847-S850.

Sanmartí, E., T. Holczinger, L. Puigjaner, and F. Friedler, 2002, Combinatorial Framework for Effective Scheduling of Multipurpose Batch Plants, *AIChE J.,* 48(11): 2557 - 2570.

Majozi, T., Friedler, F., 2006, Maximization of Throughput in a Multipurpose Batch Plant under Fixed Time Horizon: S-Graph Approach, *Ind. Eng. Chem. Res.*, 45, 6713 - 6720.