17th European Symposium on Computer Aided Process Engineering – ESCAPE17
V. Plesu and P.S. Agachi (Editors)
© 2007 Elsevier B.V. All rights reserved.

An Automated Algorithm for Throughput Maximization Under Fixed Time Horizon in Multipurpose Batch Plants: S-Graph Approach

Tibor Holczinger,[a] Thokozani Majozi,[a,b] Mate Hegyhati[a] and Ferenc Friedler[a]

[a]*Department of Computer Science, University of Pannonia, Egyetem u. 10, Veszprém, H-8200, Hungary, Friedler@dcs.vein.hu*
[b]*Department of Chemical Engineering, University of Pretoria, Lynnwood Road, Pretoria, 0002, South Africa*

Abstract

This paper presents a graph-theoretic technique for scheduling of multipurpose batch processes with the objective of maximizing throughput over a fixed time horizon. The presented technique is based on the S-graph framework which embeds the necessary scheduling information pertaining to recipes of different batches and possible resource allocation. The distinct feature of this contribution is the irrelevance of time horizon discretization which has become traditional in most mathematical programming techniques. Consequently, no presupposition of time points is required, thereby rendering the technique truly continuous in time. A comparison with time point based techniques is conducted to demonstrate the performance.

Keywords: S-graph, batch plants, continuous-time, scheduling

1. Introduction

In its basic form, the problem of scheduling involves the optimal allocation of tasks to limited resources. In a situation where the number of batches to be produced for each product is known *a priori*, the objective is usually to minimize makespan. On the other hand, it is also common in practice to have a fixed time horizon and be required to determine the maximum possible throughput or revenue that can be accomplished, given the recipes and economic contributions of various products. It is the latter problem that is considered in this paper. Since its inception, almost 3 decades ago, the scheduling problem has been tackled by several researchers using different techniques. Following the contribution of Kondili et al. (1993), recent techniques rely heavily on the discretization of the time horizon to capture the activities of various tasks in different units (Zhang and Sargent, 1998; Schilling and Pantelides, 1996). These techniques invariably require the presupposition of time points that coincide with the start or end of a task in a particular unit. Consequently, the more the number of time points, the more the number of binary variables. There currently exists no method for predicting the adequate or appropriate number of time points that results in an optimal solution before embarking on solving the problem. A detailed review on these techniques has been given by Floudas and Lin (2004).

Presented in this paper is the most recent method that does not rely on the discretization of the time horizon, but exploits the structure of the problem at hand to derive an optimal schedule (Majozi and Friedler, 2006). The method uses a graph-theoretic framework known as the S-graph (Sanmarti et al., 2002) to derive all feasible schedules and isolate the optimum schedule corresponding to a chosen performance index. The added powerful advantage of this framework is its inherent capability to exclude infeasible solutions that are almost impossible to isolate beforehand using mathematical programming techniques. An example of this infeasibility involves cross-transfer of material between two equipment units. This is a practically infeasible situation in the absence of intermediate storage. In the S-graph framework, this situation is detected by the existence of a loop or cycle. This framework has proven efficient in both the makespan minimization and throughput maximization problems. A demonstration of the performance of this technique compared to time point based techniques is also presented.

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,
determine the schedule that is concomitant with maximum throughput or revenue for all the products involved. Only the no intermediate storage (NIS) policy is considered in this paper.

3. S-graph framework

An S-graph is an advanced graphical representation that entails 2 types of arcs, i.e., recipe arcs and schedule arcs. Recipe arcs pertain to the sequence of tasks in a particular recipe, whilst schedule arcs relate to the sequencing of tasks belonging to different batches or products in a particular unit. The weight of the arc connecting nodes *i* and *j* denotes the minimum time it takes before task *j* begins after task *i* has started. Without arcs (i) and (ii), Fig. 1 depicts the recipe graph of products A, B and C. Nodes 1 – 6 represent task nodes, whilst nodes 7 – 9 represent product nodes. Also shown within the task nodes are the equipment units that are capable of conducting the corresponding tasks. With the exception of arcs (i) and (ii), all the other arcs represent recipe arcs, hence the name recipe graph. Arc (i) is a schedule arc representing the sequencing of tasks 1 and 3 in equipment 1, and arc (ii) the sequencing of tasks 5 and 2 in equipment 2. As a result, arcs (i) and (ii) are referred to as schedule arcs. If all tasks in a particular recipe graph have been scheduled, then the resulting graph is termed S-graph.

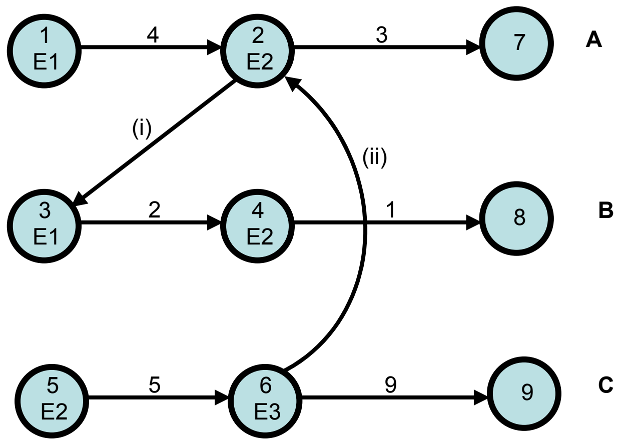

Figure 1: S-graph representation for task sequences 1-3 and 5-2

4. Solution procedure using S-graph approach

The solution procedure involves a guided search within a region derived from the structure of the problem. The search is efficient due to two main reasons. Firstly, at each point in the search a node with a unique combination of batches of products is explored, thereby eliminating unnecessary redundancy. Redundancy tends to result in lengthy CPU times which might not be warranted in practice. Each node in the search can either involve batches of the same product or different products among those to be produced over the time horizon of interest. Secondly, the composition of the search space makes it possible to eliminate the set nodes that do not involve the optimal solution. As a result, these can be eliminated *a priori* from the search. The reduction of the search space in this manner implies that fewer nodes have to be explored for possible optimality which invariably reduces the CPU time. The S-graph framework is used to determine a feasible schedule at each node of the search, since a fixed number of batches of each product of interest is known. Feasibility of a schedule implies that the makespan associated with a given node is less than the time horizon of interest and the schedule entails no cycles. The latter signify practical infeasibility, e.g. two equipment units exchanging product simultaneously or the task sequence that violates the recipe.

5. Literature example

The flowsheet for the literature example is shown in Fig. 2. The example involves a heater, two reactors and a separator. Each of the reactors can conduct 3 reactions, i.e. reaction 1, 2 and 3. In both reactors, the reaction durations are 2, 2 and 1 hour, respectively. Heating takes 1 hour, whilst the separation is 2 hours. The process operates in an no intermediate storage (NIS) philosophy. The objective is to maximize revenue for products 1 and 2 over an 18 h time horizon.

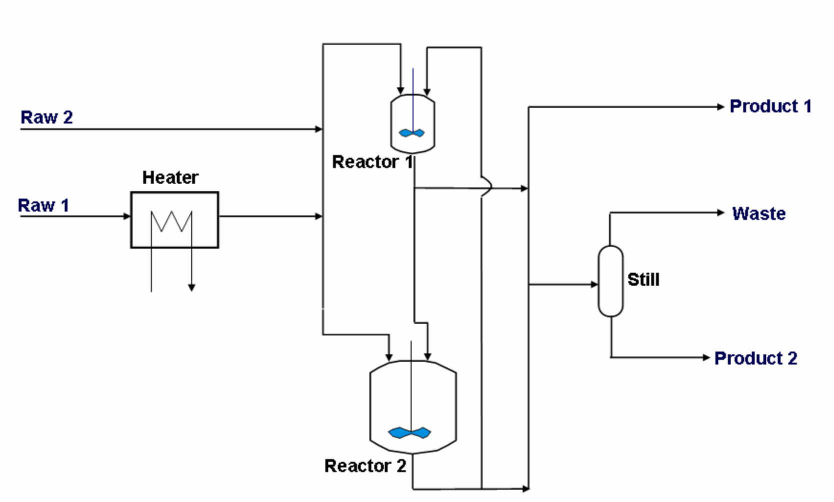

Figure 2: Flowsheet for example 1

6. Results & discussion

A time point based technique was used to determine the optimum schedule. Iteration to determine the appropriate number of time points involved a minimum of 11 and a maximum of 14 time points. Convergence of the objective value for 2 consecutive time points suggests that the right number of time points has been attained. However, there have been instances where the objective value improves with a further increase in time points, even after this convergence has been reached. Therefore, 3 instead of 2 time points were used to improve confidence on convergence. Using the time point approach, an objective value of 4515 cost units was obtained after 7951.6 CPU seconds.

On the other hand, the S-graph approach obtained the same solution in 4086.61 seconds after exploring only 172 out of 437 partial problems in the search space. Fig. 3 shows the schedule corresponding to maximum throughput.

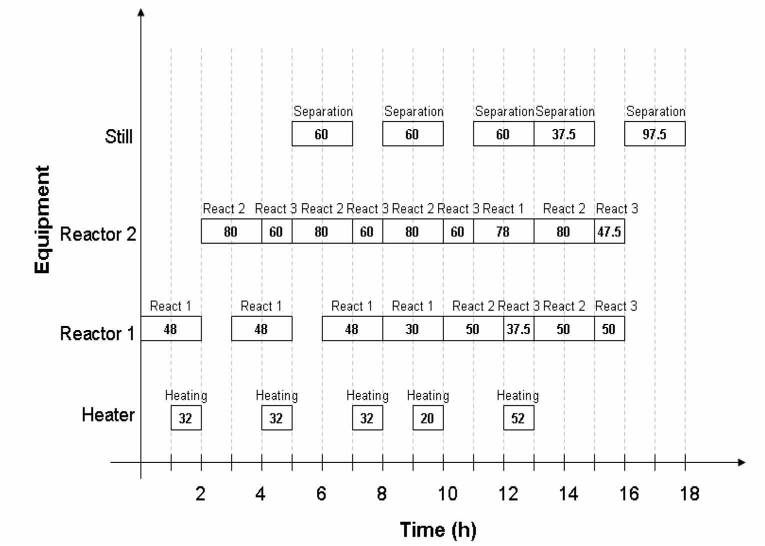

Figure 3: Optimum schedule for literature example

7. Conclusion

Presented in this paper is a continuous-time technique based on the S-graph framework for throughput maximization in multipurpose batch plants. The continuous-time feature of the presented procedure emanates from the unnecessary presupposition of time points that is prevalent in most MILP based techniques. This has proven very impractical in most industrial scale operations as it requires an initial search for the appropriate number of time points before solving the problem to optimality. This paper also highlights that optimality cannot be guaranteed using the time point based technique as the objective value can still improve after several iterations. This observation implies that convergence of the objective value is not a rigorous criterion for the appropriate number of time points.

In addition, the S-graph framework which forms the basis of the algorithm presented in this paper allows the exploitation of problem specific structure to arrive at the solution in reasonable CPU times. The unique advantage of the S-graph is its ability to isolate infeasible solutions prior to embarking on a detailed search for the optimal solution. This is mainly achieved by the identification of loops in the Schedule Graph. The latter is, in essence, a representation of one of the possible solutions in the scheduling problem under consideration. Most traditional scheduling techniques lack this powerful feature. The problem considered in this paper involves a fixed time horizon over which throughput has to be maximized. To demonstrate the performance of the technique, a literature problem was used in which an almost 50% reduction in CPU time was observed when compared to the time point based technique.

Acknowledgements

This work is supported by the South African/Hungarian bilateral agreement on science and technology under Grant Number 2072845.

References

1. E. Kondili, C.C. Pantelides and R.W.H. Sargent, "A general algorithm for short-term scheduling of batch operations. I. MILP formulation", Comp. Chem. Eng., 17(1993) 211
2. X. Zhang and R.W.H. Sargent, "The optimal operation of mixed production facilities—extensions and improvements", Comput. Chem. Eng. 22 (1998) 1287–1295
3. G. Schilling and C. C. Pantelides, "A simple continuous-time process scheduling formulation and a novel solution algorithm", Comput. Chem. Eng. 20 (1996) S1221 -S1226
4. C.A Floudas and X. Lin, "Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review", Comp. Chem. Eng., 28(2004) 2109
5. T. Majozi and F. Friedler, "Maximization of throughput in a multipurpose batch plants under fixed time horizon: S-Graph approach", Ind. Eng. Chem. Res., 45(2006) 6713
6. E.Sanmartí, T. Holczinger, L. Puigjaner and F. Friedler, "Combinatorial framework for effective scheduling of multipurpose batch plants", AIChE J., 48 (2002) 2557