Maximization of Throughput in a Multipurpose Batch Plant under a Fixed Time Horizon: S-graph Approach

Thokozani Majozi[†,‡] and Ferenc Friedler*[,†]

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

This paper presents a graph-theoretic approach for the scheduling of multipurpose batch plants with the objective to maximize economic performance indexes, such as throughput, revenue, and profit over a predefined time horizon. The approach is based on the S-graph framework, which has been previously applied in the scheduling of multipurpose batch plants for the optimization of time-based performance indexes, e.g., minimization of makespan. In contrast to most techniques published in the literature, the approach proposed in this paper does not require any presupposition of the number of time points or manipulation of the time horizon of interest, which renders it continuous in time. The optimization procedure is based on a guided search algorithm that is guaranteed to terminate at a global optimum. Furthermore, the proposed approach exploits structural uniqueness of the problem to improve computational efficiency, which is necessary for industrial-scale problems. Nonintermediate storage (NIS) operational policy is addressed in this paper.

1. Introduction

The general scheduling problem entails the determination of the optimal sequence of events using available resources. The systematic formulation for this problem was initially given by Sparrow et al.[1] as a mixed-integer nonlinear programming (MINLP) problem. In this formulation, the processing time for each batch is formulated as a function of the batch size, and the overall run length of each product over the time horizon of interest is dependent on the number of batches processed to achieve the production requirement. The lack of adequate solution procedures to guarantee global optimality of this formulation at the time stimulated extensive research in this area. The last two decades have been characterized by various mathematical formulations aimed at solving the scheduling problem in both multipurpose and multiproduct batch operations. Most of the published techniques are based on mathematical programming using mixed-interger linear programming (MILP) and MINLP. Fundamentally, these techniques mainly differ in time domain and recipe representation. In terms of time domain representation, they can be broadly categorized into discrete and continuous time formulations. On the other hand, recipe representations are mainly based on State Task Network (STN), which was initially proposed by Kondili et al.[2] This was later extended to Resource Task Network (RTN) by Pantelides.[3] Recently, a State Sequence Network (SSN) representation was proposed by Majozi and Zhu[4] in an attempt to derive smaller mathematical formulations that are solvable within reasonable computer processing unit (CPU) times. A detailed account of all the recent mathematical formulations and the recipe representations on which they are based has been given by Floudas and Lin.[5]

Otherwise, an approach that has been traditionally used is based on a graph representation combined with a branch-and-bound method. These techniques for the case of scheduling, known as edge finding methods,[6−8] have proved to be very effective for solving special types of job-shop scheduling problems. An extensive computational study of the problem was also performed by Applegate and Cooke,[9] in which the authors developed heuristics for finding feasible schedules, cutting planes for obtaining lower bounds, and a specialized branch-and-bound method. Very recently, a new graph representation called S-graph, appropriate for combinatorial algorithms, has been introduced.[10,11] After all process tasks have been represented in a recipe graph, an appropriate search strategy permits the S-graph of the optimal schedule to be generated effectively; i.e., a drastic reduction of computation time can be achieved, compared to mathematical programming solution techniques. In the past, this approach has been only applied to problems involving time-related performance indexes, e.g., minimization of makespan. In this paper, we demonstrate the capabilities of this approach in addressing problems that involve economic performance indexes. Typical examples in this category include maximization of throughput, maximization of revenue, and maximization of profit over a fixed time horizon of interest.

The overall paper is organized as follows. Section 2 presents an elaborate statement of the problem at hand. Sections 3 and 4 give the detailed description of the S-graph framework and optimization strategy, respectively. This is followed by literature examples in Section 5 and a case study in Section 6. Section 7 gives conclusions drawn from the performance of the proposed approach.

2. Problem Statement

The problem addressed in this paper can be stated 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 yields the overall maximum throughput or revenue for all the products involved. In all the problems considered in this paper, no intermediate storage (NIS) exists between consecutive tasks. However, the material can be temporarily stored within the corresponding processing equipment unit until the consecutive equipment unit is available for the next task in the recipe. The equipment units are assumed to be of equal capacities for the same task. Moreover, the amount of material processed within an equipment unit is assumed to be fixed for various

* To whom correspondence should be addressed. Tel: +36 88 424483. Fax: +36 88428275. E-mail: friedler@dcs.vein.hu.

† Department of Computer Science, University of Pannonia.

‡ Department of Chemical Engineering, University of Pretoria.

10.1021/ie0604472 CCC: $33.50
Published on Web 09/06/2006

batches of different products. These conditions do occur frequently in practice.

3. S-graph Framework

Although the detailed description of the mathematical formulation for the S-graph framework has been presented by Sanmartí et al.[11] and Romero et al.,[12] it will also be given in sufficient detail in Section 3.1, to facilitate understanding.

Graph theory has often been used to solve complex problems, including scheduling. However, the scheduling applications have been restricted to the general job-shop scheduling problem in the mechanical industry, where intermediates can be stored between operations (i.e., the UIS policy is assumed). The S-graph framework is a more sophisticated graph representation that was initially designed to solve the NIS case.

3.1. Mathematical Formulation of the S-graph. In an S-graph, two classes of arcs, the so-called recipe arcs and schedule arcs are specified. Therefore, an S-graph is given in the form of $G(N,A_1,A_2)$, where N, A_1, and A_2 denote the sets of nodes, recipe arcs, and schedule arcs, respectively. A non-negative value, $c(i, j)$, that denotes the weight of arc (i, j) is assigned to each arc. In practice, if an arc is established from node i to node j, the task corresponding to node j cannot start its activity earlier than time $c(i, j)$ after the task corresponding to node i has been started. Specific types of S-graphs are identified for a recipe (that is, recipe graph) and for a schedule of all tasks (i.e., schedule graph).

3.1.1. Recipe Graph. A recipe defines the order and types of tasks, the material transfer between them, and the set of plausible equipment units for each task. This type of information should be represented by the graph of a recipe. Let one node be assigned to each task (task node) and one to each product (product node). An arc is established between the nodes of consecutive tasks and from the nodes of tasks generating the products to the corresponding product node, which is the associated weight specified by the processing times of the tasks. If more than one batch of products is to be produced, the task nodes, the product nodes, and the arcs are multiplied appropriately. The resultant graph is called a task network, where N_t and N_p denote the set of its task nodes and product nodes, respectively ($N_t \cap N_p = \emptyset$). This task network can be used as a recipe graph, assuming the incoming arcs of a node express that the inputs of the corresponding task must be available simultaneously.

3.1.2. Schedule Graph. A specific S-graph, termed schedule graph, is introduced to describe a single solution of a scheduling problem; one schedule graph exists for each feasible schedule of the problem. S-graph $G'(N,A_1,A_2)$ is called a schedule graph of recipe graph $G(N,A_1,\emptyset)$ if all tasks represented in the recipe graph have been scheduled by taking equipment-task assignments into account. By an appropriate search strategy, the schedule graph of the optimal schedule can be effectively generated. The formal definition of schedule graph and the axioms that it must satisfy in the NIS case have been presented elsewhere.[11]

3.2. S-graph Representation for Nonintermediate Storage (NIS) Policies. When no intermediate storage is available, an equipment unit is not free after processing a task until the material stored in it has been transferred to the equipment unit assigned to the next task in the recipe. Arc or arcs express these additional constraints that are imposed by the NIS policy. Each arc of a schedule graph that does not belong to its recipe graph is a schedule arc. Let τ_j denote the set of tasks that immediately follow task j according to the recipe. If equipment unit E_i is

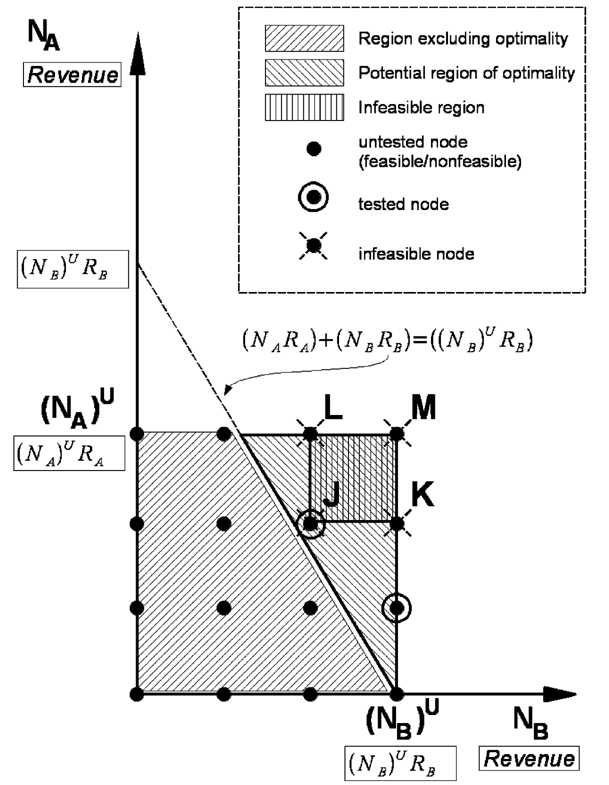

Figure 1. Search space shown as the number of batches of A versus that of B.

assigned to task j after completion of task k, then a zero-weighted arc (or an arc whose weight is equal to the length of the changeover time, if applicable) is established from each element of τ_j to k. Here, this type of arc is called an *NIS schedule arc*. For the UIS operational policy, this representation is slightly modified.[12] However, note that a feasible schedule for the UIS transfer policy may be infeasible for the NIS case. The added advantage of the S-graph is that this infeasibility can be readily detected by finding a directed cycle in the graph.[11] Moreover, the test for the existence of a feasible solution in the production of a given number of batches of products under a fixed time horizon is very effective in the S-graph framework. This test is based on the longest path algorithm of directed graphs. Also, the S-graph framework allows results from the previous feasibility test to be readily exploited in the current test, which is concomitant with improved solution times.

4. Optimization Strategy

The optimization strategy is based on a guided search within a region defined by the structure of the problem. The efficiency of the search derives from two main reasons. First, redundancy is inherently eliminated, becasue, at each point in the search, a node with a unique combination of batches of products is explored. Each node can either involve batches of the same product or different products within a set of products considered for production over the time horizon of interest. Second, the region comprised of nodes that bear no opportunity for optimality is identified and eliminated a priori from the search, thereby significantly reducing the search region. As a result, the solution is obtained much quicker than it would be in an exhaustive search. At each node of the search, a partial problem that involves a fixed number of batches of each product is solved using an equipment-oriented S-graph framework.[11] In each partial problem, feasibility is tested by the generation of a

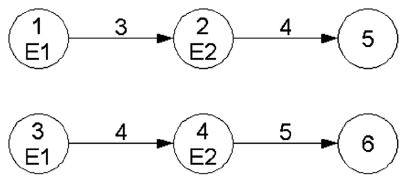

Figure 2. Recipe graphs for products A and B of the illustrative example.

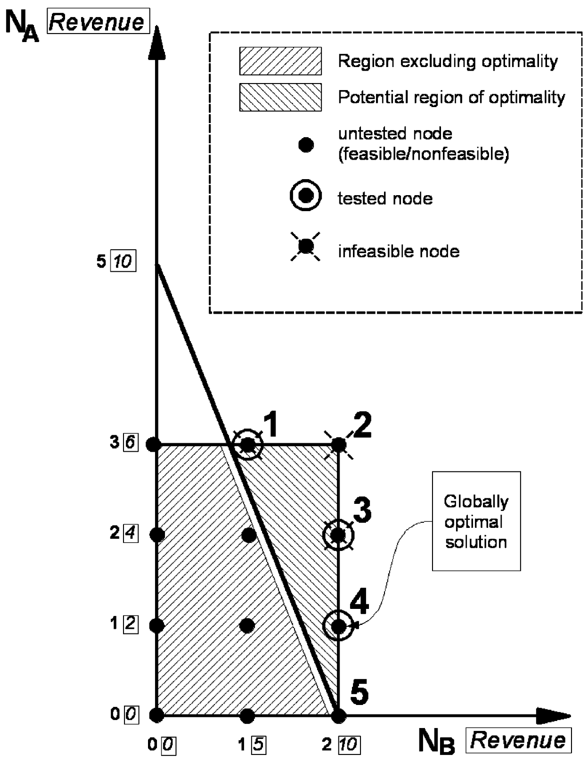

Figure 3. Search space shown as the number of batches of A versus that of B.

feasible schedule graph corresponding to the fixed number of batches of each product at a given node. The feasibility of the schedule graph is based on feasible makespan to produce the set number of batches at a given node. Therefore, it is evident that, for a feasible makespan at each node (i.e., makespan less than the predefined time horizon), the objective function is fixed.

To facilitate understanding of the optimization strategy, an illustration is shown in Figure 1. In this case, two products, A and B, are considered for production over the time horizon of interest. The vertical and horizontal axes represent the number of batches of A (N_A) and B (N_B), respectively. Given the economic contributions of products A (R_A) and B (R_B), the revenue corresponding to each node, i.e., ($N_AR_A + N_BR_B$), is readily fixed. It is evident that the origin or intersection of the vertical and the horizontal axes corresponds to the root node of a typical branch-and-bound search tree. This intersection represents zero batches of each of the products considered for production over the time horizon of interest. The vertical and horizontal boundaries of the region are determined by the maximum number of batches of each of the products considered for production over the time horizon of interest. Naturally, the search region only involves the nodes and not the overall shaded area. The maximum number of batches of the products have been represented by $(N_A)^U$ and $(N_B)^U$ for products A and B, respectively. To determine the maximum number of batches of each product, a feasibility test can be conducted along the boundary of the search region. In the illustration shown in Figure 1, it is assumed that product B has a higher economic contribution than product A, i.e., $R_A < R_B$. Therefore, even

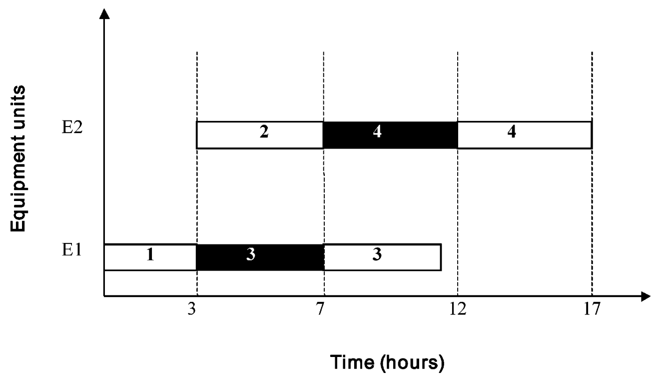

Figure 4. Schedule for the global optimum of the illustrative example.

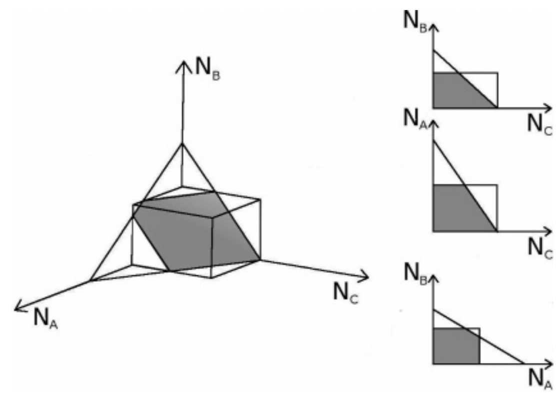

Figure 5. Search region for a three-dimensional space.

though the maximum number of batches of product B that can be produced over the time horizon of interest is less than that for product A, the revenue-producing $(N_B)^U$ batches of product B is higher than that of $(N_A)^U$ batches of product A. In the case of two products with equal revenues, this choice is irrelevant.

If a straight line is drawn diagonally from the node marked by $(N_B)^U$ on the horizontal axis to the vertical axis such that at each point along the line the overall revenue is the same, i.e., $(N_AR_A + N_BR_B) = (N_B)^UR_B$, a boundary with interesting properties emerges. Any node that lies below this diagonal line has lower revenue than the revenue on the line. On the other hand, any feasible node that lies above the diagonal line has higher revenue than the revenue on the line. Therefore, the global optimum should lie either on the diagonal line or above it. This implies that the diagonal line forms the boundary between a region excluding optimality and the region constituting the optimal point. However, note that, unlike the nodes below the diagonal line, whose feasibility is unnecessary because they exclude any potential optimality, the feasibility of nodes above the line must be tested. Further properties of the optimality region can be exploited to minimize the exhaustiveness of the feasibility test. For example, if node J of Figure 1 is infeasible, i.e., the makespan of the corresponding scheduling problem is longer than the time horizon of interest, then nodes K, L, and M will also be infeasible. Node K is infeasible, because, if the time horizon cannot allow for the production of three batches of B and four batches of A, as found in node J, then it certainly cannot allow for the production of four batches of B and four batches of A, as found in node K. The infeasibility of nodes L and M follows a similar analysis. Therefore, the region bounded by nodes J, K, L, and M can be excluded from the search without a loss of optimality.

This analysis of the search region ensures that much fewer nodes are explored than those in the exhaustive search. In essence, only the subset of the nodes on and above the diagonal line needs to be tested for global optimality.

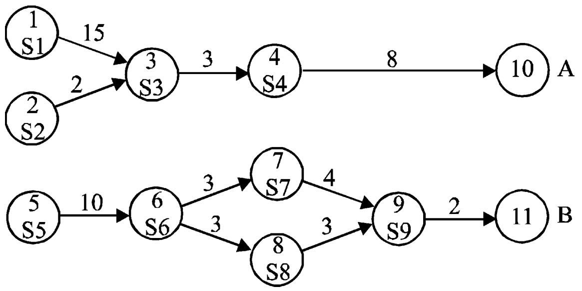

Figure 6. Recipe graph of example 1.

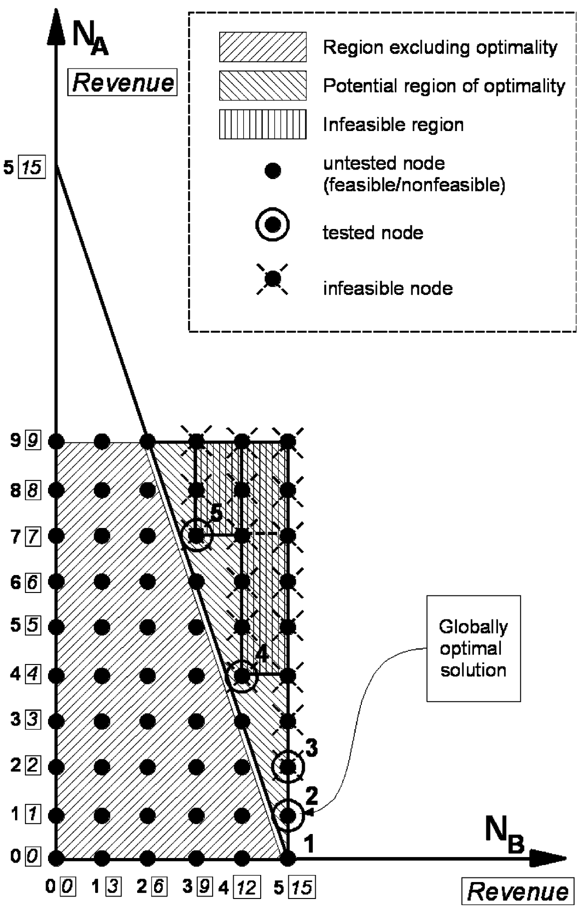

Figure 7. Search space shown as the number of batches of A versus that of B for example 1.

4.1. Illustrative Example. The recipe graph for the illustrative example is shown in Figure 2. In this example, the objective is to maximize revenue for products A and B over a time horizon of 18 h. Product A has a profit contribution of 2 cost units (cu)/batch, whereas product B has a profit contribution of 5 cu/batch. Both products share equipment units E1 and E2, and each involves two stages for a complete batch. Product A requires 3 h in E1 and 4 h in E2, whereas product B requires 4 h in E1 and 5 h in E2.

Figure 3 shows the search region for the problem considered. A maximum of two batches of product B and three batches of product A can be produced over the 18-h time horizon, hence the horizontal and vertical boundaries of the search region. This corresponds to the revenue of 10 cu for product B and 6 cu for product A. Construction of the boundary line as described in Section 4 shows that only five nodes lie in the region of optimality. These are the only nodes that need to be tested for feasibility and optimality. Node 1, which corresponds to the production of one batch of product B and three batches of product A, is infeasible, because it requires a makespan longer than the time horizon of interest (i.e., 18 h). Consequently, the infeasibility of node 2 can be inferred from the diagram in Figure 3 without any test, as elaborated for node K in Figure 1.

Node 3 is infeasible, because, to produce two batches of product B and two batches of product A, a longer time than the 18 h time horizon of interest is required. Node 4 is feasible and has a revenue of 12 cu, which is higher than the revenue of node 5; hence, it qualifies as a global optimum. Assuming that the maximum number of batches of the individual products are given, this algorithm tests only four nodes among all the nodes in the search region in obtaining the global optimum. The schedule corresponding to this solution is shown in Figure 4.

4.2. Extension to Multiple Products. Contrary to the previous cases, in the case of more than two products that need to be considered for production over the time horizon of interest, the search region is multidimensional rather than two-dimensional. However, the same analysis holds. Consider three products—A, B, and C—that are considered for production within a time horizon of interest. It is evident that the region shown in Figure 5 captures all possibilities of producing a mix of these three products. As mentioned previously, the boundaries of the search region are defined by the maximum number of batches of each of the products to be produced. The shaded region is analogous to region 1 in Figure 1 (i.e., region excluding optimality), and the clear region is analogous to region 2 in Figure 1 (i.e., potential region for optimality). Also shown in Figure 5 is the projection of each of the three planes on a two-dimensional surface to facilitate understanding.

Table 1. Scheduling Data for Example 2

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 (task 3)	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

Table 2. Stoichiometric Data for Example 2

	Stoichiometric Data	
state	output (ton/ton)	product (ton/ton)
raw 3	0.20	
raw 4	0.25	
raw 1	0.35	
raw 2	0.20	
effluent		0.7
waste		1

Table 3. Scheduling Data for the Case Study

		Production Time in Mixing Vessel (h)			
product	economic contribution (cu/batch)	V1	V2	V3	V4
cream_1	2	10	5	N/A	5
cream_2	3	12	10	7	N/A
conditioner	1	N/A	N/A	12	N/A
shampoo	3.5	N/A	8	13	N/A
lotion	1.5	10	6	N/A	9

5. Literature Examples

In this section, two literature examples are presented to demonstrate the performance of the proposed technique. The

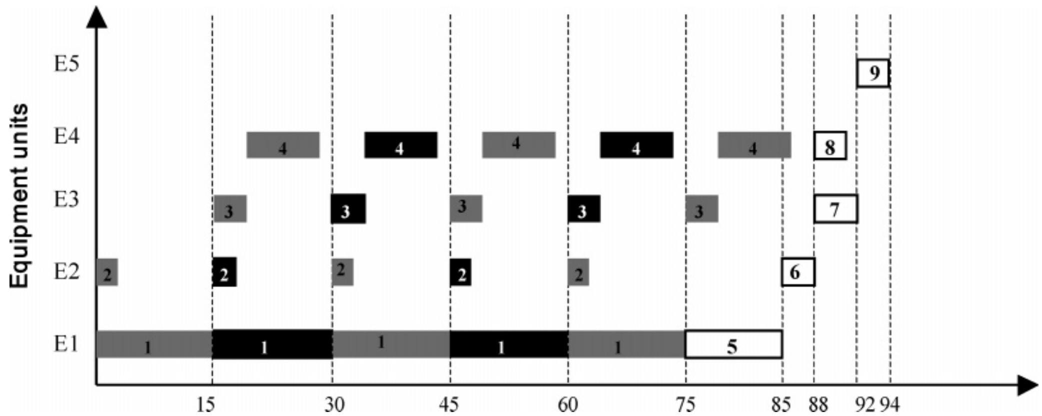

Figure 8. Schedule for the global optimum of literature example 1.

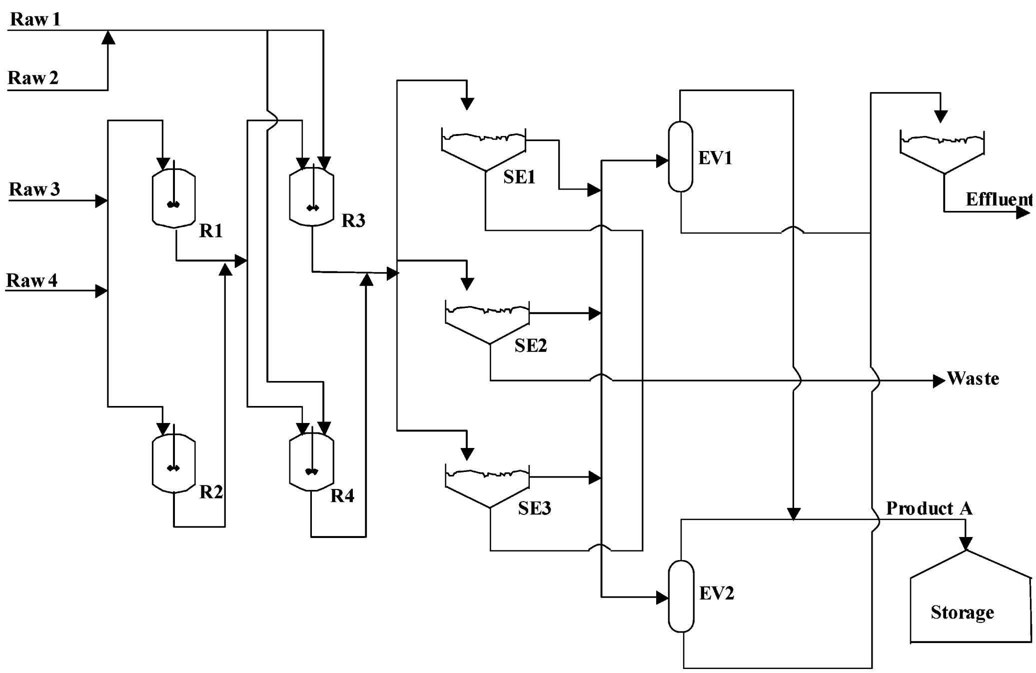

Figure 9. Flowsheet for literature example 2.

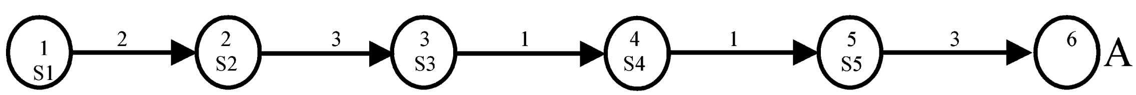

Figure 10. Recipe graph for literature example 2.

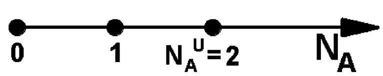

Figure 11. Search region for literature example 2.

first example is taken from Holczinger et al.,[13] and the second is taken from Majozi and Zhu.[4]

5.1. Example 1. Two products (products A and B) are to be produced, according to the recipe given in Figure 6. Si (i = 1, 2, ..., 9) denotes the set of those equipment units that can perform task i. The sets are specified as S1 = {E1}, S2 = {E2}, S3 = {E3}, S4 = {E4}, S5 = {E1}, S6 = {E2}, S7 = {E3}, S8 = {E4}, and S9 = {E5}. Product A has a revenue of 3 cu/batch, whereas product B has a revenue of 1 cu/batch. The objective is to maximize revenue over a time horizon of 100 h.

The search region, in accordance with the proposed algorithm, is depicted in Figure 7. The search region consists of 60 nodes, 19 of which lie on or above the boundary line, i.e., the optimality region. Applying the analysis presented in Section 4, it is evident that only 5 of the 19 nodes need to be tested for feasibility and optimality. These nodes are numbered accordingly in Figure 7. Node 2 entails the global optimal solution of 16 cu, which corresponds to five batches of product A and one batch of product B. Figure 8 shows the schedule corresponding to a global optimum of 16 cu. The complete schedule involves five batches of product A and one batch of product B. The optimal makespan corresponding to the complete schedule is 94 h. In Figure 8, ti_j refers to batch j of task i.

5.2. Example 2. This example is for an agrochemical process for the production of an herbicide. The flowsheet for the process is shown in Figure 9. The process that is considered consists of five consecutive steps. The first step involves a reaction that 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 occur. The first

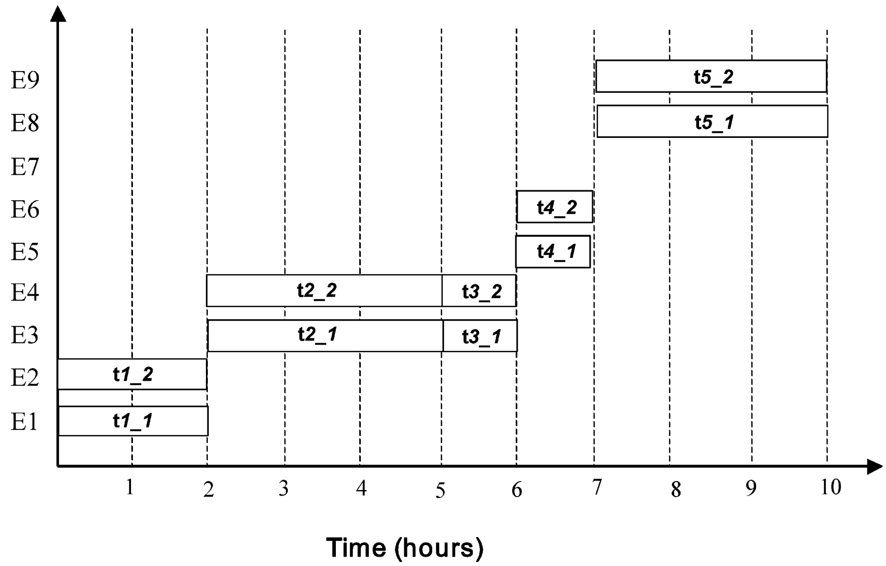

Figure 12. Schedule for the global optimum of example 2.

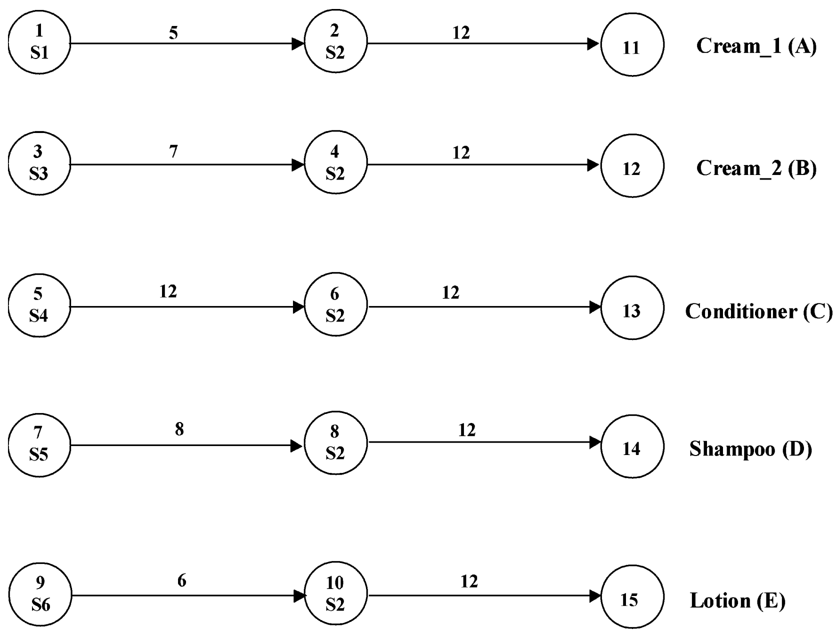

Figure 13. Recipe graph for the case study.

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 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 discarded 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 placed into storage.

Figure 10 is the recipe graph for example 2, where S1 = {R1, R2}, S2 = {R3, R4}, S3 = {R3, R4}, S4 = {SE1, SE2, SE3}, and S5 = {EV1, EV2}. Table 1 shows scheduling data, whereas Table 2 shows stoichiometric data.

The stoichiometric data shown in Table 2 is included to perform material balances in each unit operation. The second column of the stoichiometric data shows the amount of raw material required (tons) per unit mass (tons) of the overall output, i.e., effluent + waste + product A. The third column shows the ratio of each byproduct (waste and effluent) to product A in ton/ton product. The objective function is the maximization of product A output over a time horizon of 10 h.

Because this example involves only one product, the search region is one-dimensional, as shown in Figure 11. Figure 12 depicts the schedule that corresponds to the global optimum. In Figure 12, t*i*_*j* refers to batch *j* of task *i*. Only two batches can be produced over the time horizon of interest (i.e., 10 h). The global optimum for throughput is 7.42 tons of product A. This is due to the fact that, for every batch, ~3.71 tons of product A are produced, as stipulated by the stoichiometry given in Table 1.

Although this paper is not intended to compare the performance of the proposed approach with other published approaches, we note that the solution was obtained in 0.04 CPU s (CPU seconds) in a 1.2 GHz Pentium M processor. On the other hand, using a tested recent MILP mathematical formulation by Majozi and Zhu,[4] the same solution was obtained in 0.13 CPU s in the same computer. Moreover, the MILP formulation required the stipula-

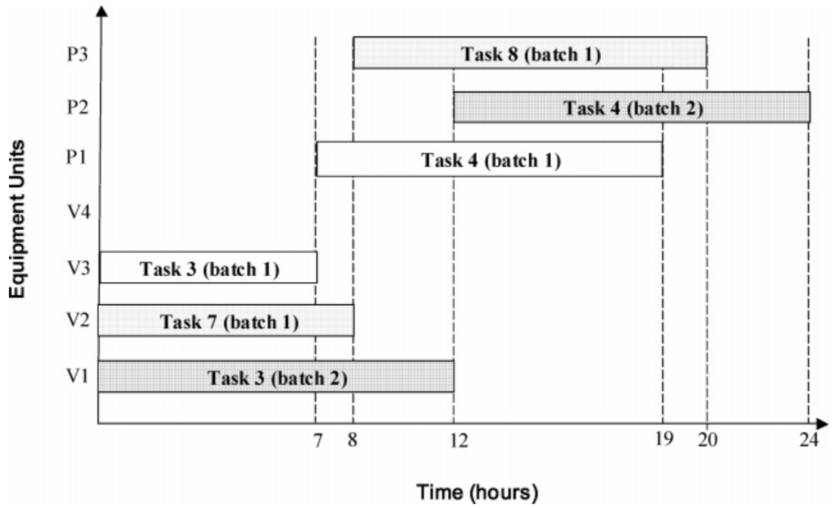

Figure 14. Schedule for global optimum of the case study.

Table 4. Search Space of the Case Study: Five Products

node number	Products A	B	C	D	E	Rev (cu)	node property[a]	node number	Products A	B	C	D	E	Rev (cu)	node property[a]	node number	Products A	B	C	D	E	Rev (cu)	node property[a]
1	0	0	0	0	0	0	NO	44	3	0	1	0	1	8.5	NO	87	3	1	0	1	0	12.5	IN
2	0	0	1	0	0	1	NO	45	1	1	0	1	0	8.5	NO	88	1	2	0	1	1	13	IN
3	0	0	0	0	1	1.5	NO	46	2	1	0	0	1	8.5	NO	89	3	2	1	0	0	13	IN
4	1	0	0	0	0	2	NO	47	3	1	0	0	0	9	IT	90	2	1	1	1	1	13	IN
5	0	0	1	0	1	2.5	NO	48	1	2	1	0	0	9	IT	91	2	3	0	0	0	13	IN
6	0	1	0	0	0	3	NO	49	0	1	1	1	1	9	IT	92	0	3	1	1	0	13.5	IN
7	1	0	1	0	0	3	NO	50	0	3	0	0	0	9	FT	93	1	3	1	0	1	13.5	IN
8	1	0	0	0	1	3.5	NO	51	2	0	0	1	1	9	IT	94	3	1	1	1	0	13.5	IN
9	0	0	0	1	0	3.5	NO	**52**	**0**	**2**	**0**	**1**	**0**	**9.5**	**FT**	95	3	2	0	0	1	13.5	IN
10	0	1	1	0	0	4	NO	53	1	2	0	0	1	9.5	IT	96	2	2	0	1	0	13.5	IN
11	2	0	0	0	0	4	NO	54	1	1	1	1	0	9.5	IT	97	2	3	1	0	0	14	IN
12	1	0	1	0	1	4.5	NO	55	2	1	1	0	1	9.5	IT	98	3	1	0	1	1	14	IN
13	0	1	0	0	1	4.5	NO	56	3	0	0	1	0	9.5	IT	99	0	3	0	1	1	14	IN
14	0	0	1	1	0	4.5	NO	57	3	1	1	0	0	10	IN	100	1	2	1	1	1	14	IN
15	1	1	0	0	0	5	NO	58	2	2	0	0	0	10	IN	101	1	3	0	1	0	14.5	IN
16	0	0	0	1	1	5	NO	59	2	0	1	1	1	10	IN	102	2	3	0	0	1	14.5	IN
17	2	0	1	0	0	5	NO	60	1	1	0	1	1	10	IT	103	3	2	1	0	1	14.5	IN
18	1	0	0	1	0	5.5	NO	61	0	3	1	0	0	10	IN	104	2	2	1	1	0	14.5	IN
19	0	1	1	0	1	5.5	NO	62	1	2	1	0	1	10.5	IN	105	3	1	1	1	1	15	IN
20	2	0	0	0	1	5.5	NO	63	3	1	0	0	1	10.5	IN	106	0	3	1	1	1	15	IN
21	3	0	0	0	0	6	NO	64	2	1	0	1	0	10.5	IT	107	2	2	0	1	1	15	IN
22	1	1	1	0	0	6	NO	65	0	2	1	1	0	10.5	IT	108	3	3	0	0	0	15	IN
23	0	0	1	1	1	6	NO	66	0	3	0	0	1	10.5	IT	109	1	3	1	1	0	15.5	IN
24	0	2	0	0	0	6	NO	67	3	0	1	1	0	10.5	IT	110	2	3	1	0	1	15.5	IN
25	1	1	0	0	1	6.5	NO	68	2	2	1	0	0	11	IN	111	3	2	0	1	0	15.5	IN
26	1	0	1	1	0	6.5	NO	69	3	0	0	1	1	11	IN	112	3	3	1	0	0	16	IN
27	0	1	0	1	0	6.5	NO	70	1	3	0	0	0	11	IT	113	2	2	1	1	1	16	IN
28	2	0	1	0	1	6.5	NO	71	1	1	1	1	1	11	IN	114	1	3	0	1	1	16	IN
29	1	0	0	1	1	7	NO	72	0	2	0	1	1	11	IT	115	2	3	0	1	0	16.5	IN
30	0	2	1	0	0	7	NO	73	2	2	0	0	1	11.5	IN	116	3	3	0	0	1	16.5	IN
31	3	0	1	0	0	7	NO	74	2	1	1	1	0	11.5	IN	117	3	2	1	1	0	16.5	IN
32	2	1	0	0	0	7	NO	75	3	1	1	0	1	11.5	IN	118	1	3	1	1	1	17	IN
33	0	2	0	0	1	7.5	NO	76	0	3	1	0	1	11.5	IN	119	3	2	0	1	1	17	IN
34	0	1	1	1	0	7.5	NO	77	1	2	0	1	0	11.5	IT	120	3	3	1	0	1	17.5	IN
35	2	0	0	1	0	7.5	NO	78	0	2	1	1	1	12	IN	121	2	3	1	1	0	17.5	IN
36	1	1	1	0	1	7.5	NO	79	3	0	1	1	1	12	IN	122	3	2	1	1	1	18	IN
37	3	0	0	0	1	7.5	NO	80	3	2	0	0	0	12	IN	123	2	3	0	1	1	18	IN
38	1	2	0	0	0	8	NO	81	1	3	1	0	0	12	IN	124	3	3	0	1	0	18.5	IN
39	0	1	0	1	1	8	NO	82	2	1	0	1	1	12	IN	125	2	3	1	1	1	19	IN
40	1	0	1	1	1	8	NO	83	2	2	1	0	1	12.5	IN	126	3	3	1	1	0	19.5	IN
41	2	1	1	0	0	8	NO	84	0	3	0	1	0	12.5	IT	127	3	3	0	1	1	20	IN
42	0	2	1	0	1	8.5	NO	85	1	3	0	0	1	12.5	IN	128	3	3	1	1	1	21	IN
43	2	0	1	1	0	8.5	NO	86	1	2	1	1	0	12.5	IN								

[a] NO, not optimal; IT, infeasible tested; IN, infeasible not tested; and FT, feasible tested.

tion of the number of time points a priori, thereby discretizing the time horizon into uneven time intervals. This is consistent with most recent MILP formulations, which have been referenced as "continuous time" in published literature. The mathematical formulation involved 66 binary variables, 481 continuous variables, and 809 equations. A detailed performance comparison of the method proposed in this paper with other published methods will be the focus of another publication.

6. Case Study

The case study is taken from a multinational pharmaceuticals facility that produces lotions, shampoos, conditioners, and various creams in South Africa. All the products involve mixing and packaging. In the physical facility, there is intermediate storage between mixing vessels and the packing lines. However, this has been deliberately omitted in this presentation, to cast the problem as an NIS case. Mixing is conducted in four mixing vessels, i.e., V1, V2, V3, and V4, and packaging is conducted in packing lines P1, P2, and P3. Because of different designs of the stirrers in mixing vessels, mixing times vary, according to the vessel used. The capacity of each mixing vessel is ∼3 tons. The duration of each type of product in mixing vessels is shown in Table 3. This table shows the relative economic contributions for each of the products. It is evident that the shampoos have the highest economic contribution. The packing duration for each product is 12 h, irrespective of the packing line. The objective in this case study is to maximize the overall economic contribution over a time horizon of 24 h. The recipe graph for the products manufactured in the chosen facility is shown in Figure 13, where S1 = {V1,V2,V4}, S2 = {P1,P2,P3}, S3 = {V1,V2,V3}, S4 = {V3}, S5 = {V2,V3}, and S6 = S1.

The problem considered requires a similar analysis to be conducted on a five-dimensional search region. Although this can be easily represented using an algorithm, it cannot be readily captured in two dimensions graphically. Therefore, a tabular representation of the five-dimensional search region was adopted, to demonstrate the applicability of the algorithm in five dimensions. This is shown in Table 4. Each row in Table 4 represents a node in the search region. A row is composed of consecutive columns of the five products and a corresponding objective value. For example, the first row involves zero production of any of the products (and, hence, zero revenue). This row corresponds to the origin of the search region, as introduced in Section 4.

The maximum numbers of batches of products A, B, C, D, and E that can be produced over a time horizon of 24 h are three, three, one, one, and one, respectively. These correspond to the revenues of 6, 9, 1, 3.5, and 1.5 cu, respectively. Following the algorithm presented in Section 3.2, only the nodes with a revenue of 9 cu or more need to be explored for global optimality. All the nodes with lower revenues can safely be omitted from the search without any loss of optimality; these nodes correspond to lines 1−46 of Table 4.

Nodes 47−51 correspond to the objective value of 9 cu. Of these nodes, only node 50 is feasible, whereas the other four are infeasible. The next higher objective is 9.5 cu, which corresponds to nodes 52−56. Also, with the exception of node 52, all of the other four nodes are infeasible.

To eliminate the other nodes that lie in the search region, but are infeasible, a simple analysis similar to the elimination of K, L, and M in Figure 1 is adopted. Consider node 67 in Table 4. This node involves three batches of A, one batch of C, and one batch of D, which requires a longer time than the time horizon of interest (24 h). This node is infeasible and should be eliminated from the search. Therefore, any node that entails at least three batches of A, one batch of C, and one batch of D will be considered infeasible without the need for a test. Nodes 79, 94, 105, 117, 122, 126, and 128 do not require a test to prove they are infeasible. Following a similar approach, all the nodes below node 57 were considered infeasible, either through a test (IT) or problem analysis without test (IN). Only 20 of the 128 nodes were tested to arrive at the globally optimal solution. The global optimal solution corresponds to node 52, i.e., two batches of B and one batch of D (highlighted in bold type in Table 4), with a revenue of 9.5 cu. The schedule that corresponds to the global optimum is shown in Figure 14.

7. Conclusions

An efficient search algorithm for the globally optimal throughput, revenue, or profit over a predefined time horizon in multipurpose batch plants has been presented. To demonstrate its performance, two literature examples and a case study from a real-life multipurpose batch facility have also been presented. In addition to guaranteed global optimality, the added advantage of the presented algorithm over its mathematical programming counterparts is that it does not require any manipulation of the time horizon of interest. Presupposition of time points to discretize the time horizon into equal or unequal time lengths is unnecessary. Therefore, it qualifies as a true continuous time methodology. The added advantages of the S-graph framework are its effectiveness in proving feasibility or infeasibility and the possible exploitation of the previous node results in the current search. Only the non-intermediate storage (NIS) is addressed in this paper. Other operational philosophies will be the subject of another publication.

Acknowledgment

The authors would like to thank Dr. Tibor Holczinger, Mr. Robert Adonyi, and Mr. Mate Hegyhati for their valuable suggestions regarding the content of the paper.

Literature Cited

(1) Sparrow, R. E.; Forder, G. J.; Rippin, D. W. T. The Choice of Equipment Sizes for Multiproduct Batch Plants—Heuristics vs Branch and Bound. *Ind. Eng. Chem. Process. Des. Dev.* **1975**, *14*, 197.

(2) Kondili, E.; Pantelides, C. C.; Sargent, R. W. H. A General Algorithm for Short-Term Scheduling of Batch Operations. I. MILP Formulation. *Comput. Chem. Eng.* **1993**, *17*, 211.

(3) Pantelides, C. C. Unified Frameworks for Optimal Process Planning and Scheduling. In *Proceedings of the Second International Conference on Foundations of Computer-Aided Process Operations*; Rippin, D. W. T., Hale, J. C., Davis, J., Eds.; CACHE Corp.: Austin, TX, 1993; p 253.

(4) Majozi, T.; Zhu, X. X. A Novel Continuous Time MILP Formulation for Multipurpose Batch Plants. 1. Short-Term Scheduling. *Ind. Eng. Chem. Res.* **2001**, *40*, 5935.

(5) Floudas, C. A.; Lin, X. Continuous-Time Versus Discrete-Time Approaches for Scheduling of Chemical Processes: A Review. *Comput. Chem. Eng.* **2004**, *28*, 2109.

(6) Adams, J.; Balas, E.; Zawack, D. The Shifting Bottleneck Procedure for Job Shop Scheduling. *Manage. Sci.* **1988**, *34*, 391.

(7) Carlier, J.; Pinson, E. An Algorithm for Solving the Job-Shop Problem. *Manage. Sci.* **1989**, *35*, 164.

(8) Cormen, T. H.; Leiserson, C. E.; Rivest, R. L. *Introduction to Algorithm*; MIT Press: Cambridge, MA, 1997.

(9) Applegate, D.; Cooke, W. A. Computational Study of the Job-Shop Scheduling Problem. *ORSA J. Comput.* **1991**, *3*, 149.

(10) Sanmartí, E.; Friedler, F.; Puigjaner, L. Combinatorial Technique for ShortOLINIT-Term Scheduling of Multipurpose Batch Plants Based on Schedule-Graph Representation. *Comput. Chem. Eng.* **1998**, *22*, S847.

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

(12) Romero, J.; Puigjaner, L.; Holczinger, T.; Friedler, F. Scheduling Intermediate Storage Multipurpose Batch Plants Using the S-Graph. *AIChE J.* **2004**, *50*, 403.

(13) Holczinger, T.; Romero, J.; Puigjaner, L.; Friedler, F. Scheduling of Multipurpose of Batch Processes with Multiple Batches of the Products. *Hung. J. Ind. Chem.* **2002**, *30*, 305.

Received for review April 10, 2006
Revised manuscript received July 29, 2006
Accepted August 7, 2006

IE0604472