I&EC research
Industrial & Engineering Chemistry Research
ARTICLE
pubs.acs.org/IECR

Combinatorial Algorithms of the S-Graph Framework for Batch Scheduling

Máté Hegyháti and Ferenc Friedler*

Department of Comupter Science and Systems Technology, University of Pannonia, 10 Egyetem u., Veszprém, H-8200, Hungary

ABSTRACT: Methods for solving batch process scheduling problems have gone through a vast development in the last 2 decades. Most of the published approaches are based on a mixed integer programming formulation. Since the difficulty of scheduling is originated from its combinatorial nature, graphs and combinatorial algorithms are more adequate to represent and solve the problem. Although, combinatorial algorithms and data structures have an enormous literature, these algorithms can not be directly applied to scheduling and further elaboration is needed. In the present work, the combinatorial nature of batch scheduling problems is analyzed. Several combinatorial algorithms are listed that can be considered for the scheduling of batch processes. Their proper adaptation is illustrated via the S-graph framework, in which the main emphasis lies on the combinatorial tools. Furthermore, Place Petri Nets and Timed Automata are also briefly described. An S-graph algorithm has been extensively compared with well-known MILP formulations.

COMBINATORIAL NATURE OF BATCH PROCESS SCHEDULING

Scheduling is a key problem in the operation of batch plants. The industry generates a wide range of batch scheduling problems, where the goal in general is to allocate the tasks of the process to the available equipment units in the most favorable way.[1,2] An ordinary batch scheduling problem is given by the master recipe of the process, the objective, and the intermediate storage policy. The most common objectives are the minimization of the whole processing time, i.e., makespan, or the maximization of the throughput or profit over a fixed time horizon. According to different problems, the storage policy can vary between unlimited intermediate storage (UIS), finite intermediate storage (FIS), common intermediate storage (CIS), nonintermediate storage (NIS), and zero-wait (ZW).[3,4] Problem specification may include further parameters, e.g., transfer times, changeover times, or variable processing times.[5] The recipe defines the set of products to be produced, the network of tasks to produce the desired products, the available equipment units, processing times, stoichiometric data, etc. In the case of a complex recipe, i.e., when the process does not have sequential characteristics, the unambiguous representation of the network of the tasks is not evident.[6] In batch process scheduling, mostly directed graphs, e.g., State-Task-Network (STN),[7] Resource-Task-Network (RTN),[8] State-Sequence-Network (SSN),[9] S-graph,[10] Timed Place Petri Net (TPPN),[11] or Priced Timed Automata (PTA)[12] are applied for this purpose. Despite the wide range of available graph representations, most of the approaches consider them only as a graphical representation and not as the model for the optimization.

The combinatorial nature of batch scheduling problems derives from the two main decisions to be made during the optimization process: (i) which processing unit is assigned to a task (if more than one is available) and (ii) what is the order of the tasks to be performed in an equipment unit. Moreover, if the objective is to maximize the throughput, the optimal number of batches has to be also identified, which is an additional computational issue.

Even though the major decisions are made in discrete space, the problems may involve decisions on continuous variables, e.g., batch sizing, that can usually be handled with an LP model, which requires much less computational effort compared to the combinatorial part of the problem.

COMBINATORIAL ALGORITHMS

Combinatorial algorithms that operate on graphs and sets rather than continuous variables or functions can usually address problems with discrete decisions in a straightforward way, which is often beneficial in terms of the computational efforts. This type of algorithms has an enormous literature,[13] with an excessive introduction to the basic algorithms is given by Cormen et al.[14] The algorithms and data structures presented in this book are usually not directly adaptable to real life problems, but they can be applied as subroutines in an algorithm that has been constructed to meet the specific requirements. Some of the combinatorial algorithms also appearing in batch process scheduling are shortly described below.

Cycle Detection Algorithm. For a given directed graph, the existence of a cycle among the vertices can be decided by the so-called colored depth-first-search algorithm. The algorithm is based on the well-known depth first search originally developed by Tarjan;[15] however, it also assigns three different colors to each vertex, denoting whether the vertex is (a) undiscovered, (b) discovered but not finished, or (c) finished. With the appropriate data structures, the algorithm can detect a cycle or prove its nonexistence in as many steps as the sum of the number of the vertices and arcs.

Special Issue: Puigjaner Issue

Received: August 27, 2010
Accepted: March 8, 2011
Revised: February 26, 2011
Published: April 27, 2011

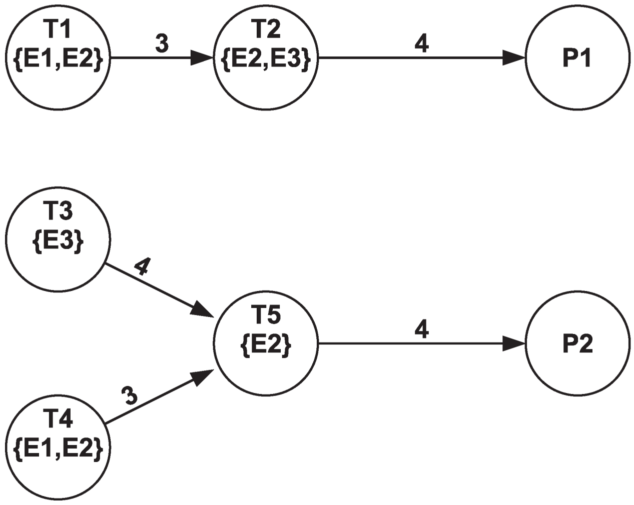

Figure 1. S-graph corresponding to the recipe.

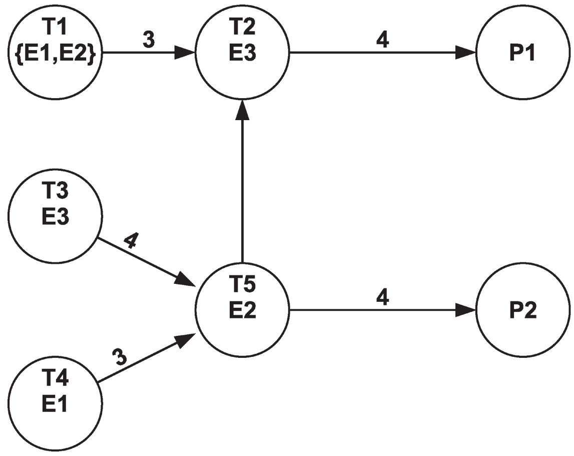

Figure 2. Partial schedule represented by an S-graph.

Longest Path Algorithm. If the arcs of a directed graph have weights, it is a common aim to find a path, in which the sum of the weights is maximal. The problem is hard to solve in its general form; however, a slightly modified version of Dijkstra's algorithm[14] can efficiently return the longest path, if the graph is known to be acyclic.

Branch-and-Bound Algorithm. When the optima is to be found in a given set, branch-and-bound algorithms[16] usually reduce the computational needs compared to an exhaustive enumeration. A branch-and-bound algorithm consists of two functions.[17] The branching function splits a set of candidate solutions into smaller subsets until singleton sets are achieved; thus, the solution of the original problem is decomposed to the solution of smaller subproblems. The bounding function provides a lower bound for the candidates in a certain subset, assuming that the objective is to be minimized. If the bound is worse than the best solution found so far, the corresponding subset is pruned from the set of open subproblems in the proven absence of optimal solution. The efficacy of this type of algorithms strongly depends on the selection of these functions and the selection strategy of unexamined subproblems.

Further Combinatorial Algorithms. Specific combinatorial algorithms are widely applied for scheduling problems of other areas.[18,19] Moreover, there is a long list of additional combinatorial algorithms that can be applied for solving specific problems or for acceleration of the search. To mention a few of these algorithms: the so-called Hungarian method[20] for the assignment problem,[21] Johnson's algorithm[22] for flow-shop problems with two machines, or Jackson's algorithm[23] for job-shop problems with two machines.

■ COMBINATORIAL ALGORITHMS FOR BATCH PROCESS SCHEDULING

The idea of applying combinatorial algorithms for the scheduling of batch processes arised at a scientific discussion initiated by professor Luis Puigjaner, which resulted in the establishment of the S-graph framework appearing in 1998. Since then, several Ph.D. works further developing the framework have been prepared in both Barcelona[24−26] and Veszprem.[27,28]

At its birth, the S-graph framework has been presented to solve makespan minimization problems with NIS policy, fixed processing time, and batch sizes. The originality of this approach was its new, graph-based mathematical model and a problem specific solution procedure. In the following subsections, the application of the previously mentioned combinatorial algorithms for batch process scheduling is illustrated via the S-graph framework.

Mathematical Model: The S-Graph. The S-graph framework employs a special directed graph for the optimization, called the S-graph.[10] The problem is represented by an S-graph, where a vertex is assigned to each task and product of the process. The precedence rules defined by the recipe are expressed by the so-called recipe-arcs, whose weight is the processing time of the task represented by the starting node of the arc.

The S-graph representing the recipe of an example is given in Figure 1. The process consists of two products (P1 and P2) that are produced through five processing steps (T1−T5). Product P1 can be performed by two sequential operations T1 and T2, which have processing times 3 and 4, respectively. For the sake of simplicity, in this illustrative example, the processing times of the tasks do not depend on the selection of the plausible equipment units. The first task of the first product can be performed either in equipment unit E1 or E2. The second task has also two plausible units, namely, E2 and E3. The production of the second product (P2) has two initiative tasks, T3 and T4, which provide different inputs for task T5 that produces the final product. Task T4 is identical to task T1, and tasks T3 and T5 can be performed in 4 h by the units E3 and E2, respectively. In this example one batch is to be produced from both of the products. In the case of multiple batches, the graph corresponding to the product has to be copied to construct the S-graph representing the recipe. Note, that recipe-arcs represent timing precedences, i.e., if a recipe-arc leads from task Ti to Tj with the weight of PTj, the execution of task Tj has to start at least PTi, time units later than the execution of task Ti.

During the optimization procedure, the discrete decisions described in the first section are made and the graph is modified accordingly. If a unit is assigned to a task, the set of plausible units is replaced by that single unit. Moreover, the ordering of tasks assigned to the same unit is expressed by zero-weighted arcs, so-called schedule-arcs. In Figure 2, a partial schedule is represented by an S-graph. Unit E1 has been assigned to task T4; moreover, it has been decided that task T2 will be performed in unit E3 right after it has finished performing T3 and loaded its output to unit E2 for task T5. This latter decision is expressed by the zero-weighted schedule-arc leading from task T5 to task T2, since E3 cannot undertake task T2 until it has been emptied.

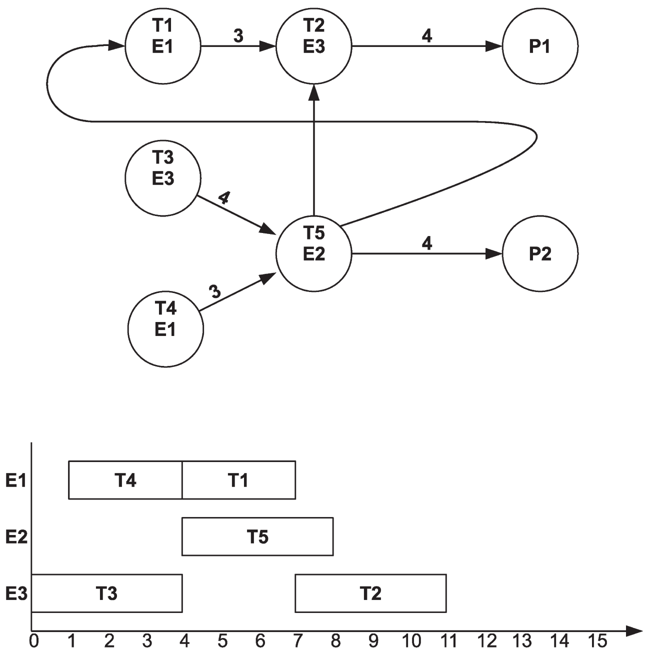

Figure 3. Schedule represented by an S-graph and the corresponding Gantt-chart.

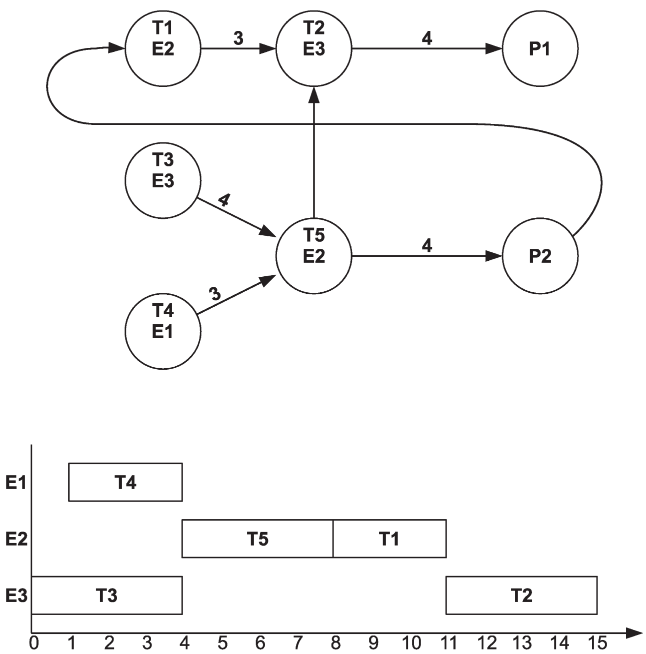

Figure 4. An other schedule represented by the corresponding S-graph and Gantt-chart.

Since both the schedule and the recipe-arcs express timing precedence between the execution of the tasks, the existence of a cycle would mean an infeasible schedule. Hence, after each decision, the resultant S-graph is tested for containing a cycle with the colored depth-first-search algorithm. If the graph is not acyclic, the decision is disapproved.

If all the necessary decisions have been made, the resultant S-graph uniquely defines a schedule and the corresponding Gantt-chart. In Figure 3, the remaining decisions have been made in such a way that both tasks T1 and T4 will be performed

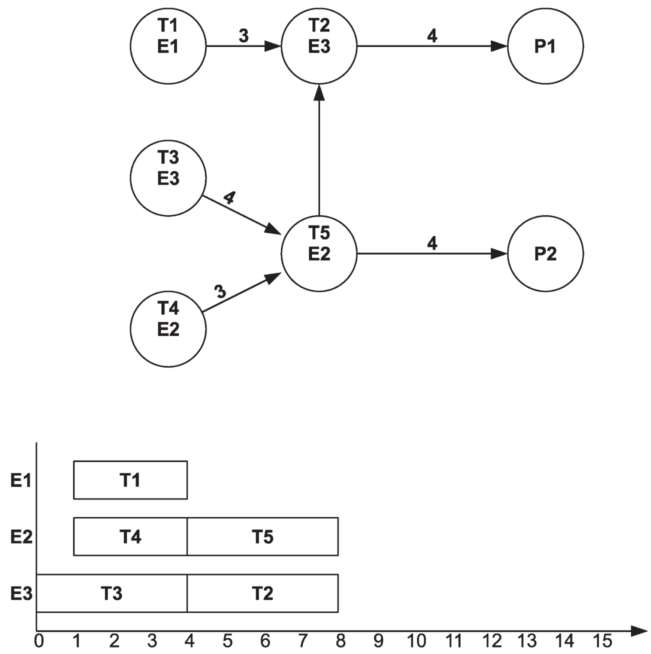

Figure 5. S-graph for the optimal schedule and the corresponding Gantt-chart.

in unit E1, starting with T4. For a scheduled S-graph, the longest path represents the critical path in the schedule, thus its value is the makespan itself. In this schedule the longest path is T3 → T5 → T1 → T2 → P1 with a length of 11 h.

An other possible schedule including the decisions of the partial schedule in Figure 2 is given with its Gantt-chart in Figure 4 that has the makespan of 15 h. The optimal schedule with the makespan of 8 h is given in Figure 5.

With minimization of the makespan, the optimization algorithm explores a branch-and-bound search tree, in which one S-graph corresponds to each node. The root of the tree is the S-graph that corresponds to the recipe, while the S-graphs at the leafs represent the possible schedules of the problem. The intermediate nodes correspond to partial schedules that are an extension of the S-graph corresponding to their parent node.

As it was mentioned, the efficiency of a branch-and-bound algorithm strongly depends on the selection of the branching and the bounding function and the node selection strategy. Two different types of branching functions have been published in the literature: the equipment-based method,[10] in which the decision is made to select the next task to be performed by a certain unit; and the task-based method,[28] in which an equipment unit is to be selected for a certain task. Both methods have their advantages, and the properties of scheduling problems determining which branching is more efficient to apply have been identified. The most simple bound that can be considered is the longest path of a partial schedule. Although it is easy to compute, it can provide considerably sharp bounds in many cases.

Additional Applications of S-Graphs. The previously described algorithm minimizes the makespan for a certain demand. Majozi Friedler[29] has introduced a new concept to apply S-graphs for throughput maximization. Laínez et al.[30] has adapted this concept to develop a new algorithm for maximizing the expected profit for stochastic scheduling problems with uncertain demand and price.

Romero et al.[31] have extended the original method to address batch scheduling problems with FIS policy that often appears in

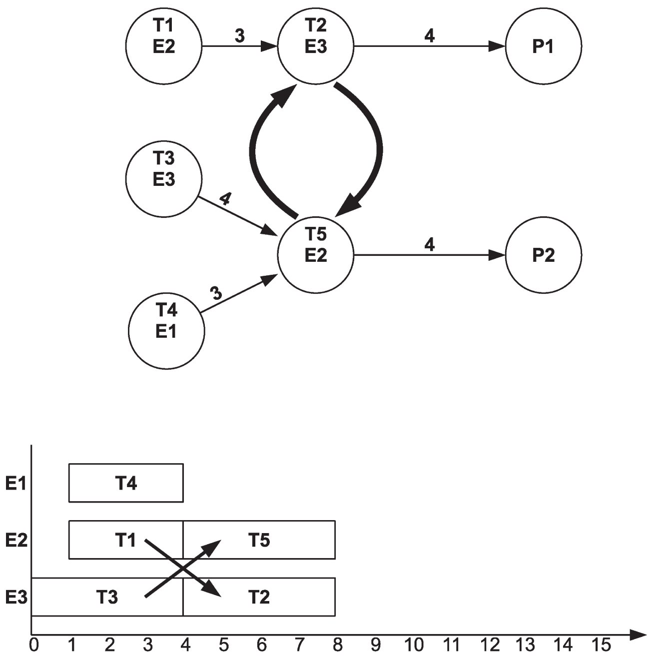

Figure 6. Infeasible schedule represented by an S-graph and its Gantt-chart.

the industry. Adonyi et al.[32] have developed an S-graph based algorithm addressing scheduling problems with cleaning costs and cleaning times that is common in paint production facilities. Adonyi et al.[33] have extended the original approach to take into account streams with various temperatures. In their approach, utility cost can be minimized for a given time horizon or the makespan can be minimized with an upper bound on the utility cost. Romero et al.[34] introduced a methodology based on the S-graph framework to provide optimal schedule by exploiting the inner flexibility of the batch process at the plant level.

A critical modeling issue has been recognized by Ferrer-Nadal et al.[35] and Hegyháti et al.[36] in the literature, which made MILP models result in infeasible solutions. The infeasibility arised as a situation, where a set of units must exchange their materials without available storage. It has been shown[36] that the combinatorial meaning of this infeasibility is a zero-weighted cycle in the ordering of tasks, which can be detected by a cycle search as any other infeasibility when applying the S-graph framework. In Figure 6, an infeasible schedule of the former example is given with its Gantt-chart and S-graph, where the zero-weighted cycle is highlighted. At 4 h of the time horizon, units E2 and E3 try to exchange their materials, which is inapplicable in practice. This issue highlights the advantage and importance of using straightforward combinatorial tools for the modeling and optimization of the scheduling of batch processes.

■ AUTOMATA AND PETRI NETS FOR SCHEDULING

Finite automata and Petri nets are widely applied in the analysis of discrete event systems.[37] These mathematical models can be especially useful to explore the reachable set of states of the system to find deadlocks and other potential undesired behaviors of a system. Several extensions exist to address timing in these models.

Timed automaton were introduced by Alur and Dill[38] extended to Priced Timed Automaton (PTA) by Behrmann et al.[39]

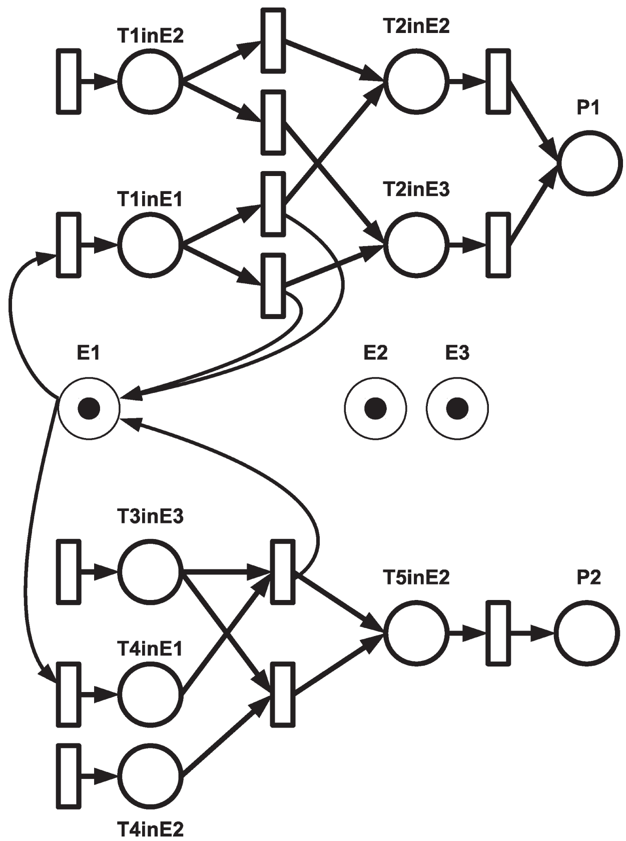

Figure 7. Petri net representation of the problem given in Figure 1. (Note that the arcs connected to places E2 and E3 are not shown for the sake of visibility.)

On the basis of PTA and the zone abstraction technique, the formulation of multiproduct batch scheduling problems was given by Panek et al.[12] Ghaeli et al.[11] proposed a modeling of batch scheduling problems based on Timed Place Petri nets. Both approaches apply branch-and-bound algorithms to explore the set of reachable states and find the goal state with the smallest value.

The Petri net representation of the problem given in Figure 1 is shown in Figure 7. In this Petri net, a place corresponds to the execution of a task in a unit. Moreover, a place is defined for each product and equipment unit. Transitions refer to the starting or ending of a task in a particular unit. For the sake of visibility, the arcs connected to the places representing units E2 and E3 are not shown in the figure and only the arcs related to equipment unit E1 are given. Note, that automata and Petri net based approaches also avoid cross transfer, which appears in these models as a deadlock.

■ COMPUTATIONAL COMPARISON

To illustrate the expediency of combinatorial algorithms in batch process scheduling, an exhaustive comparison has been made for a multiproduct scheduling problem,[40] where 4 products, A, B, C, and D are to be produced in three consecutive steps. The recipe with the five available equipment units (U1−U5) and processing times is shown in Figure 8.

The objective is to minimize the makespan for the given number of batches. In this study, 28 different cases were examined: starting from 1 batch up to 7 batches from each product. Each case was solved by four different approaches: a global time point based formulation by Maravelias and Grossmann;[41] a unit specific time point based formulation by Shaik et al;[42] a precedence based formulation by Mendez and Cerda;[43] and the S-graph based algorithm by Sanmarti et al.[10]

Each test was performed on the same computer with an AMD ATHLON 1.8 GHz processor and 1.5 GB of memory. For the S-graph based approach, a recent implementation by Smidla and Heckl[44] was applied. The MILP models were solved by CPLEX version 11.1.1. The time limit for the optimization was 1000 CPU seconds if an approach failed to finish in this time limit and the current best solution was reported. In the case of time point based formulations, the iteration on time points was performed until the second occurrence of the same objective value. For the three-index formulation, the Δn parameter was considered to be 0.

The results of the comparison are given in Table 1. The S-graph based solver and the precedence based formulation could provide the best results. The more general time point based formulations reached the 1000 s limit at much smaller batch numbers.

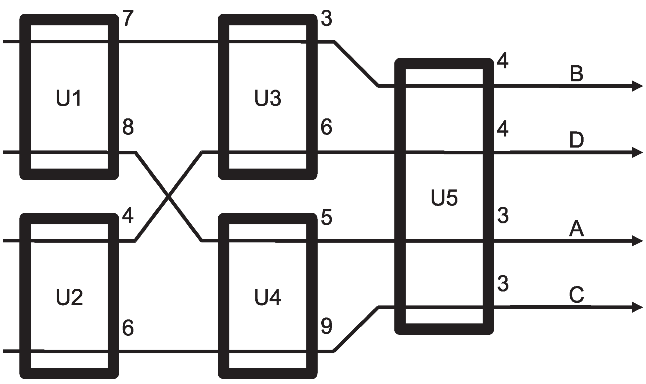

Figure 8. Recipe of the example for the comparison study.

■ CONCLUDING REMARKS

The combinatorial nature of batch process scheduling has been investigated. Problem representation and modeling that fits the scheduling is crucial both in guaranteeing the optimal solution and reducing the computational effort. S-graph framework initiated by Puigjaner and Friedler proved to be appropriate for problem representation and for involving combinatorial algorithms in the solution procedure. Several additional combinatorial algorithms have been presented here as an illustration.

Because of the industrial requirements, scheduling problems are getting more and more complicated; therefore, the efficacy of the solution procedures is crucial. The present work illustrated that combinatorial algorithms and tools play important role in adequate modeling and in the effective solution of batch scheduling problems. Combinatorial algorithms can further be exploited to solve industrial problems with additional requirements, e.g., simultaneous heat integration and scheduling. It is assumed that S-graph framework provides an appropriate base for that.

Table 1. Comparison Results for the Example

no. of batches				S-graph (2002)[10]		M&G (2003)[41]		M&C (2003)[43]		S&F (2009)[42]	
A	B	C	D	Obj.	CPU time	Obj.	CPU time	Obj.	CPU time	Obj.	CPU time
1	1	1	1	25 h	0 s	25 h	26 s	25 h	0 s	25 h	1 s
2	1	1	1	31 h	0 s	31 h	73 s	31 h	0 s	31 h	2 s
2	2	1	1	37 h	0 s	37 h	318 s	37 h	0 s	37 h	2 s
2	2	2	1	39 h	0 s	≤39 h	1000 s	39 h	0 s	39 h	11 s
2	2	2	2	41 h	0 s	≤41 h	1000 s	41 h	0 s	41 h	29 s
3	2	2	2	46 h	0 s	≤54 h	1000 s	46 h	0 s	47 h	99 s
3	3	2	2	52 h	0 s	≤54 h	1000 s	52 h	0 s	52 h	88 s
3	3	3	2	55 h	0 s	≤57 h	1000 s	55 h	0 s	55 h	788 s
3	3	3	3	57 h	1 s	≤60 h	1000 s	57 h	0 s	≤57 h	1000 s
4	3	3	3	62 h	1 s	≤69 h	1000 s	62 h	1 s	≤63 h	1000 s
4	4	3	3	67 h	1 s	≤76 h	1000 s	67 h	1 s	≤68 h	1000 s
4	4	4	3	71 h	2 s	≤77 h	1000 s	71 h	3 s	≤72 h	1000 s
4	4	4	4	73 h	8 s	≤81 h	1000 s	73 h	8 s	≤74 h	1000 s
5	4	4	4	78 h	6 s	≤88 h	1000 s	78 h	8 s	≤79 h	1000 s
5	5	4	4	82 h	4 s	≤93 h	1000 s	82 h	6 s	≤85 h	1000 s
5	5	5	4	87 h	35 s	≤103 h	1000 s	87 h	13 s	≤89 h	1000 s
5	5	5	5	89 h	138 s	≤102 h	1000 s	89 h	96 s	≤92 h	1000 s
6	5	5	5	94 h	96 s	≤102 h	1000 s	94 h	25 s	≤98 h	1000 s
6	6	5	5	98 h	63 s	≤121 h	1000 s	98 h	30 s	≤02 h	1000 s
6	6	6	5	103 h	580 s	≤126 h	1000 s	103 h	336 s	≤11 h	1000 s
6	6	6	6	≤105 h	1000 s	≤125 h	1000 s	105 h	921 s	≤15 h	1000 s
7	6	6	6	≤110 h	1000 s	≤137 h	1000 s	110 h	306 s	≤12 h	1000 s
7	7	6	6	≤114 h	1000 s	≤140 h	1000 s	113 h	600 s	≤21 h	1000 s
7	7	7	6	≤119 h	1000 s	≤146 h	1000 s	≤119 h	1000 s	≤24 h	1000 s
7	7	7	7	≤123 h	1000 s	≤146 h	1000 s	≤121 h	1000 s	≤28 h	1000 s

dx.doi.org/10.1021/ie101800x |Ind. Eng. Chem. Res. 2011, 50, 5169–5174

AUTHOR INFORMATION

Corresponding Author

*E-mail: friedler@dcs.uni-pannon.hu. Phone: +3688424483. Fax: +3688428275.

REFERENCES

(1) Mendez, C. A.; Cerda, J.; Grossmann, I. E.; Harjunkoski, I.; Fahl, M. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Comput. Chem. Eng.* **2006**, *30*, 913–946.

(2) Hegyhati, M.; Friedler, F. Overview of Industrial Batch Process Scheduling. *Chem. Eng. Trans.* **2010**, *21*, 895–900.

(3) Shaik, M. A.; Floudas, C. A. Novel unified modeling approach for short-term scheduling. *Ind. Eng. Chem. Res.* **2009**, *48* (6), 2947–2964.

(4) Kilic, O. A.; van Donk, D. P.; Wijngaard, J. A discrete time formulation for batch processes with storage capacity and storage time limitations. *Comput. Chem. Eng.* **2010**in press.

(5) Kopanos, G. M.; Puigjaner, L. Simultaneous Batching and Scheduling in Multi-product Multistage Batch Plants through Mixed-Integer Linear Programming. *Chem. Eng. Trans.* **2010**, *21*, 505–510.

(6) Friedler, F.; Tarjan, K.; Huang, Y. W.; Fan, L. T. Graph-Theoretic Approach to Process Synthesis: Axioms and Theorems. *Chem. Eng. Sci.* **1992**, *47*, 1973–1988.

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

(8) Pantelides, C. C. Unified frameworks for optimal process planning and scheduling. In *Foundations of Computer Aided Process Operations; Rippin*, D. W. T., Hale, J. C., Davis, J., Eds.; CACHE: Austin, TX, 1994; pp 253—274.

(9) 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* (25), 5935–5949.

(10) Sanmarti, E.; Holczinger, T.; Puigjaner, L.; Friedler, F. Combinatorial framework for effective scheduling of multipurpose batch plants. *AIChE J.* **2002**, *48* (11), 2557–2570.

(11) Ghaeli, M.; Bahri, P. A.; Lee, P.; Gu, T. Petri-net based formulation and algorithm for short-term scheduling of batch plants. *Comput. Chem. Eng.* **2005**, *29*, 249–259.

(12) Panek, S.; Engell, S.; Subbiah, S.; Stursberg, O. Scheduling of multi-product batch plants based upon timed automata models. *Comput. Chem. Eng.* **2008**, *32*, 275–291.

(13) Korte, B. H.; Vygen, J. *Combinatorial Optimization: Theory and Algorithms*, 4th Edition; Springer Publishing Company: Berlin, Germany, 2008.

(14) Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Stein, C. *Introduction to Algorithms*, 3rd Edition; MIT Press: Cambridge, MA, 2009.

(15) Tarjan, R. E. Depth-First Search and Linear Graph Algorithms. *SIAM J. Comput.* **1972**, *1*, 146–160.

(16) Land, A. H.; Doig, A. G. An automatic method of solving discrete programming problems. *Econometrica* **1960**, *28* (3), 497–520.

(17) Friedler, F.; Varga, J. B.; Feher, E.; Fan, L. T. Nonconvex Optimization and Its Applications. In *State of the Art in Global Optimization, Computational Methods and Applications*; Floudas, C. A.; Pardalos, P. M., Eds.; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1996; Chapter Combinatorially Accelerated Branch-and-Bound Method for Solving the MIP Model of Process Network Synthesis, pp 609—626.

(18) Imreh, C. Scheduling problems on two sets of identical machines. *Computing* **2003**, *70*, 277–294.

(19) Nagy-György, J.; Imreh, C. On-line scheduling with machine cost and rejection. *Discrete Appl. Math.* **2007**, *155*, 2546–2554.

(20) Kuhn, H. W. The Hungarian Method for the assignment problem. *Naval Res. Logistics Q.* **1955**, *2*, 83–97.

(21) Burkard, R.; Dell'Amico, M.; Martello, S. *Assignment Problems*; Society for Industrial and Applied Mathematics: Philadelphia, PA, 2009.

(22) Johnson, S. Optimal Two and Three-Stage Production Schedules with Setup Times Included. *Naval Res. Logistics Q.* **1954**, *1*, 61–67.

(23) Jackson, J. An Extension of Johnson's Results on Job Lot Scheduling. *Naval Res. Logistics Q.* **1956**, *3*, 201–203.

(24) Romero, J. *Contribution to Flexible-Process-Systems development in the Batch Processing Industry*. Ph.D. Thesis, Escola Tècnica Superior d'Enginyeria Industrial de Barcelona - Universitat Politècnica de Catalunya, Barcelona, Spain, 2003.

(25) Nadal, S. F. *Contribution to the Optimization and Flexible Management of Chemical Processes*. Ph.D. Thesis, Escola Tèccnica Superior d'Enginyeria Industrial de Barcelona - Universitat Politècnica de Catalunya, Barcelona, Spain, 2008.

(26) Lainez, J. M. *Towards Integrated Management of the Supply Chain for Enterprise Sustainability*. Ph.D. Thesis, Escola Tècnica Superior d'Enginyeria Industrial de Barcelona - Universitat Politècnica de Catalunya, Barcelona, Spain, 2009.

(27) Holczinger, T. *Method for scheduling non-intermediate storage batch process systems*. Ph.D. Thesis, Doctoral School of Information Science and Technology - University of Veszprem, Veszprem, Hungary, 2004.

(28) Adonyi, R. *Batch process scheduling with the extensions of the S-graph framework*. Ph.D. Thesis, Doctoral School of Information Science and Technology - University of Veszprem, Veszprem, Hungary, 2008.

(29) Majozi, T.; Friedler, F. Maximization of throughput in a multipurpose batch plant under fixed time horizon: S-graph approach. *Ind. Eng. Chem. Res.* **2006**, *45*, 6713–6720.

(30) Laínez, J.; Hegyháti, M.; Friedler, F.; Puigjaner, L. Using S-graph to address uncertainty in batch plants. *Clean Technol. Environ. Policy* **2010**, *12*, 105–115.

(31) Romero, J.; Puigjaner, L.; Holczinger, T.; Friedler, F. Scheduling Intermediate Storage Multipurpose Batch Plants Using the S-Graph. *AIChE J.* **2004**, *50* (2), 403–417.

(32) Adonyi, R.; Biros, G.; Holczinger, T.; Friedler, F. Effective scheduling of a large-scale paint production system. *J. Cleaner Prod.* **2008**, *16* (2), 225–232.

(33) Adonyi, R.; Romero, J.; Puigjaner, L.; Friedler, F. Incorporating heat integration in batch process scheduling. *Appl. Therm. Eng.* **2003**, *23*, 1743–1762.

(34) Romero, J.; Espuna, A.; Friedler, F.; Puigjaner, L. A New Framework for Batch Process Optimization Using the Flexible Recipe. *Ind. Eng. Chem. Res.* **2003**, *42* (2), 370–379.

(35) Ferrer-Nadal, S.; Capón-García, E.; Méndez, C. A.; Puigjaner, L. Material transfer operations in batch scheduling. A critical modeling issue. *Ind. Eng. Chem. Res.* **2008**, *47*, 7721–7732.

(36) Hegyháti, M.; Majozi, T.; Holczinger, T.; Friedler, F. Practical infeasibility of cross-transfer in batch plants with complex recipes: S-graph vs MILP methods. *Chem. Eng. Sci.* **2009**, *64*, 605–610.

(37) Cassandras, C. G.; Lafortune, S. *Introduction to Discrete Event Systems*; Springer-Verlag New York, Inc.: Secaucus, NJ, 2006.

(38) Alur, R.; Dill, D. L. A theory of timed automata. *Theor. Comput. Sci.* **1994**, *126*, 183–235.

(39) Behrmann, G.; Fehnker, A.; Hune, T.; Larsen, K.; Pettersson, P.; Romijn, J. *Efficient Guiding Towards Cost-Optimality in UPPAAL*; BRICS, Department of Computer Science, University of Aarhus: Aarhus, Denmark, 2001.

(40) Voudouris, V. T.; Grossmann, I. E. MILP model for scheduling and design of a special class of multipurpose batch plants. *Comput. Chem. Eng.* **1996**, *20*, 1335–1360.

(41) Maravelias, C. T.; Grossmann, I. E. A New Continous-Time State Task Network Formulation for Short Term Scheduling of Multipurpose Batch Plants. *Comput.-Aided Chem. Eng.* **2003**, *14*, 215–220.

(42) Shaik, M. A.; Floudas, C. A.; Kallrath, J.; Pitz, H.-J. Production scheduling of a large-scale industrial continuous plant: Short-term and medium-term scheduling. *Comput. Chem. Eng.* **2009**, *33*, 670–686.

(43) Mendez, C. A.; Cerda, J. An MILP Continuous-Time Framework for Short-Term Scheduling of Multipurpose Batch Processes Under Different Operation Strategies. *Optim. Eng.* **2003**, *4*, 7–22.

(44) Smidla, J.; Heckl, I. S-graph based parallel algorithm to the scheduling of multipurpose batch plants. *Chem. Eng. Trans.* **2010**, *21*, 937–942.