CHEMICAL ENGINEERING TRANSACTIONS

VOL. 70, 2018

A publication of AIDIC

The Italian Association
of Chemical Engineering
Online at www.aidic.it/cet

Guest Editors: Timothy G. Walmsley, Petar S. Varbanov, Rongxin Su, Jiří J. Klemeš
Copyright © 2018, AIDIC Servizi S.r.l.
ISBN 978-88-95608-67-9; **ISSN** 2283-9216

DOI: 10.3303/CET1870020

Batch Process Scheduling with *e*S-graph: A Case Study

Máté Hegyháti

Department of Information Technology, Széchenyi István University, Győr, Hungary
hegyhati@sze.hu

The scheduling of batch processes is a widely researched field of chemical engineering. Over the last few decades, a great number of tools have been developed for industrial examples and literature problems. These methods vary not only in their applied model, but in their representation of the problem inputs as well. The most well-known representations are the State-Task Network, the Resource-Task Network, the State Sequence Network, and the S-graph. While the latter also serves as the mathematical model for the related optimization approaches, the others only act as an intermediate model between the raw problem data and the model used for optimization, e.g., a mixed-integer linear programming model. The *e*S-graph model is a generalization of the S-graph, where the one-to-one relation between nodes and tasks has been relaxed, allowing a much wider range of scheduling problems to be tackled. Processes may simultaneously occupy several units, and release them at different stages of execution, and certain stages of separate processes can be forced to overlap in time. As in the case of the S-graph, the *e*S-graph models can be solved to optimality by specially designed combinatorial algorithms or serve as a basis for precedence based linear programming formulations. In this work, the modelling capacities of the *e*S-graph framework are illustrated via a Polymer production case study, where complex timing constraints are present.

1. Introduction

Batch process are advantageous for high value products with changing demands. Their operation, however, requires additional attention compared to their continuous counterparts: the scheduling of tasks to units is often a complex, and mathematically challenging problem. Depending on the processes and the facility, wide range of different scheduling problems can be identified (Mendez et al., 2006) which have been addressed by diverse techniques in the literature over the years (Hegyhati and Friedler, 2010). The most widely applied approaches are the various MILP (Mixed-Integer Linear Programming) formulations (Floudas and Lin, 2004) and the S-graph framework (Sanmarti et al., 2002), however, other techniques such as the Linearly Priced Timed Automaton (Panek et al., 2008) or Timed Placed Petri Nets (Ghaeli et. al., 2005) were also investigated.

For MILP approaches, the raw problem data is usually represented in an intermediate model also, such as an STN (State-Task-Network; Kondili, 1993), RTN (Resource-Task-Network; Pantelides, 1993), or SSN (State-Sequence-Network; Majozi and Zhu, 2001). The other techniques use their own mathematical model as a graphical representation, i.e., the S-graph, timed automaton or Petri nets. While all of these are excellent tools for building a mathematical model, or visualizing the problem and/or its solution, they are often restricted, and cannot directly express common practical constraints, such as multiple or overlapping usage of resources. To overcome this shortcoming, the models are often extended, or the problem itself is converted to an already addressed class of scheduling problems.

The *e*S-graph (event scheduling graph) model was introduced (Hegyhati, 2015) to provide a more flexible frame, that allows to solve wider range of scheduling problems, if an optimization approach is provided for this general model. Thanks to its roots in the formerly mentioned S-graph model, an *e*S-graph model can be solved to optimality by a similar branch-and-bound approach, as the one developed for the S-graphs. Precedence based MILP models, however, also show a lot of similarities. Thus, in this paper, the illustrative case-study has been solved by an *e*S-graph based MILP model.

Please cite this article as: Hegyhati M., 2018, Batch process scheduling with es-graph: a case study , Chemical Engineering Transactions, 70, 115-120 DOI:10.3303/CET1870020

2. The *e*S-graph model

To facilitate understanding, a brief introduction is given here to the eS-graph model. The basic building blocks are the events and the subprocesses. Event, as their name suggest are instantaneous changes in the state of the system, such as the start of a transfer operation, the ending of a production step, a material reaching a certain temperature, etc. Subprocesses are parts of a process, that have well defined resource needs. Subprocesses are defined by the events they must span over with. This way, subprocesses may also overlap. As an example, a reaction subprocess requiring the reactor overlaps with the transfer subprocess requiring the pipes between the events of the reaction ending, the transfer starting, and the transfer finishing. Subprocesses can have several modes, each of them requiring different resources, and the decision of mode selection may also alter the timing of some events.

The *e*S-graph is a directed graph, whose vertices are the events, and each subprocess is a subset of those events. The arcs between the vertices are defined by required timing differences of events, similar to the S-graph framework, however, each arc has a lower and upper bound on it. This way, Limited- and Zero-Wait storage policies are straight-forward to express.

If the model is solved by an S-graph-like branch-and-bound algorithm, the weight of the arcs may change during mode allocation decisions, and each scheduling decision will enforce the addition of new arcs. Two subprocesses have to be ordered, if such modes are chosen for them, that have overlapping resource needs.

3. The proposed precedence based MILP model

The case study presented in Section 4 was solved by a precedence based MILP model, which will be briefly introduced here.

Variables

The model has a non-negative continuous variable assigned to all events, that represents its timing: t_e, where *e* is an event from event set *E*. Similar to the allocation variables commonly used in precedence-based models, the mode selection variables are binary, and defined for all of the modes of all of the subprocesses: y_{sm}. The variable takes the value of 1, if mode *m* is selected for subprocess *s*. The model applies general precedence variables, $X_{sms'm'}$ is defined if mode *m* for subprocess *s* requires at least one resource that is also required for *s'* in mode *m'.* The variable takes the value of 1 if and only if modes *m* and *m'* are selected for subprocesses *s* and *s'*, respectively, and *s* is scheduled to be earlier than *s'.* Lastly, *MS* is a non-negative continuous variable representing the makespan of the whole process

Constraints

The model is composed of the following nine constraints. Equations (1) and (2) express the timing bounds defined by the arcs in *A.*

$$t_{e'} \geq t_e + bound^L_{(e,e')} \qquad \forall (e,e') \in A \tag{1}$$

$$t_{e'} \leq t_e + bound^U_{(e,e')} \qquad \forall (e,e') \in A \tag{2}$$

Equation (3) ensures that exactly one mode is selected for all of the subprocesses.

$$\sum_{m \in M_s} y_{sm} = 1 \qquad \forall s \in S \tag{3}$$

Equations (4) and (5) express the timing bounds between events that are the result of a mode selection for a subprocess.

$$t_{e'} \geq t_e + \sum_{m \in M_s} bound^L_{s,(e,e')} \qquad \forall (e,e') \in A_s \tag{4}$$

$$t_{e'} \leq t_e + \sum_{m \in M_s} bound^U_{s,(e,e')} \qquad \forall (e,e') \in A_s \tag{5}$$

Equation (6) ensures, that a sequencing variable is set to 0, if at least one of its mode selections are not selected. Also if both *m* is selected for *s,* and *m'* for *s'*, then they can be sequenced in only one order.

$$x_{sms'm'} + x_{s'm'sm} \leq \frac{y_{sm} + y_{s'm'}}{2} \tag{6}$$

$$\forall s, s' \in S,\, m \in M_s,\, m' \in M_{s'},\, s \neq s',\, R_{sm} \cap R_{s'm'} \neq \emptyset$$

Equation (7) ensures, that if both of the aforementioned assignments are made, they are ordered one way or another.

$$x_{sms'm'} + x_{s'm'sm} \geq y_{sm} + y_{s'm'} - 1 \tag{7}$$

$$\forall s, s' \in S,\, m \in M_s,\, m' \in M_{s'},\, s \neq s',\, R_{sm} \cap R_{s'm'} \neq \emptyset$$

Equation (8) enforces the events of *s'* should happen later than that of *s*, if it is sequenced later.

$$t_e \geq t_{e'} - M \cdot (1 - x_{sms'm'}) \tag{8}$$

$$\forall s, s' \in S,\, e \in E_s,\, e' \in E_{s'},\, m \in M_s,\, m' \in M_{s'},\, s \neq s',\, R_{sm} \cap R_{s'm'} \neq \emptyset$$

Finally, Equation (9) sets the value of the *MS* variable, that is to be minimized.

$$MS \geq t_e \qquad \forall e \in E \tag{9}$$

4. PHBA production case study

To illustrate the modelling power of the *e*S-graph, the first part of a case study is taken, where PHBA is produced. First, high purity (99.85 %) phenol (PhOH) is heated up to 50 ºC and then pumped to the phenolate reactor, that takes 2 h. Here the PhOH is heated further up to 60 degrees, before the 60 % KOH solution is introduced (2 h). When KOH is added, the reaction starts, and within 30 min, phenolate (PhOK) and water is produced. The intermediate product must immediately be transferred to the carboxylation reactor, which transfer takes 1 hour. After the transfer, the reactors must be cleaned (1 hour). At the arrival of the intermediate, heated (230 degree) marlotherm (MN) should be already present in the carboxilation reactor. MN is pumped to the reactor a-priori (2 h) and heated up there using an external circulation oil heater (1 hour). When the intermediate enters into the carboxylation reactor, nitrogen is heated up, and passed through the marlotherm to help with the drying process, that takes around 30 min above the transfer time. After this, hot CO_2 and nitrogen are introduced, and PHBAK2 and PhOH are produced, while any excess MN, nitrogen or CO_2 is removed together with PhOH. The reaction, cooling, and emptying the reactor takes 3.5 h.

The *e*S-graph model of the process

The *e*S-graph model of the process is shown in Figure 1. The model has 21 events and 3 subprocesses. The events are:

e1,e2 - Starting and finishing of the PhOH fill into the phenolation reactor
e3,e4 - Starting and finishing of the PhOH heating in the phenolation reactor
e5,e6 - Starting and finishing of the phenolation reaction
e7,e8 - Starting and finishing of the transfer between the two reactors
e9,e10 - Starting and finishing of the cleaning of the phenolation reactor
e11,e12 - Starting and finishing of the MN fill into the carboxylation reactor
e13,e14 - Starting and finishing of the MN heating in the carboxylation reactor
e15 - Finishing of the drying process
e16,e17 - Starting and finishing of the carboxylation reaction
e18,e19 - Starting and finishing of the cooling in the carboxylation reactor
e20,e21 - Starting and finishing of the PhBAK2 removal from the carboxylation reactor

The three subprocesses overlap for some events. The phenolation and carboxylation reactors are both in use during the transfer of PhOK. Moreover, both the external heater, and the carboxylation reactor are in use when MN is heated up.

The arcs in most of the cases have only lower bounds (having infinity as an upper bound), as most of the processing steps could be done slower. The *(e6,e7)* arc, however, has an upper bound of 0, as PhOK must immediately be transferred to the carboxylation reactor where MN is already heated up.

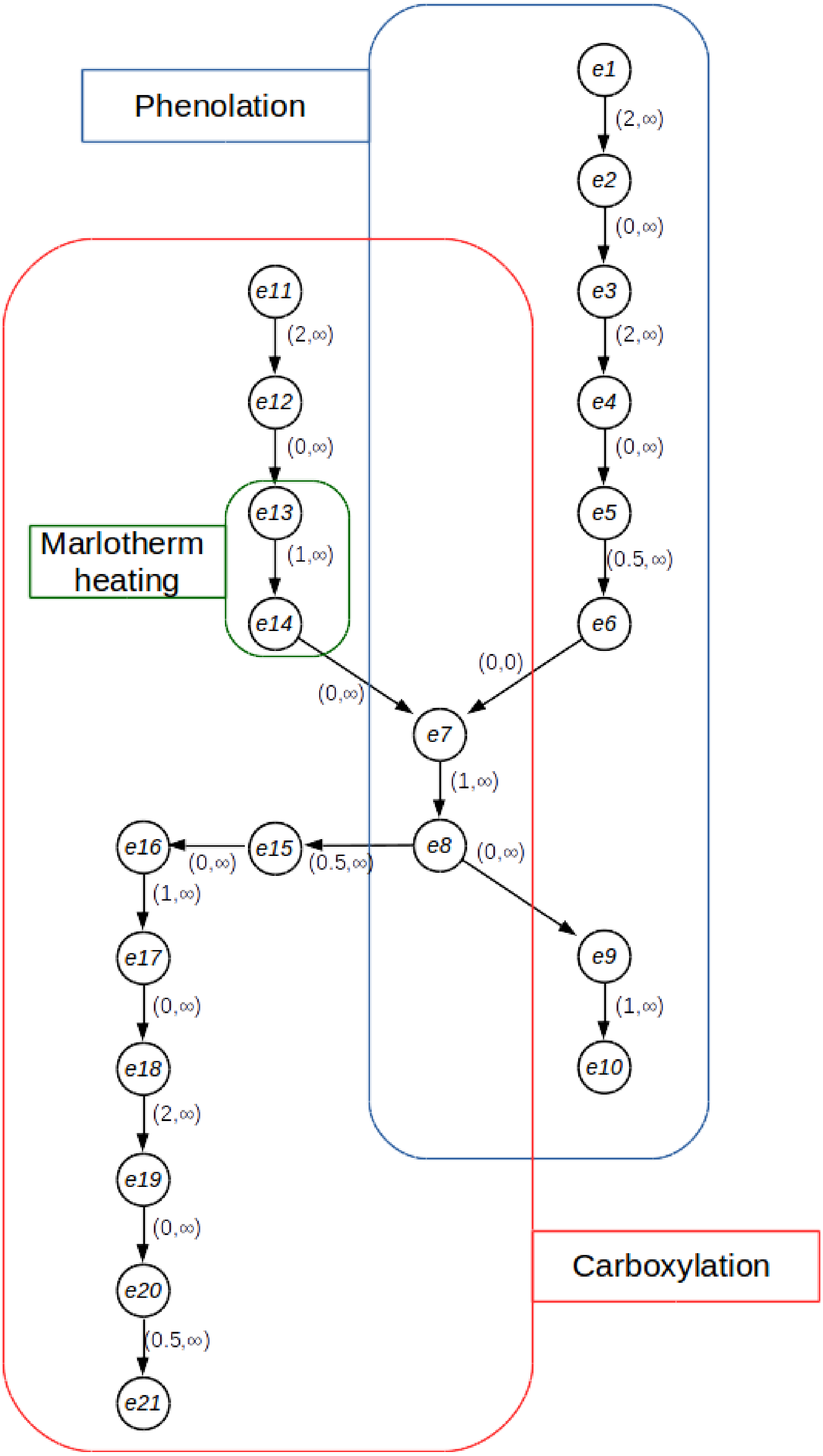

Figure 1: eS-graph model of the case study

This model has been implemented and solved to minimize the makespan with various configurations. Figure 2 shows the optimal schedule with 22.5 h for 3 batches with 1 phenolation reactor, 2 carboxylation reactors, and 1 external heater available. The colors represent the 3 batches, and the connected parts are the intervals *e7-e8* and *e13-e14,* that are shared by two subprocesses. In this scenario, the phenolation reactor is fully utilized, the carboxilation reactors on the other hand have some spare capacity. If *C1* or *C2* were to break down, the production could carry on with a somewhat increased makespan.

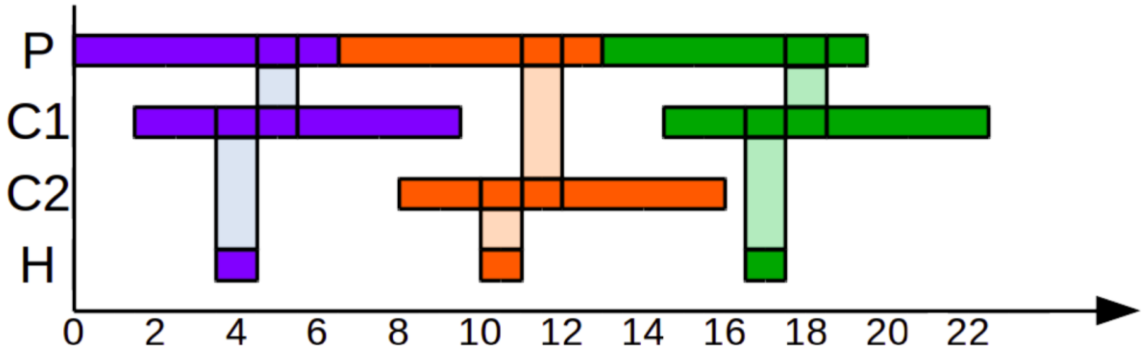

Figure 2: Gantt chart of the optimal solution for 3 batches

The minimal time required by a batch of product in the phenolation and carboxylation reactors are 6.5 and 8 hours, respectively, assuming that no delays are induced by the schedule. In the above example the normalized load for phenolation is 6.5 h per batch, and the same is just 4 hours for the carboxylation reactors, resulting the gaps in the schedule of *C1* and *C2.* Theoretically, the most ideal combination of reactors would be 13 for phenolation and 20 for carboxylation, so that the normalized loads match up exactly. An ideal schedule without gaps would still may be infeasible due to the schedule of the heater, not to mention the raison d'étre of investing into an additional 30 reactors for just that purpose.

A more balanced scenario, however, could be investigated, if only one additional reactor is added from both type, bringing the normalized loads to 3.25 and 2.67 hours respectively. The optimal solution for producing 6 batches of PhOK in this case is shown in Figure 3. The makespan is 24 h, however, in this schedule, the heated marlotherm has to wait for 1 h for the first (blue) and fifth (purple) batches, as indicated. This is due to the fact, that the heater becomes a bottleneck in this schedule. If the marlotherm cannot keep its temperature for 1 h, the upper limit on the arc between *e14* and *e7* must be set to a proper value. If that is less than one hour, the makespan of 24 h cannot be achieved, unless additional heaters are purchased.

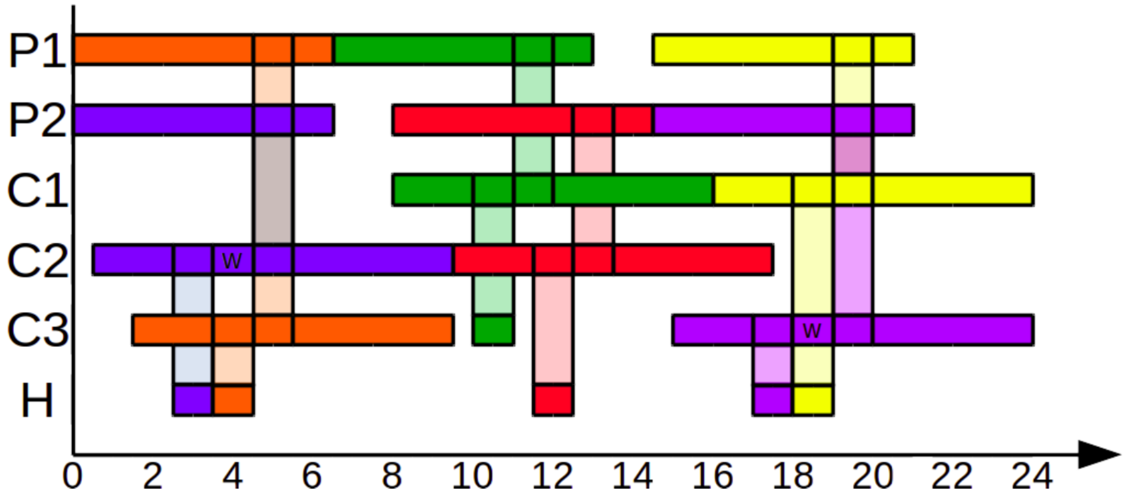

Figure 3: Gantt chart of the optimal solution for 6 batches with more reactors available

5. Conclusions

*e*S-graph is a powerful tool that can model complex recipes with various timing constraints. With the application of *e*S-graph there is no need for additional model transformations, or extensions to address recipes with overlapping resource demands, various timing constraints, etc. If a problem can be modeled as an *e*S-graph, it can be solved to optimality with either a combinatorial branch-and-bound algorithm, or a precedence based MILP model. To illustrate the capabilities of this new modelling tool, an industrial case study with a complex recipe has been solved for various configurations.

Acknowledgments

This research was supported by the EFOP-3.6.1-16-2016-00017; "Internationalization, initiatives to establish a new source of researchers and graduates, and development of knowledge and technological transfer as instruments of intelligent specializations at Szechenyi University" grant.

References

Floudas, C. A., Lin, X., 2004, Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. Computers & Chemical Engineering, 28(11), 2109–2129.

Ghaeli, M., Bahri, P. A., Lee, P., Gu, T., 2005, Petri-net based formulation and algorithm for short-term scheduling of batch plants. Computers & Chemical Engineering, 29(2), 249–259.

Hegyhati, M., Friedler, F., 2010, Overview of Industrial Batch Process Scheduling. Chemical Engineering Transactions, 21, 895–900.

Hegyhati M., 2015, Batch Process Scheduling: Extensions of the S-graph Framework, PhD Thesis, University of Pannonia, Doctoral School of Information Science and Technology, Veszprem, Hungary

Kondili, E., Pantelides, C. C., Sargent, R. W. H., 1993, A general algorithm for short-term scheduling of batch operations--I. MILP formulation. Computers & Chemical Engineering, 17(2), 211–227.

Majozi, T., Zhu, X. X., 2001, A novel continuous-time MILP formulation for multipurpose batch plants. 1. Short-term scheduling. Industrial & Engineering Chemistry Research, 40(25), 5935–5949.

Mendez, C. A., Cerda, J., Grossmann, I. E., Harjunkoski, I., Fahl, M., 2006, State-of-the-art review of optimization methods for short-term scheduling of batch processes. Computers & Chemical Engineering, 30(6–7), 913–946.

Panek, S., Engell, S., Subbiah, S., Stursberg, O., 2008, Scheduling of multi-product batch plants based upon timed automata models. Computers & Chemical Engineering, 32(1–2), 275–291.

Pantelides, C. C., 1993, Unified frameworks for optimal process planning and scheduling. In R. D. W. T., H. J. C., & D. J. (Eds.), Proceedings of the second international conference on foundations of computer-aided process operations, 253–274.

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