Computers and Chemical Engineering 106 (2017) 883–891

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Industrial scheduling solution based on flexible heuristics

Iiro Harjunkoski*, Reinhard Bauer

ABB Corporate Research, Wallstadter Str. 59, 68526 Ladenburg, Germany

ARTICLE INFO

Article history:
Received 30 September 2016
Received in revised form 27 January 2017
Accepted 9 February 2017
Available online 16 February 2017

Keywords:
Scheduling
Standards
Heuristics
Large-scale problems

ABSTRACT

This paper presents a generic heuristic-based scheduling solution. It highlights the flexibility that a simple heuristic method can offer and shows that using the ISA-95 standard it is possible to express the most relevant problem requirements. In order to illustrate the possible benefits, the paper also compares the solution quality of a smaller scale example scheduling problem to a rigorous mixed-integer linear programming (MILP) approach and shows how a heuristic approach scales towards large-size industrial problems. The paper concludes with a discussion of the advantages and disadvantages of both approaches, showing that for certain types of problems, the heuristic approach is fully sufficient, even if it cannot be expected to result in optimal solutions.

1. Introduction

Planning and scheduling of a production process and/or a production facility is often a complex task and requires high expertise to be successfully performed. First of all, the process understanding must be sufficient in order to know what exactly needs to be considered within the scheduling activity and what the true consequenses and impacts of the decisions are on the production level. The decisions that are taken at the scheduling level highly depend on the company, as well as, on the existing IT-infrastructure. Secondly, the technique to perform the scheduling task requires either a highly experienced operator trained on the job or a person with good mathematical and analytical skills – preferably both. Nevertheless, these skills do not always meet. Apart from the fact of qualification and personal skills, scheduling is often a full-day activity, especially in the modern networked world where the presence and above all the awareness of frequent disturbances or changes in the process and in the surrounding environment are more a rule than an exception. These can upset the schedule causing that it becomes practically invalid to be executed on the plant floor. In this situation correcting the schedule becomes the main effort making all optimization targets secondary. Therefore it is valuable to have access to supporting tools that ensure correct, agile and more efficient reactions to changes and that open the possibility for optimization also within a complex and frequently changing environment.

In general, scheduling optimization problems tend to be complex due to their highly combinatorial nature, caused by the large number of alternatives e.g. in selecting the best production sequence and equipment assignment options, but also because of the fact that a scheduling model – or for the most part almost any mathematical model – is always a simplification of reality. In the scope of scheduling optimization the major modeling challenges are to select the most relevant aspects that are needed for considering real-life production, the decisions that must be optimized and how to transfer the obtained results onto the plant floor to create the desired impact. From the scheduling point of view, events occuring in production are relevant only if they have an impact on the feasibility or profitability of the current schedule. The size or severity of the impact should then decide whether a new schedule must be triggered or if some local adjustments are sufficient. In other words, it would be important to close the "control loop" for scheduling. The overall challenge of seeing scheduling only as part of a larger entity is addressed by the concept Enterprise-Wide Optimization (EWO) (Grossmann, 2005) and related mathematical programming challenges are discussed in Grossmann (2014). A number of industrial case studies with some of these practical considerations have been reported. Pinto et al. (2000) focuses on planning and scheduling of refinery operations, and more recently Zhao et al. (2017) presents an integrated optimization approach coupling up-stream refinery and down-stream ethylene plant operations. Tang et al. (2001) provides a review for steel production and Janak et al. (2006) for the chemical industry, for which EWO aspects are discussed in Wassick (2009). Laínez et al. (2012) gives an overview of the EWO-opportunities in the pharmaceutical industry and O'Sullivan and Newman (2015) provides strategies to schedule

* Corresponding author.
E-mail addresses: iiro.harjunkoski@de.abb.com (I. Harjunkoski), reinhard.bauer@de.abb.com (R. Bauer).

http://dx.doi.org/10.1016/j.compchemeng.2017.02.018
0098-1354/

entire underground mine operations. A thorough discussion on the industrial aspects is given in Harjunkoski et al. (2014), where also various solution and modeling approaches are discussed, along the lessons learned from the industrial implemenations of scheduling. In order to be able to solve larger problem instances that are mostly triggered either by EWO-problem extensions or real case studies, a number of approaches can be found in the literature. Decomposition algorithms based on MILP (Kopanos et al., 2010), as well as in Zhao et al. (2017), hybrid approaches combining various techniques such as CP and MILP (Jain and Grossmann, 2001), MILP, heuristics and simulation (Basán and Méndez, 2016), timed-automata based approaches (Subbiah et al., 2011), sequential MILP-based algorithms building a schedule in a constructive manner (Roslöf et al., 2002), as well as greedy heuristics (Pranzo et al., 2003) are examples of these. Other drivers of related research come from the fact of linking production scheduling closer to the online processes. Good discussions and examples of these are shown in Gupta and Maravelias (2016), Gupta et al. (2016) and Kopanos and Pistikopoulos (2014), where reactive scheduling aspects enable closer analysis of the actual process states. More generic methodological steps are presented e.g. in Vegetti and Henning (2015). There exists also vast literature on algorithmics and optimization. A good overview can be found for example in Cormen et al. (2009), as well as in Manber (1989). A good discussion of algorithms and optimization in the context of manufacturing scheduling is given in Framinan et al. (2014). This work covers algorithmic complexity, exact, approximate and heuristic approach as well as the respective models. The academic literature is very broad across different research communities and a more thorough review is only possible with a more narrow and specialized scope such as about a specific industry.

In order to be able to utilize existing modeling and solution techniques efficiently, one of the best ways is to productize them as part of a larger scheduling and dispatching framework (Harjunkoski, 2016). The main challenges when productizing scheduling solutions are:

- Defining a software-landscape that can host the algorithmic environment providing both flexibility to alternate between solution approaches and sufficient computational capacity
- Finding a generic problem description that is able to express realistic problem instances and that can be configured to meet also more specialized needs
- Gathering the necessary data and communicate the results into production such that any deviations can be detected efficiently
- Providing algorithms that work efficiently for various cases and provide – if not optimal – solutions that can be used in practice
- Creating configuration environments that allow a non-expert to easily maintain and adapt the solution.

In this paper, we address the above challenges and present an approach that mainly fulfills them. Providing a relatively simple and clearly represented example allows us to highlight some of the main aspects of using heuristic approaches, as well as, to compare the resulting solution with a rigorous MILP approach. The scalability issue is shown by reporting the performance of the heuristics for much larger problem instances, showing that many intractable MILP problems can still be solved within seconds using a suitable heuristic algorithm, naturally compromising on the optimality. Thus, it will be shown that heuristic solutions are a good alternative to rigorous optimization in some cases and that an especially interesting challenge for the future is to design hybrid solution approaches for larger scheduling instances that combine robust mathematical programming methods with "non-optimal" heuristics.

This paper compares the practical performance of full-space MILP models and rather simple heuristic approaches. While those

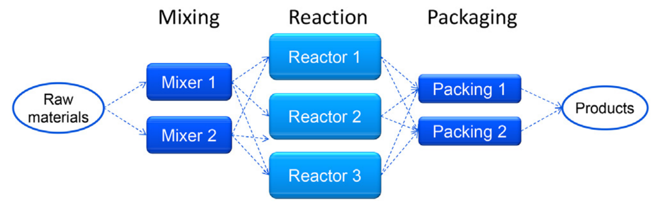

Fig. 1. Considered three-stage example production process.

two approaches follow completely different paradigms, both exist on their own right, being able to fulfill the requirements of different applications. A detailed discussion on the advantages and disadvantages of both paradigms and their corresponding scope is given later in the text. We would like to point out that math-heuristics stand in between both approaches, being the best choice for some applications that require a balance between algorithm runtime and solution quality.

2. Problem definition

For making it possible to treat the problem in a generic way and also putting the rest of this paper into context we define an example problem, which in this case is a simple chemical multi-stage batch process. The problem is an adaptation from Harjunkoski and Bauer (2014) and Harjunkoski and Bauer (2016), where we mainly have adjusted the processing times such that the bottleneck can shift depending on the solution candidate. The optimal solution is of course unique but having the variation can especially give a hint if non-optimal heuristic algorithm is driven towards a suboptimal solution and cannot thus fully use the desired flexibility. The example process comprises as before three stages: mixing, reaction and packaging. Here the reaction stage has three parallel machines and the other stages only have two alternatives to choose from. As earlier, all products cannot be processed on all machines, since the mixer 2 is incompatible for processing product B. There are also sequent-dependent changeover times, which must be considered. Fig. 1 shows a generic process overview.

Each of the stages consumes electricity and materials, however, when comparing heuristics with a mixed-integer programming formulation we exclude these in order to be able to use a standard precedence-based continuous-time approach with the main target of minimizing the make span. Nevertheless, it is straightforward to add the material constraints to the heuristic approach without large performance loss as will be shown later in this paper. Also, using a discrete-time approach (e.g. RTN in Castro et al., 2013) would be possible allowing full consideration of the material constraints but in this case, the problem size and resulting feasible grid density may not result in a global optimal solution from the makespan point-of-view, which is here of primary interest. The problem data is as follows: There are three different products that can be made (A, B, C) and we assume that a sequence-independent preparation or setup time is always needed, which is at minimum 15 min for mixers and reactors, depending on the sequence and always 60 min for the packaging machines, independent of the production sequence. This time is used e.g. for preparing and adjusting the equipment for the next task. The processing times are shown in Table 1, the field

Table 1
Processing times (min) of each product at each equipment.

Product	Mixer1	Mixer2	Reactor1	Reactor2	Reactor3	Packing1	Packing2
A	65	75	120	180	180	30	40
B	110	N/A	240	120	120	45	60
C	75	80	150	210	40	40	40