

A CUSTOMIZABLE PRODUCTION SCHEDULING FRAMEWORK FOR ERP SYSTEMS

János Beluzsár¹, Péter Martinek²

PhD student¹, associate professor²

Budapest University of Technology and Economics

ABSTRACT: Production scheduling is an essential process in manufacturing companies. ERP systems support any kind of business processes at any industrial and service area today but they usually don't offer full automation for solving the production scheduling task. In this paper we introduce a mathematical model for describing a production scheduling problem and design and implement a framework for solving it based on a standard constraint programming engine. Our solution able to handle complex multi-objective scheduling problems and uniquely capable of run time model parameter customization. Moreover the input of the model is originated from the manufacturing data of ERP systems via web service based interface. It makes our standard components based framework source system independent, but by the interoperability opportunities it is highly integrated to the enterprise business processes and the ERP processes as well.

INTRODUCTION

The steadily increasing amount of information generated in enterprise business processes requires fast and effective data management, which is impossible nowadays without IT support. In order to satisfy these requirements, Enterprise Resource Planning (ERP) systems are widely applied in today's companies. Market leader ERP vendors, like SAP, Oracle or Microsoft offer standard software suites with numerous cooperative modules supporting complex business processes. However some fields of the enterprise decision making processes are only partially covered by these software solutions, which often implies the use of additional dedicated programs. These areas are typically detailed production scheduling, product lifecycle management, transportation management or load- and truck-route optimization [1, 2]. However ERP systems offer standard interoperability possibilities, which makes possible to connect these use-case specific programs to them creating satisfying solutions [2].

As we mentioned earlier one of these supplementary ERP functions is detailed production scheduling at manufacturing companies. It determines the order of the manufactured goods on the machines for a time period, determining the short-term output of a plant. It has great influence on machine utilization or production lead times, which all play an important role of the overall manufacturing efficiency of a company, which defines significantly the customer satisfaction. However creating proper detailed production scheduling is a challenging task especially in mid-sized and large companies. Often experienced industry gurus create the daily production schedule on paper or with the help of huge and chaotic Excel sheets keeping all the important information in their head. This practice implies the uncontrollable goodness of the created schedules, and the day-by-day manual scheduling and

rescheduling processes. Unfortunately today's ERP systems generally don't offer efficient solution for the automation of this process. They often make possible to register and report the result of the detailed production scheduling, but they usually don't support the automation of the scheduling process. However ERP systems contains almost all of the necessary input data of the scheduling process, so they serve as a good fundament for automatic production scheduling solutions [3].

In this paper we introduce a production scheduling framework, which makes possible to create automatically production schedules based on the data of an ERP system. Our approach is using a standard solver software to generate run time the mathematical model of the problem based on the data collected from ERP system and find its optimal solution(if exists) even in case of multi-objective problems. In order to find the proper methodology to replicate the business objects of ERP systems into parameters of a scheduling problem we also examined the input data need and defined a standard interface for the framework, which makes possible to connect to any type of ERP systems. In this way the fast and smooth data exchange between the ERP and the framework is assured, making our solution ERP platform independent.

In the next chapter we introduce a brief literature review of the ERP's functionality and collaboration with third party software solutions. Then in the third chapter we discuss the model of our framework, and the theory of its working. We finally describe and summarize the experimental results in the fourth and fifth chapter.

LITERATURE REVIEW

Using ERP systems gives competitive advantages for companies by standardizing business processes, collecting and analyzing enterprise-wide information and supporting the decision making in different business levels [4, 5]. Therefore the popularity of ERP systems have increased over the past years and researches forecast continuous growing of the ERP market until the end of the decade [6].

ERP systems consist of several interconnected functional modules, each for supporting specific business needs. The most important and widely implemented modules are financials, controlling, materials management, production planning, sales and distribution, general logistics, project system, plant maintenance, quality management, human resources, supply chain management, customer relationship management, e-commerce and advance planner optimizer. To use any of these modules standardized, repeatable business processes and disciplined data recording are required in order to gain the desired operational improvement goals [7]. However industry or customer specific needs cannot always be realized by the standard ERP functionality Therefore ERP implementation projects mostly contain customer specific customizations as well. Because of that ERP vendors offer solutions for business logic, user interface and report output customization, data model extension and third party software communication [8]. This last opportunity makes possible to integrate data and/or business processes among ERPs and other dedicated programs mentioned in the introduction. These enterprise application

integration (EAI) scenarios nowadays usually implemented through widely-supported web-service technologies, driven by service oriented architecture (SOA) design principles.

THE INTEGRATION FRAMEWORK MODEL FOR SCHEDULING PROBLEMS

The main aim of production scheduling is to determine the production order on the manufacturing machines of a shop-floor. Therefore a huge amount of parameters, rules and objectives must be considered, making it a challenging task. In the literature this problem is referenced as job-shop scheduling problem. Jobs are manufacturing tasks of specified quantity of a product, which need to be completed before the deadline (e.g. producing 10 bicycles till Friday). Each job consists of a set of operations, which demand specific machine capacity for predetermined time interval in order to be completed. There are also precedence constraints among operations within a job which limits the possible sequences of operations in a schedule according to the technological regulations. Solving this problem means determining the order of the operations executed on the machines, regarding the defined constraints [9]. A simple scheduling problem with 3 jobs and 3 machines and a Gantt-chart represented solution of it is on Table 1 and Figure 1.

Table 1
The operations' manufacturing times on the particular machines

Job	Operation time/machine		
	1	2	3
J ₁	5/M ₁	4/M ₂	2/M ₃
J ₂	3/M ₁	4/M ₃	-
J ₃	2/M ₂	5/M ₁	3/M ₃

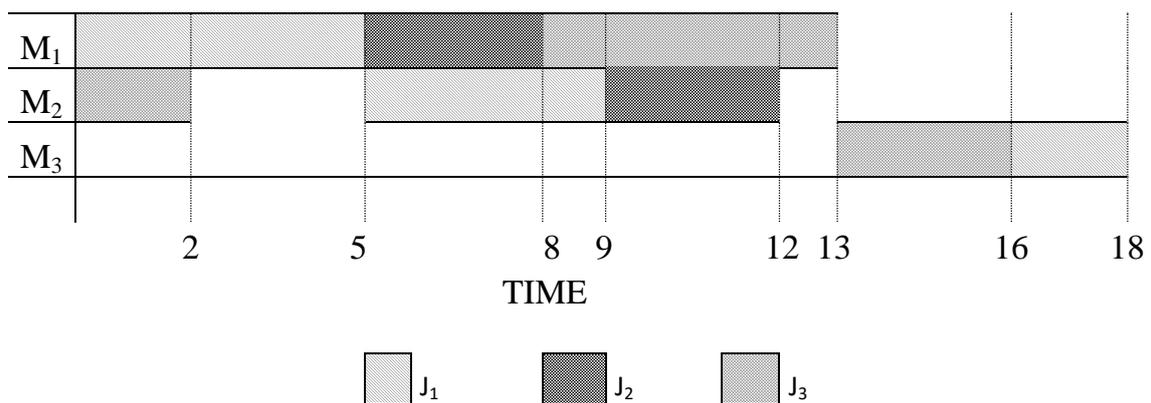


Figure 1
The Gantt-diagram representation of the solution of job-shop scheduling problem described in Table 1

As we can see in the example it is possible, that not every job has the same number of operations, therefore some jobs doesn't need to be manufactured on

every machine. For example job J_2 has only 2 operations and doesn't require work on machine M_3 . Moreover in some situations operations can be executed not only on one machine but also on one of a set of technologically compatible machines. In that case the goal of the scheduling is determining not only the sequence of operations on the machines, but also selecting an execution machine for every operation from the given set. This special case of job-shop scheduling called flexible job-shop scheduling [10]. There are also a huge variety of constraints to be considered in real-world scenarios. These are for example changeover times between operations (which represents sequence-dependent setup times between operations on certain machines), limited resource availability (e.g. molds, qualified operational staff for given machines or a specific raw material) or machine availability [11].

Besides the described context there is one more important parameter of job-shop scheduling problems. This is called the objective function. It represents the goal, needs to be reached with the scheduling method and algorithms. It is usually derived from business needs and its value makes possible to measure the goodness of a particular schedule. It can be calculated by the parameters of a concrete schedule and its typical types are minimizing makespan, tardiness or production costs. Therefore finding an optimal solution of a specific job-shop scheduling problem means finding a schedule where the value of the objective function is minimal. Making things more complex, there are usually more than one objective in production environments. For example in injection molding it is important to minimize the tardiness, because it has direct effect on customer satisfaction, but it is also essential to minimize changeover times, because it has negative influence on machine utilization. When there are more than one objective function, it is called multi-objective job-shop scheduling problem.

In order to integrate the capability of solving scheduling problems into ERP systems, we examined the characteristic of job-shop scheduling problem, and developed a mathematical model describing the input and output parameters of it. These are shown in Table 2.

Table 2
The input and output parameters of our model

<i>Notation</i>	<i>Description</i>	<i>Type</i>
$J = \{J_i\}_{i=1}^n$	set of jobs	input
$O = \{O_{ik}\}_{k=1}^{S_i}$	set of operations	input
$M = \{M_l\}_{l=1}^m$	set of machines	input
$\mu_{ik} \subseteq M$	set of machines compatible with O_{ik}	input
P_{ikl}	processing time of O_{ik} on machine M_l	input
SDS_{ikopl}	setup time between O_{ik} and O_{op} if O_{ik} directly before O_{op} on machine M_l	input
D_i	the due date of J_i	input
S_{ik}	the start time of O_{ik}	output
SM_{ikl}	1, if O_{ik} is executed on M_l 0, otherwise	output

We also studied the production planning modules of different ERP systems, and defined a common dataset, which is suitable for representing the input and output

parameters of the model. We examined three ERP systems, SAP R/3, QAD Enterprise Applications and Microsoft Dynamics NAV. All of these systems contains a business object representing a job. It is called work order or production order. It is an order for producing a specified quantity of a product for a given due date. It has a related dataset, called routing, which contains the operation sequence of a work order. Every operation refers to a work center in the ERP systems, on which they need to be executed, and they also contains the run time they load the work center. The operations of a work order routing represent the operations of a job in our model, and the work centers are identical with machines. Moreover work order independent routings are also available in the systems, which makes possible to register the technologically alternative work centers and operational times for every manufactured products. In this way the alternative machine information necessary for the flexible job-shop scheduling problem can correspond between ERP systems and our model.

Changeover times are also available in the SAP and QAD systems, but slightly different way, than it is defined in our model. In this two ERP systems, changeover times can be defined between manufactured parts, not the operations. However it is corresponding for our model, because work orders refer to manufactured parts, so the work order operations and the changeover times between manufactured parts can be linked indirectly. It means, that the changeover times between two operations of different work orders on a machine can be determined by searching the manufactured parts of the two work orders and selecting the changeover times defined between them on the appropriate machine.

Beside the input parameters, the output parameters can be linked to the model as well. The start time and the executing machine of an operation is also a parameter of the operation of the work order routing. In this way the result of the scheduling can be easily registered into the ERP systems.

The summary of the connections defined between the business objects of the ERP systems, and our model is described in Table 3.

Table 3
The connection between our model and ERP system data

<i>Notation</i>	<i>ERP business object</i>
$J = \{J_i\}_{i=1}^n$	work orders
$O = \{O_{ik}\}_{k=1}^{S_i}$	operations of the work order routing
$M = \{M_l\}_{l=1}^m$	work centers
$\mu_{ik} \subseteq M$	compatible work centers of an operation based on the standard routings assigned to the proper manufactured product
P_{ikl}	processing time of an operation on a work center based on the standard routings assigned to the proper manufactured product
SDS_{ikopl}	changeover times derived from manufactured product level to operational level
D_i	due date of the work order
S_{ik}	start time of the operation of the work order routing
SM_{ikl}	work center of the operation of the work order routing

As we mentioned earlier, when solving a job-shop scheduling problem there is always an objective function taking into consideration, which determines the goodness of a solution. In our approach, corresponding to the industrial needs, more than one objective functions can be taken into account in a hierarchical order. It means, that the objectives can be ordered by the user, and our framework tries to find the best solution in every level, but not generating worse schedules than the ones determined in prior levels. This approach is completely missed from ERP based production scheduling solutions. Most of them use priority based algorithms, some of them single-objective optimization, and even in multi-objective cases the objective function hierarchy is not taken into consideration. Therefore our framework represents a new direction among ERP based production scheduling solutions.

EXPERIMENTAL RESULTS

Based on the problem representation we described in the last chapter the scheduling framework has been implemented. For solving the job-shop scheduling problem we used the standard constraint programming(CP) engine of the IBM ILOG CPLEX Optimization Studio. It makes possible to build and solve constraint programming tasks, like job-shop scheduling. However we wanted to make our framework flexible, so we didn't limit the capabilities of the framework for only the model described earlier, but created a method for dynamic ERP driven problem instance generation. It means that first we created the basic elements (parameters, constraints, objective functions) like building bricks of the problem in the CP engine in Optimization Programming Language(OPL). It is the basic instruction set of the framework which can be combined into different concrete problems by the ERP systems using our framework. Moreover, we loosely-coupled this element definitions to the framework, which makes possible to easily extend them if other specific parameters, constraints or not yet defined objective functions need to be considered.

After defining and implementing the building elements of the problem, we created the interface, which connects our framework to ERP systems. It is a web-service based solution which defines standard functions for building up the problem representation from the predefined elements, adding the values of the input parameters, requesting the solver processing and returning the results. Because of the extensive computational need for solving a job-shop scheduling problem, we designed the interface for asynchronous work. This means that the ERP system needs to define a callback method with given input and output parameters, in order to return the result of the processing. In this way long-running calculations won't block the processes in the ERP system by unnecessary waiting for framework response.

According to the descriptions, the basic process flow of the framework is shown in Figure 2.

The described scheduling framework was implemented in C# and it was connected to the SAP R/3 ERP system. In order to test the configurability and goodness of our solution we have chosen benchmark problems of different problem categories from [12] and [13] to test it. The test cases are summarized in Table 4.

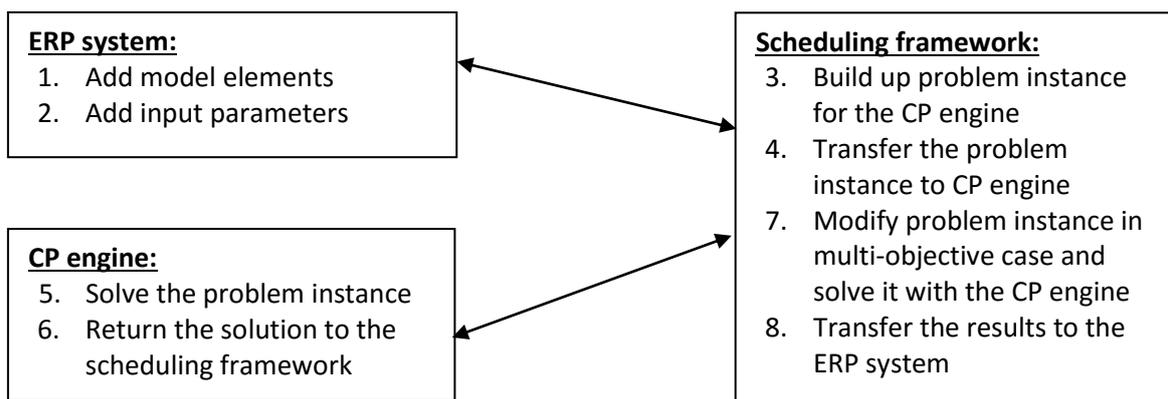


Figure 2

The basic components and process flow of the solution

Table 4

The tested benchmark problems, and the accuracy of our solutions

<i>Name</i>	<i>Objective function</i>	<i>Accuracy</i>
edata[13]	Minimize makespan	77%
rdata[13]	Minimize makespan	85%
vdata[13]	Minimize makespan	64%
table XV[12]	Minimize makespan and minimize machine workload	98%

The benchmark problems edata, rdata and vdata are similar type of scheduling problems with different sizes. We tested them in our framework and measured the average distance of our objective function value of the ones given in the literature. Table 4 shows the average accuracy of our solutions calculated by the distances.

CONCLUSION AND FUTURE WORKS

By implementing a scheduling framework using standard software components, we have done a great step forward in production scheduling automation. Our approach is based on standard ERP systems as data source and standard CP optimization engine to solve the problem. Multi-objective scheduling is also implemented in it, which puts the theoretical approach into ERP practice. Moreover the configurable, source system independent framework architecture makes it possible to easily integrate our solution with any type of ERP system.

As a future goal the designed and implemented mathematical model of the job-shop scheduling problem can be expanded by elaborating more parameters and constraints in the model, like raw material or machine availability limitations, precedence rules among jobs, or more objective functions.

REFERENCES

- [1] PAVIOT, Thomas; CHEUTET, Vincent; LAMOURI, Samir. A PLCS framework for PDM/ERP interoperability. *International Journal of Product Lifecycle Management*, 2011, 5.2-4: 295-313.
- [2] BOTTA-GENOULAZ, Valerie; MILLET, P.-A.; GRABOT, Bernard. A survey on the recent research literature on ERP systems. *Computers in industry*, 2005, 56.6: 510-522.
- [3] HARJUNKOSKI, Iiro, et al. Scope for industrial applications of production scheduling models and solution methods. *Computers & Chemical Engineering*, 2014, 62: 161-193.
- [4] RAM, Jiwat; WU, Ming-Lu; TAGG, Roger. Competitive advantage from ERP projects: Examining the role of key implementation drivers. *International Journal of Project Management*, 2014, 32.4: 663-675.
- [5] LAFRAMBOISE, Kevin; REYES, Felipe. Gaining competitive advantage from integrating enterprise resource planning and total quality management. *Journal of Supply Chain Management*, 2005, 41.3: 49-64.
- [6] PANG, Chris; DHARMASTHIRA, Yanna; MONTGOMERY, Nigel (2015, May 6). Market Share Analysis: ERP Software, Worldwide, 2014
- [7] MADAPUSI, Arun; D'SOUZA, Derrick. The influence of ERP system implementation on the operational performance of an organization. *International Journal of Information Management*, 2012, 32.1: 24-34.
- [8] DITTRICH, Yvonne; VAUCOULEUR, Sebastien; GIFF, Stephen. ERP customization as software engineering: knowledge sharing and cooperation. *Software, IEEE*, 2009, 26.6: 41-47.
- [9] RODAMMER, Frederick A.; WHITE JR, K. Preston. A recent survey of production scheduling. *Systems, Man and Cybernetics, IEEE Transactions on*, 1988, 18.6: 841-851.
- [10] XIONG, Jian; XING, Li-ning; CHEN, Ying-wu. Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns. *International Journal of Production Economics*, 2013, 141.1: 112-126.
- [11] PINEDO, Michael L. *Scheduling: theory, algorithms, and systems*. Springer Science & Business Media, 2012.
- [12] KACEM, Imed; HAMMADI, Slim; BORNE, Pierre. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 2002, 32.1: 1-13.
- [13] BEHNKE, Dennis; GEIGER, Martin Josef. Test instances for the flexible job shop scheduling problem with work centers. 2012.