New multi-objective methods to solve resource-constraint multi-project scheduling problems in integrated ERP-APS systems

Krisztián Mihály¹, Mónika Kuleszárné Forrai¹ and Gyula Kulesár¹

¹ Department of Information Engineering, University of Miskolc, Miskolc-Egyetemváros, Miskolc 3515, Hungary

krisztian.mihaly@sap.com, aitkfm@uni-miskolc.hu, iitkgy@uni-miskolc.hu

Abstract. This paper presents a new scheduling model and methods to solve multi-project scheduling problems considering strict resource constraints and multiple objective functions simultaneously. The solution approach is based on an integration of enterprise resource planning system (ERP) and an advanced planning and scheduling application (APS). An ERP system software component defines a set of projects with their details (activities, due dates, processing times, precedence constraints, etc.) and the available resources with their capabilities and real constraints. The user can specify many objective functions and their priorities. The goal of this scheduling problem type is to create a detailed execution plan that meets all the hard constraints and gives a near optimal compromised solution from objective functions point of view. In real execution environments, more than one projects may run in parallel, and they may require the same resources. For example, projects may produce the standard type of products, and they use the same manufacturing resources. In different decision-making situations, dynamic priorities may be defined for projects or activities to express the relative importance of items. The paper focuses on modeling and solving of multi-objective multi-project scheduling problems, and we present the integration of ERP and APS systems.

1. Introduction

Field of operation research and project management has a significant history, with many proven results. Great books and articles show approaches to solve different problem types. The valid reason to invest further activities in this research area is the high variety of the real-life problems and the fact which they are proven as NP-Hard problems. Scientists and software engineers are looking for more efficient ways to deal with these problems or enhance the models with new, relevant aspects.

In this paper, we are focusing on the resource-constrained projects where the scheduling result shall complete multi-objective goal functions. Next, to the models, we are presenting a standard, state of the art market leader ERP solution to model and schedule projects and we target different alternatives to enrich the standard functionality.

A project management problem requires planning and scheduling decisions. The planning process focuses on distributing the total project into manageable tasks or activities. The project planner estimates the demand for different types of resources and processing time (duration) of all the tasks of the project. The tasks must be assigned to the resources, and the executions of tasks must be scheduled over time. The project scheduling deals with these decision-making problems. The role of the project scheduler
covers the allocation of the available resources to the project tasks to determine the start time and finishing time of each task or activities [1].

2. Resource Constrained Project Scheduling

The resource-constrained project scheduling problem (RCPSP) is a well-known model describing a special version of project scheduling problems. The problem itself contains a various number of tasks (activity, job) and limited resources (machine, people, executor). The model is describing a single project, which consists of a set $T = \{1, 2, 3, \ldots, n\}$ of different tasks to be executed. Set $T$ may be enhanced with virtual task 0 and $n + 1$ as “project start” and “project end”, respectively. To execute the project a set of resource types $K = \{1, 2, 3, \ldots, m\}$ are defined. Resource type $k$ ($k \in K$) has a limited capacity of $R_k$ at any point in time. Resources are renewable, meaning execution of a task does not decrease the available capacity of the resource type $k$ during the project execution. Execution of task $j$ cannot be break, when execution on defined resources has been started. The capacity of a given resource type is constant for the time horizon.

Each task may define two kinds of constraints:
- If task $j$ ($j \in T$) has immediate predecessor tasks defined by set $P_j$, then all task $i$ ($i \in P_j$) shall be finished before execution of task $j$ is possible.
- If task $j$ ($j \in T$) requires one or more resource they are defined by set of pairs $(r_{j,k})$.

Each task has a pre-determined execution time on resource type $k$, defined by $p_{j,k}$. It is not allowed to define a circle in the predecessor tasks chain. Further constraint for virtual tasks are, that task 0 cannot have predecessor tasks and task $n + 1$ cannot be predecessor for any task. Furthermore, virtual tasks do not require any resource and virtual task processing time is 0.

Let $C_j$ denote the completion time of task $j$ ($j \in T$). A feasible solution can be described as a vector, where each completion time of a task is given by the corresponding $C_j$. Let $A(t) = \{ j \in T | C_j - p_j \leq t \leq C_j \}$ the set of tasks, which are being processed at time $t$. The objective of the RCPSP is to find precedence and resource feasible completion times for all tasks such that the makespan of the project is minimized.

3. Schedule Generation Schemes and Priority Based Rules for RCPSP

One solution approach to solve an RCPSP problem is to generate a feasible schedule with predefined rules. These are called as schedule generation schemes (SGS), and they are the core of further specific implementations. The generation starts from an empty schedule and in each iteration, a feasible schedule is constructed until all tasks of the project are scheduled. The SGS implementations can be distinguished into activity- and time-incrementation based generations. In this paper, we are focusing on the activity-based generation.

Serial SGS is an activity-incrementation based generation scheme. It consists of $g = 1, 2, \ldots, n$ stages. Each stage starts with a feasible solution and represents one task selection with its addition to the actual schedule at the earliest precedence- and resource-feasible completion time.

Two disjoint sets of tasks can be defined. Set $S_g$ contains tasks, which have been already scheduled, set $D_g$ contains tasks which are eligible to be scheduled. Let $R'_k(t) = R_k - \sum_{j \in A(t)} r_{j,k}$ be the remaining capacity of resource type $k$ at time instant $t$ and let $C_g = \{ C_j \ | \ j \in S_g \}$ be the set of all finishing times. Let further $D_g = \{ j \in T \setminus S_g \ | \ P_j \subseteq S_g \}$ be the set of eligible tasks. The serial SGS is given as follows [2]:
Initialization $C_0 = 0, S_0 = \{0\}$

For $g = 1$ to $n$ do

1. Calculate $D_g, C_g, R'_k(t) (k \in K; \ t \in C_g)$
2. Select one $j \in D_g$
3. $E_j = \max_{h \in P_j} \{C_h\} + p_j$
4. $C_j = \min\{t \in [E_j - p_j, LC_j - p_j] \cap C_g \mid r_{j,k} \leq R'_k(t), k \in K, \tau \in [t, t + p_j] \cap C_g\} + p_j$
5. $S_g = S_{g-1} \cup \{j\}$

$C_{n+1} = \max_{h \in P_{n+1}} \{C_h\}$

The algorithm starts from the initial feasible schedule when virtual project start task is scheduled with its completion time 0 and all resources are available with maximum capacity. At the beginning of each stage $g$, the decision set $D_g$, the set of completion times $C_g$ are calculated. Then one activity $j$ is selected from the decision set. Once $j$ has been selected, the earliest precedence feasible completion time $E_j$ is calculated.

4. Multi-project Scheduling Problems

The allocation of resources and scheduling the tasks for multiple projects are non-polynomial (NP) hard problems and they are more difficult than a single project [3].

In the literature, different classical optimization methods have been used to solve the multi-project scheduling problems. Paper [4] proposed a zero-one programming approach and paper [5] introduced an integer programming for generating the schedule and a simulation method for testing the heuristic rules to choose the best schedule. In [6], Deckro et al. formulated the multi-project scheduling problem as a general integer programming model and presented a decomposition approach to solve large size problems. Jolayemi [7] proposed an integer programming approach for multi-project scheduling and considered a special penalty function. These outlined studies give good solutions for small size problems by using traditional optimization approaches. If the problem sizes are middle or large, then the classical methods cannot solve the problems within a reasonable time.

In the literature, we can found several heuristic and meta-heuristic methods for multi-project scheduling. Researchers developed many useful algorithms to generate feasible solutions for multi-project schedules [8]. The goal of this development was to increase the efficiency of the heuristic methods, to extend the scope of the problems with new methods, and reduced the computation time.

Many researchers have applied artificial intelligence methods to solve the multi-project resource-constrained project scheduling problems. For example in [9], Kim et al. proposed a combined genetic algorithm to create the schedule. Their goal was to minimize the total project completion time. Kumanan et al. applied a genetic algorithm based approach for generating the schedule of multi-project scheduling problems [10]. The objective function of the optimization was also the project completion time. Damak et al. also proposed a genetic algorithm variant with a local search strategy to solve the problem considering the project delay minimization as the optimization goal [11].

After reviewing the connected literature, we can conclude that there is a need to develop efficient and flexible methods to improve the quality and robustness of the solutions of resource constrained multi-project scheduling problems.

In real execution environments, multiple projects are running parallelly, and they are not independent of each other. They may have defined requirements of the same resource, or the execution of project tasks may be dependent on the completeness of a task in another project.

As circle in the tasks precedence path is still not allowed, technically it may be possible to restructure multi-project dependencies to independent projects, where connection points between projects are modeled as virtual sub-project start and sub-project end tasks. Each project has an own defined objective
to meet. The project management declares objective functions to be optimized varying in time, like as to achieve the earliest completion time, to minimize the maximum tardiness, to maximize the robustness, to minimize the sum of costs and to maximize the average utilization of resources. An extended, adaptive and effective scheduling software is needed to satisfy these flexible multi-objective requirements of the users.

The remaining part of this paper summarizes our approach, development, and implementation to realize an integrated multi-objective multi-project solver system.

5. Multi-objective Approach

5.1. Schedule generation scheme with multiple priority rules

Multi-objective optimization is an optimization problem when more than one objective function has to be optimized simultaneously.

Serial SGS implementation selects at stage \( g \) an arbitrary task from the decision set \( D_g \). As enhancement of first RCPSP model all tasks may have a predefined priority. At stage \( g \) from decision set \( D_g \) the task with the highest (lowest) priority is selected. Definition of priorities may be calculated during runtime, depending on the actual applied schedule. These dynamic priorities are called as priority rules. For example, latest finishing time (LFT) or minimal slack (MS) based priority rules.

Let us denote priority rule of a task as \( v_j(T) \). Depending on applied and supported priority rules, a set of priority rules can be combined to enforce tasks selection with multiple aspects.

\[ v_j = m_1 \cdot v_{j,1} + m_2 \cdot v_{j,2} + \cdots + m_L \cdot v_{j,L} = \sum_{l=1}^{L} m_l \cdot v_{j,l} \quad (1) \]

The difficulty of this method is to give the appropriate values of the weighting factors. The choices of these values strongly influence the solution. It is not possible to distinguish the normalization and the expression of the importance sharply. Normalization is required to add priorities with different values and dimensions. The importance expression focuses on the role of the different priority rules in the actual decision-making process.

5.1.2. Multiple priorities with the relative comparison

We propose a new approach to choose the task to be scheduled from the decision set in the task selection steps of the schedule generation schemes. A relative rating method has been developed to compare the task of the decision set according to multiple priorities. The priorities are given such that:

\[ v_l : T \rightarrow \mathbb{R}^+_0 \]  

Coefficients \( w_l \) as input parameters support that the user may calibrate the actual importance of each priority rule independently. Each \( w_l \) is an integer value (\( w_l \geq 0 \)) that expresses the importance of \( v_l \).

Let \( t_x, t_y \in T \) be two candidate tasks. The function \( F \) is defined by (3) to express the relative importance of \( t_y \) compared to \( t_x \) as a real number.

\[ F : T^2 \rightarrow \mathbb{R}, \quad F(t_x, t_y) := \sum_{l=1}^{L} w_l \cdot D(v_l(t_x), v_l(t_y)) \quad (3) \]

The function \( D \) defined by (4) means the comparison of \( t_x \) and \( t_y \) according to \( v_l \).
\[
D : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad D(a, b) := \begin{cases} 
0, & \text{if } \max(a, b) = 0 \\
\frac{b - a}{\max(a, b)}, & \text{otherwise}
\end{cases}
\] (4)

The relational operators are overloaded with (5) using the definition (3) of the function F.

\[
(t_y, t_x) := \left( F(t_x, t_y) \equiv 0 \right)
\] (5)

Any of the relational operators can be used between two tasks to compare them like two real numbers. For example: \( t_y \) is less important than \( t_x \), if \( F(t_x, t_y) \) is less than zero.

The most important task can be selected in the actual step of the schedule generation scheme, comparing candidate tasks in an actual decision set. According to our proposal, the relative importance of a task is more expressive than the weighted linear combination of its priorities. The base of the approach is that the relative importance of the candidate task is measured by comparing it with another task in the decision set.

5.2. Multi-objective searching algorithm with relative qualification

Different scheduling goals can occur managing real-life processes or systems. In many cases, for example, it is very important that the task be completed as close as possible to its due date. A task that is tardy causes delays in the processes or projects, while early completion of a task can be damaging if the storage space is limited or this event has an extra cost. Typically, many other objective functions must be considered, and they are usually conflicting. The scheduling problems often involve more than one aspect, so it is required to use multiple criteria simultaneously. For this reason, a general method has been developed to measure the performance of the fine schedules by calculating management indices or performance indicators based on tasks, projects, and resources.

A generalized mathematical model has also been elaborated for a relative qualification that serves for comparing the created schedules according to multiple objectives [12], [13].

According to our proposal, the relative goodness (quality) of a solution is more important than the absolute goodness (quality) of one. The base of the approach is that the relative goodness of the candidate solution is measured by comparing it with another solution in the feasible solution space.

Let \( \mathcal{Y} \) be the solution space. Suppose that the objective functions are given such that:

\[
f_z : \mathcal{Y} \rightarrow \mathbb{R}_0^+, \quad f_z \rightarrow \min. \forall z \in \{1, 2, ..., H\}
\] (6)

Coefficients \( u_z (z = 1, 2, ..., H) \) as input parameters support that the user may calibrate the actual importance of each objective function independently. Each \( u_z \) is an integer value \( (u_z \geq 0) \) that expresses the importance of \( f_z \).

Let \( s_x, s_y \in \mathcal{Y} \) be two candidate solutions. The function \( F \) is defined by (7) similarly to (3), in order to express the relative goodness of \( s_y \) compared to \( s_x \) as a real number.

\[
F : \mathcal{Y}^2 \rightarrow \mathbb{R}, \quad F(s_x, s_y) := \sum_{z=1}^{H} \left( u_z \cdot D(f_z(s_x), f_z(s_y)) \right)
\] (7)

The function \( D \), which is defined before by (4), means the comparison of \( s_x \) and \( s_y \) according to \( f_z \).

Using the re-definition (7) of the function \( F \), the relational operators can also be overloaded by (8).

\[
(s_y, s_x) := \left( F(s_x, s_y) \equiv 0 \right)
\] (8)

In this case, any of the relational operators can be used between two solutions to compare them like two real numbers. For examples: \( s_y \) is better than \( s_x \), if \( F(s_x, s_y) \) is less than zero.

These definitions of the relational operators are suitable for applying in meta-heuristics like taboo search, simulated annealing, evolutionary or genetic algorithms to solve multi-objective multi-project resource-constraint scheduling problems.
Our solver system solves the scheduling problem in two phases. In the first phase, combined heuristic schedule generation schemes and multiple priority rules create solutions. The best solution becomes the initial solution of the second phase. In this second phase, the solver performs iterative improvements by using multi-objective searching methods.

6. Integration of ERP and APS systems

ERP systems are mainly used to manage all resources in an enterprise company. They consist of a set of different modules, responsible for coupled, separated, and integrated application areas, like manufacturing, finance, project management, and so on. We were focusing on SAP ERP system in our work, but our conceptual model can be used by other ERP vendors as well, after adjustment of the implemented artifacts to the vendor-specific platform.

As all standard project management systems, SAP ERP PM supports management of project-related entities, such as the creation of the project, the definition of tasks and dependencies, the definition of resources and their task assignments. In project execution phase the reporting of execution, collaboration with external parties are covered.

In SAP ERP this module is called project management (PM) or project and portfolio management (PPM) [14] [15]. Both implementations follow the same architecture design. Data is persisted in a relational database which is Open SQL compatible; business logic is written in ABAP language; user interfaces are developed for various sets of modeled business roles in different UI technologies, like SAP Dynpro, ABAP WebDynpro or JavaScript-based SAP UI5.

The standard SAP solutions are supporting scheduling of projects. The state-of-the-art implementation supports so-called up-down and bottom-up scheduling algorithms. These algorithms are mainly an extended CPM implementation. Two main directions have been identified to enrich project standard scheduling capabilities with advanced scheduling capabilities.

6.1. APS integration within ERP

SAP standard solution offers a set of extension points to influence system behavior and enhance the business logic. ABAP language and the SAP NetWeaver platform offers a rich set of features to implement a complete new APS module. The concept of the integrated APS is depicted in Figure 1.
The NetWeaver based APS implementation is using the standard project database as source and APS models and configuration used to enrich scheduling capabilities. The APS agent is triggered by the SAP PM or by ERP user. Scheduler reads actual project management and resource booking data, reads the scheduler configuration for calibrated scheduling parameters, may check the historical scheduling information and retrieves for the ERP user a feasible schedule or failure information in case of non-feasible scheduling setup.

Advantages of this approach are to have tight integration with standard PM system, system-to-system communication is not required, and the NetWeaver platform supports all transactional requirements. The disadvantage of the approach is, that solution is SAP platform specific and requires a complete SAP NetWeaver stack.

6.2. Integration with external APS
Integration concept of external APS system usage depicted in Figure 2.

In this concept, APS is a separated system with own stack. ERP and APS system can communicate via predefined channels, like HTTP, RFC, IDoc and so on. The ERP user and the APS user is modeled independently. The ERP system calls the APS system and the execution of an advanced project scheduling consists of the following steps:
- ERP system collects all relevant data required for the scheduling service and maps it to the expected structure. As an alternative, the APS system can call-back for required data by a reverse proxy.
- System-to-system communication is build up, and scheduling request is sent.
- APS system interprets the request, map it to APS models and execute the scheduling algorithms.
- In case of success task schedule is sent as a response, or in case of failure, a proper error code with detailed is replied.
- ERP system interprets the APS response.

The advantage of the approach is that the APS system is ERP system independent. APS and ERP system has a separated software lifecycle. The disadvantage is that scheduling requires system-to-system
communication, which is more resource- and time-consuming compared to an integrated solution. ERP system data is sent over to an external system what is in many cases not acceptable for an ERP customer.

7. Software Development Results
We implemented in SAP NetWeaver platform a standalone APS component. The high-level concept is presented in Figure 3. The user is reaching the APS user interface via SAPGUI. In the user interface the user can select the data set, where the projects can be selected, and scheduling parameters can be calibrated. Once the project scheduler is started, the actual data is read from the data source, the detected scheduling algorithms are executed, and the scheduling results are stored via the scheduling result manager. Scheduling result is displayed in the APS user interface. In the subchapters, the main entities are detailed out on the conceptual compositional level.

![APS System Agents](image1)

**Figure 3. APS system agents**

7.1. Data source
Access to project definition is unified via object-oriented ABAP artifacts. The data source package is separated into internal and external objects. For the consumer, the data source is an interface, which may have a various implementation. All implementations are hidden from consumers, and each implementation can be requested via the data provided factory. Data provider factory instantiates the hidden implementation, based on consumer request.

![Data Source Modeling](image2)

**Figure 4. Data source modeling**

As an example, the data source can be a test data container, used to evaluate scheduling algorithms functional correctness by comparing the project scheduling execution results with expected schedule result. A file system as input can serve a background task-based scheduling. Benchmark data sources can serve comparison of implementation by other solution in regard to quality, execution performance
and correctness. SAP PM is aimed to be the access directly to SAP standard project management database tables or indirectly via SAP PM APIs.

7.2. Project Scheduler

Main development artifacts of the project scheduler are depicted in Figure 5. APS scheduler consists of two disjoint sets of entities. One is visible for APS consumers and the second is hidden from consumers. Disjoint separation aims to offer a stable interface to code against it and be flexible to change the used implementation. The external interface consists of the supported project scheduling problem definition interfaces (CPM, RCPSP), the resource manager read interface and a solver factory.

RCPSP problems without the resource constraints can be solved with the critical path method (CPM) algorithm. CPM result may be an import parameter of a dynamic priority rule calculation, that is why RCPSP problem may require a CPM solver and CPM entity is an individual entry in the model. RCPSP problem solver has different implementation alternatives, and the concept is flexible to change at any time. The reason for this flexibility is to be able to analyze implementation alternatives without any effect on consuming system artifacts.

The implemented solution realizes the generation-based solvers, like SGS serial, the extended generation based solvers as RCPSP heuristic solver and the multi-objective search based solver. The resource manager is responsible for checking the feasibility of scheduling, provided by the scheduling algorithms. The resource manager has two interfaces:

- Write interface for management of resources and management of resource bookings
- Read interface for consumers to read and display scheduling in a user-friendly way

8. Conclusions

In this paper, a new approach is proposed to solve multi-objective multi-project scheduling problems considering strict resource constraints and multiple objective functions simultaneously. The description was planned from a practice-oriented point of view. The novelties of the solutions are based on the special combination of multiple priority rules in schedule generation schemes, and the flexible usage of multi-objective relational operators for comparing qualities of schedules in search algorithms. The proposed methods can consider exactly what the actual resource environment should perform in the
planned time horizon. Each schedule created for the extended problems is a feasible solution because all the hard constraints are considered. The proposed extended model and methods ensure to develop and implement adaptive scheduling software applications for the practice.

Our solver system prototype is based on an integration of enterprise resource planning system and an advanced planning and scheduling application. SAP technologies were used for implementation. The ERP functionalities can preferably be used for managing the projects and resources. In a conflicting decision-making situation, the integrated APS methods solve the actual problems by using the dynamic priorities and relative importance of items. The user can specify the solver settings.

Our research focused on modeling and solving of multi-objective multi-project scheduling problems. The proposed approach and the results are independent of the technical or management areas, so the results can effectively be deployed for multi-project scheduling problems in different fields, such as software development, manufacturing and assembly systems, supply chain management, human resource managements, and so on.

References

[1] Amol, S., 2014, Resource Constrained Multi-Project Scheduling with Priority Rules & Analytic Hierarchy Process, Proc. Eng., 69, pp. 725–34.

[2] Kolisch R, Hartmann S, Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis 1999, Proj. Scheduling. Intern. Series in Op. Res. & Mngng. Scn., vol 14. 147-78.

[3] Garey M R, Johnson D S, Computers and intractability a guide to the theory of NP completeness 1979, W. H. Freeman and Company

[4] Pritsker A, Allan B, Watters L J, Wolfe P M, Multiproject scheduling with limited resources: A zero-one programmingapproach 1969, Mngmnt. Science, 16, pp. 93–108.

[5] Mohanthy R P, Siddiq M K, Multiple projects multiple resources-constrained scheduling: Some studies 1989, Int. Jnl of Prod. Res., 27, pp. 261–80.

[6] Deckro R F, Winkofsky E P, Hebert J E, Gagnon R, A decomposition approach to multi-project scheduling 1991, Europ. Jnl. of Oper. Res., 51, pp. 110–18.

[7] Jolayemi J K, Scheduling of projects under penalty and reward arrangements: A mixed integer programming model and its variants 2012, Acad. of Inf. and Mngmt Science J., 15, pp. 29–52.

[8] Gawiejnowicz S, Lin B M T, Scheduling time-dependent jobs under mixed deterioration 2010, Appl. Math. and Comp., 216, pp. 438-47.

[9] Kim K W, Yun Y S, Yoon J M, Gend M, Yamazaki G, Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling 2005, Comp. & Ind. Eng., 56, pp. 143–60.

[10] Kumanan S, Jegan J G, Raja K, Multi-project scheduling using a heuristic and a genetic algorithm 2006, Int. Jour. of Adv. Manuf. Techn., 31, pp. 360–6.

[11] Damak N J B, Siarry P, Loukil T, Differential evolution for solving multi-mode resource constraint scheduling problems 2009, Comp. and Op. Res., 36, pp. 2653–9.

[12] Kulescár Gy, Erdélyi F, A New Approach to Solve Multi-Objective Scheduling and Rescheduling Tasks 2007, Int. J. of Comp. Int. Res., Vol. 3, No. 4, pp. 343-51.

[13] Kulescár Gy, Kulescárne F M, Detailed production scheduling based on multi-objective search and simulation 2013, Prod. Syst. and Inf. Eng. Vol. 6, pp. 41-56.

[14] SAP SE, SAP Project and Portfolio Management 2015, https://help.sap.com/doc/saphelp_ppm400/4.0.24/en-US/02/4b1a417c4bf423e10000000a155106/frameset.htm

[15] SAP SE, SAP S/4HANA 1709 FPS01, Project System (PS) documentation 2018, https://help.sap.com/viewer/4dd8eb7b1c484b4b93af84d00f60f6b8/1709%20001/en-US/1ad4b65334e6b54ce1000000a174cb4.html