Chapter

Flexible Project Scheduling Algorithms

Zsolt T. Kosztyán

Abstract

Despite the emerging importance of flexible project management approaches, such as agile extreme and hybrid methodologies, the algorithmic support of these approaches is still insufficient. In addition, single project scheduling has received far more attention than have schedules of multilevel projects, such as project portfolios or multi projects. This lack of scheduling techniques is especially true for flexible portfolios, such as agile, hybrid, and extreme project portfolios. While multilevel project scheduling algorithms already exist for fixed multilevel project structures, they are not able to handle flexible structures. This chapter proposes algorithms to schedule both flexible single and multilevel projects. The proposed algorithms handle both flexible and unplanned tasks and dependencies. They handle both single and multimode completion modes, and both renewable and nonrenewable resources. In addition, this chapter proposes a matrix-based risk-valuation framework to evaluate risk effects for flexible projects and portfolios. With this framework, project scheduling approaches are compared.

Keywords: agile, extreme, hybrid projects, schedule, multilevel projects, agents

1. Introduction

Despite the flexibility, such as agile (APMa), hybrid (HPMa), and extreme (EPMa), project management approaches come from the software project environment [16], and they are being increasingly used in nonsoftware environments as well. Flexible project management methods require flexible project scheduling methods, which allow flexible project structure [5]. Because of the time complexity, only a few methods handle the flexible nature of the projects. Nevertheless, in recent years, a new family of flexible scheduling methods [13] has been proposed. Instead of network-based methods, these algorithms are based on domain mapping (DMM) [4] and multidomain mapping matrices (MDMs) [1]. To support the agile project management approach, the proposed matrix-based methods handle both the priority of the task completions and the flexibility of dependencies between tasks [5, 8]. In addition to supporting extreme project management approaches, unplanned tasks can be scheduled [7]. The proposed matrix-based method can also be used to plan traditional, nonflexible projects. In this way, multimode completions can be specified. Nevertheless, the multimode and flexible project structures can also be combined to support hybrid project management approaches. The proposed flexible scheduling algorithms can be used not only in the projects but also at the project portfolio level [6].

The proposed matrix-based algorithm is based on the former studies [5–8]; however, they are unified into the common matrix-based model, and they are extended to handle nonrenewable resources too. The proposed matrix-based model and the scheduling algorithm have the following features.

1. They handle fixed, but also *flexible* dependencies and mandatory but also *supplementary* tasks; therefore, traditional but *agile* project plans can also be modeled.

2. They handle planned, but also *unplanned* tasks, therefore, they can also model *extreme* project plans.

3. They handle single, but *multimode* completion modes; therefore they can also model traditional, but *hybrid* project plans, too.

4. They can handle *renewable*, but *non-renewable* resources, too.

5. They can model single, but also *multilevel* project plans.

In order to keep the adaptability of the proposed method, all features in italic style are optional. It means, when scheduling traditional project plans, there is no need to plan any supplementary tasks, flexible dependencies, or unplanned tasks; however, these features are optionally used in a flexible, such as agile, hybrid, or extreme project planning. Selecting from the multiple completion modes (called discrete technologies) is only relevant if there are more alternative technology. And last but not least, planning in a multilevel project environment is also an optional feature; however, it is crucial in the case of planning the resource sharing within the multi-projects.

In addition to the tasks, the risk effects of project scheduling can also be modeled [10]. The matrix-based risk evaluation has the following features:

1. Risk factors, risk effects, goals, and stakeholders can be modeled in a unified matrix-based model.

2. It handles independent, but also interdependent risk factors.

3. Both the planning and the tracking phases are covered.

All the proposed algorithms have software application support [9]. Therefore, after the model has been developed, the proposed methods can be compared. The MATLAB add-on [9] with examples guides users from the project planning to the risk evaluation.

The aim of this chapter is twofold. First, the chapter shows how to model flexible single and multilevel project plans and their risks. Second, the chapter summarizes the algorithms, so-called project management agents, that simulate a decision maker. In this way, the different project management approaches can compete, and the best one can be selected.

2. Matrix-based project planning models

In terms of scheduling, a project is a set of tasks, which has to be solved,

1. within a time-frame (= time constraint, C_t),

2. within the budget (= cost constraint, C_c),

3. within the renewable/nonrenewable resources ($\mathbf{C}_r, \mathbf{C}_n$), and

4. with adequate quality (C_q),

5. with adequate scope (C_s).

The fixed dependency between (successor and a predecessor) tasks specifies, that a successor task may start if the predecessor task has finished (sequential completion). While flexible dependency, based on a later decision either allows either a sequential or a parallel completion.

In terms of planning and scheduling, the multilevel project is the set of projects. Within a multilevel project, at least two overlapping projects specify a *multiproject* [3], if they have common resources. While at least two projects specify *programs*, if the goals of the projects are similar and they have dependency between them. In terms of scheduling, *project portfolios* can contain single projects, multiprojects, and programs too. While, *multilevel projects* can contain project portfolios, too.

Apart from network planning methods, matrix-based project planning is used to model complex project plans [2]. Matrix-based project planning methods are often based on the design (or dependency) structure matrix (DSM) [14]. The domain mapping matrix (DMM) is an extended version of the DSM, with multiple domains [4]. In this chapter, a modified project-oriented version of a domain mapping matrix (DMM) is used, which is called the project domain matrix (PDM) [5].

The PDM contains two mandatory and four supplementary domains.

LD The logic domain is an n by n matrix, where n is the number of tasks. Each cell contains a value from the [0,1] interval.

TD The time domain is an n by m matrix with positive real values, where m is the number of completion modes[1].

The first mandatory domain is the logic domain. Diagonal values in **LD** represent the priority values of the tasks. If a diagonal value is 0, then this task will not be completed. If the diagonal value is 1, then the task is a mandatory task, while if the diagonal value is between 0 and 1, then it is a supplementary task, which means that depending on the decision, either it will be completed or omitted/postponed. Out-diagonal values represent the dependency between tasks. If an out-diagonal value $[\mathbf{LD}]_{ij} \in \mathbf{LD}$ is 1, then task i precedes task j. In the case of $[\mathbf{LD}]_{ij} = 0$, there is no precedence relation from task i to task j. If $0 < [\mathbf{LD}]_{ij} < 1$, then there is a flexible dependency between task i and task j, which means the dependency is on whether decision task i precedes task j. Since all project networks from the considered databases do not contain any cycle, in other words, they can be ordered topologically, the logic domain of the topologically ordered project networks is an upper triangular (sub) matrix. Formally $[\mathbf{LD}]_{ij} := 0$, if $i > j$. The other mandatory domain of the PDM is the time domain. The positive values of the time domains represent the possible duration of tasks. For each task, k duration values can be assigned; nevertheless, the duration values may also match each other.

The additional supplementary domains are:

CD Cost domain, which is an n by m nonnegative matrix of task costs

QD Quality domain, which is an n by m nonnegative matrix of quality parameters of tasks

---

[1] A task within a project can be solved by different kind of technology, which requires different kind of (time, cost, quality, resource) demands and it has different kind of quality parameters. These technologies are called as *completion modes*.

ND The nonrenewable resource domain is an n by $m \cdot \eta$ nonnegative matrix of nonrenewable resource demands, where η is the number of types of nonrenewable resources.

RD The renewable resource domain is an n by $m \cdot \rho$ nonnegative matrix of renewable resource demands, where ρ is the number of types of renewable resources.

Table 1 shows an example of a fully filled PDM matrix. There are 3 (2 mandatory, 1 supplementary) tasks, 3 (2 fixed, 1 flexible) dependencies, 2 completion modes, 2 nonrenewable resources, and 3 renewable resources. The optional domains can be either ignored or filled out with zero values.

Since PDM can model flexible dependencies and task priorities, it can be used to model both traditional and flexible approaches, such as agile and extreme approaches, and hybrid project planning approaches (see details in Section 3). Nevertheless, handling completion priorities and flexible dependencies alone raises the number of possible project plans.

The project can be organized into a multilevel project. The projects in the applied M[5] (matrix-based multimode multilevel (project) management model) share their domains. **Table 2** shows an example of a multilevel project plan. The common logic domain allows us to plan flexible dependencies both within and between projects. It

PDM	Logic domain			Time domain		Cost domain		Quality domain		Nonrenewable resource domain				Renewable resource domain					
	A	B	C	t_1	t_2	c_1	c_2	q_1	q_2	n_{11}	n_{12}	n_{21}	n_{22}	r_{11}	r_{12}	r_{21}	r_{22}	r_{31}	r_{31}
A	1	1	.6	1	2	3	2	.7	.8	4	5	3	6	3	6	3	5	4	2
B	0	.7	1	2	2	4	3	.8	.9	3	3	4	4	4	7	4	6	4	3
C	0	0	1	4	3	5	3	.9	.8	4	2	5	3	4	2	4	7	6	4

Table 1.
The structure of the project domain matrix.

	(Logic Domain, LD)																	Time Domain (TD)		Cost Domain (CD)		Renewable resource Domain (RD)							
	Project 1					Project 2				Project 3					Project 4														
M[5]	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	t_1	t_2	c_1	c_2	r_{11}	r_{12}	r_{21}	r_{22}	r_{31}	r_{32}	r_{41}	r_{42}
A	0,8	1,0	0,8	0,2	0,1													4	6	2,4	3,4	1	1						
B		1,0		0,4	0,9													2	3	1,8	2,6	1	1						
C			0,9		0,1													4	8	9,6	9,9	1	1						
D				0,4	0,3													10	10	4,2	4,2	1	1						
E					0,7													3	4	0,9	1,2	1	2						
F						1,0	1,0	0,8	0,2									4	4	2,4	2,4			2	2				
G							1,0		0,4									2	2	1,8	1,8			2	2				
H								1,0										4	4	9,6	9,6			1	1				
I									1,0	0,8								10	10	4,2	4,2			2	2				
J										1,0	1,0	0,8	0,2	0,1				3	4	0,9	1,2					1	2		
K											1,0		0,4	0,9				3	4	0,9	1,1					1	1		
L												0,9		0,1				4	6	2,4	3,4					1	2		
M													1,0	0,3				2	3	1,8	2,6					1	1		
N														1,0				4	8	9,6	9,7					1	2		
O															1,0			10	12	4,2	4,0	1	1					1	2
P																1,0	1,0	3	4	0,9	0,9							1	1
Q																	1,0	3	3	0,9	1,0							1	1

Hybrid project (flexible dependencies and task priorities + multimodal completions)
Multimodal completion modes
Time-independent resource demands
Agile project (flexible dependecies + task properties + unimodal completions)
Conditional dependency between projects => Either programmes or multi-projects
Hybrid project pan
Traditional project plan (Strict dependency & Same priority + multi-model completions)
Common resource demands

Table 2.
Matrix-based multimode multilevel (project) management model.

DOI: http://dx.doi.org/10.5772/intechopen.101838

(a) Single PSM

P S M	Logic Domain			TD	CD	QD	ND		RD		
	A	B	C	t	c	q	n_1	n_2	r_1	r_2	r_3
A	X	X	X	1	3	.7	4	3	3	3	4
C		X		2	4	.8	3	4	4	4	4
C			X	4	5	.9	4	5	4	4	6

(b) Multilevel PSM

P	(Logic Domain, LD)																	TD	CD	RD			
S	Project 1					Project 2				Project 3					Project 4								
M	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	t	c	r_1	r_2	r_3	r_4
A	X	X	X															4	2,4	1			
B		X																2	1,8	1			
C			X															4	9,6	1			
D				X														10	4,2	1			
E					X													3	0,9	1			
F						X	X	X										4	2,4		2		
G							X											2	1,8		2		
H								X										4	9,6		1		
I									X	X								10	4,2		2		
J										X	X	X						3	0,9			1	
K											X			X				3	0,9			1	
L												X						4	2,4			1	
M													X					2	1,8			1	
N														X				4	9,6			1	
O															X			10	4,2	1			1
P																X	X	3	0,9				1
Q																	X	3	0,9				1

Table 3.
Single and multilevel PSM.

handles the different completion modes; therefore, all the traditional, hybrid, and agile project plans can be planned (see **Table 2**).

Because of the numerous possible project plans, there is no chance to compute all possible projects or multilevel project plans. A fast, exact method is required to find the best project or multilevel project structure.

After deciding which tasks are completed, which flexible dependency is required, and which completion mode is selected, we obtain a (multilevel) project plan. **Table 3** shows, a single (a) and a multilevel (b) project schedule matrix (PSM).

Table 3(a) shows a project schedule matrix of PDM (see **Table 1**), which contains 6 domains. **Table 3(b)** shows a possible project schedule M^5 matrix representation of a hybrid multilevel project (see **Table 2**), where there are 4 domains. In both models, empty cells represent 0 values, and 'X' represents 1 value. 'X' also indicates that this value is the result of a decision. The decision is always binary: either include or exclude a task or a dependency. The PSM contains only one mode, i.e., the selected completion mode for every task. PSM optionally contains the scheduled starts (SS) of tasks. Otherwise, the default start is the early start (ES) of tasks.

3. Project management agents

Previously, [16] found that in his study of the practices of software project managers, only 20% of IT projects were managed by a traditional project management (TPM) methodology. Generally, methods for investment and construction projects cannot be directly applied to software development or R&D projects, as these are managed by agile project management (APM) approaches. Currently, hybrid (i.e., combinations of traditional and agile and extreme) approaches are becoming increasingly popular [11]. However, flexible approaches are thus far not privileges for software development projects [15]. Rapidly changing environments increasingly enforce flexible approaches. Project planning and scheduling algorithms can support decision makers in managing projects; however, there are only a few algorithmic procedures that can support flexible approaches. Therefore, it is important to study how to extend project planning and scheduling methods to handle flexible and changing environments. Planning and scheduling methods, as agents, can also imitate decision makers; therefore, not only the methods but also the scheduling and project planning approaches can be modeled (see **Table 4**).

While, a project manager who follows a traditional project management (TPM) approach can use tradeoff or multimode methods to reduce task duration or cost/ resource demands, an agile and extreme project manager tries to restructure the project. If the project structure is flexible (see **Table 4**), then the project duration can be reduced without increasing the project cost by reducing the number of flexible dependencies. In addition, in real project situations, decision makers can choose from different kinds of technologies (i.e. completion modes); therefore, the TPM and APM approaches can be integrated. Agile approaches usually split the projects into smaller so-called "sprints" that are usually 2–6 weeks. The content of sprints is specified by the customer and developers together. However, when running a sprint, unplanned new tasks and new requirements can be involved only until the next sprint. The extreme project management (EPM) approach handles the new tasks and new requirements during the implementation of the project. Extreme project management can confirm the extra costs and the increased project duration due to the extra tasks.

Flexible approaches require flexible project structures; however, in addition to the opportunity to reorganize the project, different kinds of technology (completion modes) should also be considered; therefore, traditional and flexible approaches should be combined into hybrid project management approaches [11, 12, 15]. Nevertheless, hybrid approaches should be supported by algorithmic methods to help decision makers ensure the project's success.

There are different combinations of agile and traditional project management approaches [11, 12, 15]. However, there are very few exact algorithms (see, e.g., [7, 8]) that can be used to solve hybrid multimode problems that can handle unplanned tasks and dependencies. Nevertheless, R&D and IT projects, such as introducing and setting up new information systems, may require reorganizing part of the project, and R&D projects may require handling unplanned tasks, particularly in the development phase. However, decreasing the time demands of mandatory tasks and those of the new unplanned tasks may also be an important requirement. Neither the agile approach, nor the extreme approach can handle this situation properly, nor can traditional approaches. Traditional approaches, or network-based methods, assume static logic plans, but the reorganization of projects may produce insufficient reductions in project duration and/or supplementary tasks, and important tasks may be excluded from the project due to budget constraints and/or project deadlines. A hybrid project management (HPM) approach can combine traditional, agile, and extreme approaches; however, these kinds of

Planning approaches	Features			
	Project structure	**New tasks**	**Multimode**	**Constraints**
Traditional (TPM)	Fixed	Not allowed	Handled	Fixed
Agile (APM)	Flexible	Not allowed	Not handled	Fixed
Extreme (EPM)	Flexible	Allowed	Not handled	Flexible
Hybrid (HPM)	Flexible	Allowed	Handled	Optional

Table 4.
Comparison of the traditional and flexible approaches.

HPM approaches are not yet supported by project planning methods. The proposed algorithm combines agile, extreme, and traditional approaches. This method extends the traditional multi-mode resource-constrained project scheduling problem by allowing for the restructuring and reorganizing of projects and handling of unplanned new tasks.

The proposed hybrid time–cost and hybrid time-quality-cost tradeoff models [8] and multimode methods [7] manage flexible project plans and allow us to restructure or reorganize these project plans to satisfy customer and management claims. In contrast to the traditional project scoring and selection methods, there is no need to specify all project alternatives to select the most desirable project scenario or the one with the shortest duration or lowest cost. The following definition specifies a matrix representation of a flexible (multilevel) project plan and its possible realization.

Definition 1. Denotes $\{\cdot\}$ as a supplementary domain, which can be an empty matrix. Denote $\mathbf{M} = [\mathbf{LD}, \mathbf{TD}, \{\mathbf{CD}\}, \{\mathbf{QD}\}, \{\mathbf{ND}\}, \{\mathbf{RD}\}]$ as a flexible (multilevel) project plan, where $\mathbf{LD}$ and $\mathbf{TD}$ are mandatory but $\mathbf{CD}, \mathbf{QD}, \mathbf{ND}, \mathbf{RD}$ are supplementary domains. $\mathbf{M}'' = [\mathbf{LD}'', \mathbf{TD}'', \{\mathbf{CD}''\}, \{\mathbf{QD}''\}, \{\mathbf{ND}''\}, \{\mathbf{RD}''\}]$ is a realized (fixed) (multilevel) project plan of $\mathbf{M}$ if $\mathbf{M}''$ is at least an n by $n+1$ but maximum n by $n+3+\eta+\rho$ matrix with two mandatory and six supplementary domains. The following properties are satisfied: if m denotes the number of nodes, ρ denotes the number of renewable resources, η denotes the number of nonrenewable resources, and p denotes the number of projects, then.

$\mathbf{LD}''$: n by n logic domain, where $\mathcal{L}''_{i_k,j_s} = [\mathbf{LD}'']_{i_k,j_s} \in \{0,1\}$.. $\mathcal{L}''_{i_k,j_s} = \mathcal{L}_{i_k,j_s} = [\mathbf{LD}]_{i_k,j_s}$, if $\mathcal{L}_{i_k,j_s} \in \{0,1\}$ and either $\mathcal{L}''_{i_k,j_s} = 1$ or $\mathcal{L}''_{i_k,j_s} = 0$, if $0 < \mathcal{L}_{i_k,j_s} < 1, k, j = 1, .., p$.

$\mathbf{TD}''$: n by 1 column vector (time domain), where $\mathcal{T}''_i = [\mathbf{TD}'']_{i_k} = \mathcal{T}_{i_k,\omega_{i_k}} = [\mathbf{TD}]_{i_k,\omega_{i_k}}$, and $i_k = 1, 2, .., n_k, \omega_{i_k} \in \{1, 2, .., m\}, k = 1, 2, .., p$.

$\mathbf{CD}''$: n by 1 column vector (cost domain), where $\mathcal{C}''_i = [\mathbf{CD}'']_{i_k} = \mathcal{C}_{i_k,\omega_{i_k}} = [\mathbf{CD}]_{i_k,\omega_{i_k}}$, and $i_k = 1, 2, .., n_k, \omega_{i_k} \in \{1, 2, .., m\}, k = 1, 2, .., p$.

$\mathbf{QD}''$: n by 1 column vector (quality domain), where $Q''_i = [\mathbf{QD}'']_{i_k} = \mathcal{Q}_{i_k,\omega_{i_k}} = [\mathbf{QD}'']_{i_k,\omega_{i_k}}$, and $i_k = 1, 2, .., n_k, \omega_{i_k} \in \{1, 2, .., m\}, k = 1, 2, .., p$.

$\mathbf{ND}''$: n by η nonrenewable resource domain, where $\mathcal{N}''_{i_k,w} = [\mathbf{RD}'']_{i_k,w} = \mathcal{N}_{i_k,m\cdot(w-1)+\omega_{i_k}} = [\mathbf{ND}]_{i_k,m\cdot(w-1)+\omega_{i_k}}$, and $i_k = 1, 2, .., n_k, \omega_i \in 1, 2, .., m, k = 1, 2, .., p, w \in \{1, .., \eta\}$.

$\mathbf{RD}''$: n by ρ renewable resource domain, where $\mathscr{R}''_{i_k,r} = [\mathbf{RD}'']_{i_k,r} = \mathscr{R}_{i_k,m\cdot(r-1)+\omega_{i_k}} = [\mathbf{RD}]_{i_k,m\cdot(r-1)+\omega_{i_k}}$, and $i_k = 1, 2, .., n_k, \omega_i \in 1, 2, .., m, k = 1, 2, .., p, r \in \{1, .., \rho\}$.

Definition 1 proposes a unified matrix-based model, both for single and multilevel project plans, and both for single and multimode completions. In addition, by increasing n allows involving the unplanned tasks.

3.1 Demands

Definition 2. Let $\mathbf{M}'' = [\mathbf{LD}'', \mathbf{TD}'', \{\mathbf{CD}''\}, \{\mathbf{QD}''\}, \{\mathbf{ND}''\}, \{\mathbf{RD}''\}]$ be a matrix representation of the realized (multilevel) project, which contains $p > 1$ projects. Assume that $i_k < j_k \Rightarrow [\mathbf{LD}'']_{i_k,j_k} = 0$. Denote $SF_{i_k} = SS_{i_k} + \mathcal{T}_{i_k}$ as the scheduled finish time of task i_k from project k, where SS_{i_k} is the scheduled start time and t_{i_k} is the duration of task i from project k. Denote $R_{r,k}(\tau)$ as the maximum resource demand of renewable resource r for project k at a time τ. Denote τ_{0_k} as the start time of project k and τ_0 as the start time of the multilevel project. The total project values are defined as follows:

TPT: Total project time $TPT_k\mathbf{M}''$) of project k and the duration of the multilevel project ($TPT\mathbf{M}''$).

$$TPT_k\mathbf{M}'' = \max_{i_k} SF_{i_k}, \tag{1}$$

$$TPT\mathbf{M}'' = \max_{i} SF_i. \tag{2}$$

TPC: Total project cost ($TPC_k\mathbf{M}''$) of project k and the total cost of the multilevel project ($TPC\mathbf{M}''$).

$$TPC_k\mathbf{M}'' = \sum_{i_k} [\mathbf{CD}'']_{i_k}, \tag{3}$$

$$TPC\mathbf{M}'' = \sum_{k} TPC_k\mathbf{M}'' \tag{4}$$

TPQ: (relative) total project quality ($TPQ_k\mathbf{M}''$) of project k and the total quality of the multilevel project ($TPQ\mathbf{M}''$).

$$TPQ_k\mathbf{M}'' = \sum_{i_k} [\mathbf{QD}'']_{i_k} / \max_{m} [\mathbf{QD}]_{i_{m,k}}, \tag{5}$$

$$TPQ\mathbf{M}'' = \sum_{k} TPQ_k\mathbf{M}'' / \max_{m} [\mathbf{QD}]_{i_{m,k}} \tag{6}$$

where m is the completion mode.

TPN: Total project nonrenewable resource demands j ($TPN_{e,k}\mathbf{M}''$) of project k and the total nonrenewable resource e of the multilevel project ($TPN_e\mathbf{M}''$).

$$TPN_{e,k}\mathbf{M}'' = \sum_{i_k} [\mathbf{ND}'']_{i_k}, \tag{7}$$

$$TPN_e\mathbf{M}'' = \sum_{k} TPN_{e,k}\mathbf{M}'' \tag{8}$$

TPR: Total project renewable resources ($TPR_{r,k}\mathbf{M}''$) of project k and the total project resources of the multilevel project for resource r ($TPR_r\mathbf{M}''$).

$$TPR_{r,k}\mathbf{M}'' = \max_{\tau_{0_k} \le \tau \le TPT_k\mathbf{M}''} R_{r,k}\tau, \tag{9}$$

$$TPR_r\mathbf{M}'' = \max_{\tau_0 \le \tau \le TPT\mathbf{M}''} R_r\tau \tag{10}$$

TPS: Total project score ($TPS_k\mathbf{M}, \mathbf{M}''$) of project k and the total score of the multilevel project ($TPS\mathbf{M}, \mathbf{M}''$).

$$TPS_k\mathbf{M}, \mathbf{M}'' = \sum_{l''_{i_k,i_k}=1} l_{i_k,i_k}, \tag{11}$$

$$TPS\mathbf{M}, \mathbf{M}'' = \sum_{l''_{i,i}=1} l_{i,i}. \tag{12}$$

3.2 Relative constraints

Definition 3. Denote $C_X, X \in \{T, C, Q, N, R, S\}$ as the time, cost, quality, nonrenewable resource, renewable resource, and score constraints, respectively. Denote $C_X\% = \frac{C_X - TPX_{\min}}{TPX_{\max} - TPX_{\min}}$ as the relative constraints.

It is important to note that the relative constraint should be within the [0,1] interval ($C_X\% \in [0,1]$) to find a feasible solution. Nevertheless, the minimal and maximal demands ($TPX_{\max}, TPX_{\min}$) can be calculated without specifying all possible solutions (see Section 3.4).

(C_{X_k}) constraints can be defined not only for project $k = 1, .., p$ but also for multilevel projects.

3.3 Target function

Either simple or composite target functions can be specified both for single and multilevel projects.

Simple target functions:

$$TPT_k \to \min, TPT \to \min. \tag{13}$$

$$TPC_k \to \min, TPC \to \min. \tag{14}$$

$$TPQ_k \to \max, TPQ \to \max. \tag{15}$$

$$TPN_{e,k} \to \min, TPN_e \to \min. \tag{16}$$

$$TPN_{r,k} \to \min, TPN_r \to \min. \tag{17}$$

$$TPS_k \to \max, TPS \to \max. \tag{18}$$

where $k = 1, .., p, e = 1, .., \eta, r = 1, .., \rho$.

The composite target function handles all possible targets with their importance:

$$\prod_{X \in \{T,C,N,R\}} \left(\frac{TPX - TPX_{\min}}{TPX_{\max} - TPX_{\min}} \right)^{v_X} \cdot \prod_{X \in \{Q,S\}} \left(\frac{TPX_{\max} - TPX}{TPX_{\max} - TPX_{\min}} \right)^{v_X} \to \min \tag{19}$$

where v_X is the weight of the importance of demands ($\sum_X v_X := 1$).

3.4 Main properties of the exact evaluation

Due to the size constraints, only the main feature of the proposed algorithm is summarized. See the details in [6, 8].

1. The evaluation contains three steps.

    a. First, the diagonal values of the **LD** are evaluated. All supplementary tasks ($0 < [\mathbf{LD}]_{ii} < 1$) are decided either to exclude from or to include in the (multilevel) project.

    b. In the second step, all flexible tasks ($0 < [\mathbf{LD}]_{ij} < 1, i > j$) are evaluated. They are either excluded or included.

    c. In the third step, all completion modes are evaluated. For every task, a completion mode is selected.

2. For every kind of multilevel project plan, the minimal (maximal) demands can be specified without calculating all possible solutions.

a. If all supplementary tasks are excluded (included), the *TPS* is minimal (maximal).

b. If all supplementary tasks are excluded (included), and all completion modes require minimal (maximal) demands, *TPC* and *TPN* are minimal (maximal).

c. If all supplementary tasks are excluded (included), and all dependencies are excluded (included) and all completion modes require minimal (maximal) demands, *TPT* is minimal (maximal).

d. If all supplementary tasks are excluded (included), but all dependencies are included (excluded) and all completion modes require minimal (maximal) demands, *TPR* is minimal (maximal).

3. Due to the evaluation, if the minimal (maximal) demands are greater (lower) than the constraint, neither the project plan nor their derived plans are feasible.

Based on these properties, exact back and forth algorithms are proposed to find a single project schedule [7] or multilevel project structure [6]. All methods contain

(a) Original PDM matrix

PDM	Logic Domain					Time Domain			Cost Domain			Quality Domain			Resource Domain					
	A	**B**	**C**	**D**	***E***	t_1	t_2	t_3	c_1	c_2	c_3	q_1	q_2	q_3	r_{11}	r_{21}	r_{31}	r_{12}	r_{22}	r_{32}
A	1.0	1.0	0.8	0.2	*0.0*	4	5	6	2	4	3	0.8	0.9	0.8	2	3	0	3	0	3
B	0.0	1.0	0.0	0.4	*0.0*	2	3	1	3	3	5	0.8	0.9	0.9	3	5	2	2	2	1
C	0.0	0.0	0.9	0.0	*0.4*	8	6	4	5	5	4	0.7	0.9	0.9	0	1	0	1	0	2
D	0.0	0.0	0.0	0.4	*0.0*	9	9	9	7	7	7	0.9	0.9	0.9	2	2	2	0	0	0
E	*0.0*	*0.0*	*0.0*	*0.0*	*1.0*	*1*	*2*	*3*	*3*	*3*	*2*	*1.0*	*1.0*	*1.0*	*2*	*2*	*1*	*2*	*2*	*1*
Constraints: C_s = 3.3						C_t = 10			C_c = 15			C_q = 0.75			C_{r_1} = 3			C_{r_2} = 4		

(b) Phase 1 - task selection

PDM	Logic Domain					Time Domain			Cost Domain			Quality Domain			Resource Domain					
	A	**B**	**C**		***E***	t_1	t_2	t_3	c_1	c_2	c_3	q_1	q_2	q_3	r_{11}	r_{21}	r_{31}	r_{12}	r_{22}	r_{32}
A	X	1.0	0.8		*0.0*	4	5	6	2	4	3	0.8	0.9	0.8	2	3	0	3	0	3
B	0.0	X	0.0		*0.0*	2	3	1	3	3	5	0.8	0.9	0.9	3	5	2	2	2	1
C	0.0	0.0	X		*0.4*	8	6	4	5	5	4	0.7	0.9	0.9	0	1	0	1	0	2
				∅																
E	*0.0*	*0.0*	*0.0*		*X*	*1*	*2*	*3*	*3*	*3*	*2*	*1.0*	*1.0*	*1.0*	*2*	*2*	*1*	*2*	*2*	*1*
Constraints:						C_t = 10			C_c = 15			C_q = 0.75			C_{r_1} = 3			C_{r_2} = 4		

(c) Phase 2 - dependency selection

PDM	Logic Domain					Time Domain			Cost Domain			Quality Domain			Resource Domain					
	A	**B**	**C**		***E***	t_1	t_2	t_3	c_1	c_2	c_3	q_1	q_2	q_3	r_{11}	r_{21}	r_{31}	r_{12}	r_{22}	r_{32}
A	X	X	X		∅	4	5	6	2	4	3	0.8	0.9	0.8	2	3	0	3	0	3
B	∅	X	∅		∅	2	3	1	3	3	5	0.8	0.9	0.9	3	5	2	2	2	1
C	∅	∅	X		∅	8	6	4	5	5	4	0.7	0.9	0.9	0	1	0	1	0	2
				∅																
E	∅	∅	∅		*X*	*1*	*2*	*3*	*3*	*3*	*2*	*1.0*	*1.0*	*1.0*	*2*	*2*	*1*	*2*	*2*	*1*
Constraints:						C_t = 10			C_c = 15			C_q = 0.75			C_{r_1} = 3			C_{r_2} = 4		

(d) Phase 3 - completion mode selection

PSM	Logic Domain				TD	CD	QD	RD	
	A	**B**	**C**	***E***	*t*	*c*	*q*	r_1	r_2
A	X	X	X	∅	4	2	0.8	2	3
B	∅	X	∅	∅	1	5	0.9	2	1
C	∅	∅	X	∅	4	4	0.9	0	2
E	∅	∅	∅	*X*	*1*	*3*	*1.0*	*1*	*1*
Codified constraints:					21	22	0.75	3	4

(e) Total project demands

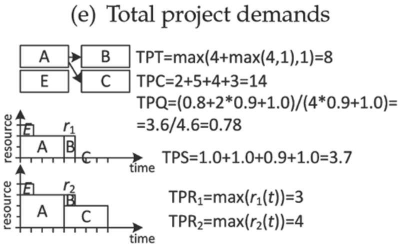


Table 5.
The phases of the computation.

three phases. In the first phase, binary decisions select supplementary tasks to include the project plan. The excluded tasks' demands and dependencies (i.e., rows and columns in the PDM) are also ignored. The result is a *project scenario*, which still contains flexible dependencies, but it is decided that all tasks will be completed. Phase two decides flexible dependencies to include or exclude, or in other words, the flexible dependency between tasks either to be specified or to be ignored. A single (multilevel) *project structure* is the result of this phase, which contains only fixed dependencies. In the last phase, we obtain traditional project plans, which must be solved by standard project scheduling algorithms. The result of this phase is the single (multilevel) *project schedule*, where the completion modes and start times of all tasks are specified.

Table 5 shows an example for the computation process of a single project, where the target function is the minimal project duration. Task E is unplanned, which means, the constraints are specified before task E occurs. **Table 5(a)** shows the original project plan, where there are 3 completion modes, 2 renewable resources, cost demands, and quality parameters. There is no nonrenewable resource, however, the cost demand can be considered as a special nonrenewable resource. While keeping constraints, the algorithm has to find a minimal project duration.

The minimal TPT occurs if all flexible tasks and dependencies are excluded, however, neither the scope nor the quality constraint does not allow to exclude all tasks (see **Table 5(b)**), but only task D. In addition, all flexible dependencies cannot be excluded. For example, in the case of the parallel execution of task A and task C, the resource constraint cannot be kept (see **Table 5(c)**). The algorithm excludes the infeasible structures in phases 1–2. The result of phase 2 provides a multimode resource-constrained project scheduling problem, which can be solved by traditional scheduling algorithms. The final result is a project schedule matrix (PSM) (see **Table 5(d)**), where both the structure of the project plan and its demands are specified (see **Table 5(e)**).

4. Risk evaluation

Kosztyán et al. [10] proposed a flexible matrix-based method for risk evaluation for single projects, where all the *risk factors*, such as changes of durations, changes of resource demands, or even the changes of the task priorities; *risk effects*, such as delays, overbudgets, etc.; *stakeholders*, such as managers, vendors and developers and their *goals*, such as minimal project duration, minimal costs, maximal quality, etc., and their inter-dependencies can be modeled. Nevertheless, phases of risk evaluation can be extended to the multiproject level.

The proposed survival analysis-based risk evaluation (SABER) contains three stages (or phases). In all phases, the feasibility of the projects is checked. *A project plan is a surviving project plan if it is still feasible at the end of the risk evaluation process.* The evaluation process covers the preparatory, planning, and tracking stages. With SABER, the agents can be compared and competed with each other. The decision maker decides whether a traditional approach, such as TPMa, or a flexible approach, such as APMa, EPMa, or even the hybrid (HPMa) approach, should be applied. The SABER contains the following stages:

1. Since before starting projects, the boundary conditions of the project are agreed upon, at *stage one*, the effects of changing constraints are examined.

2. At stage two, a two-step Monte Carlo Analysis (MCA) is applied. In the first step, the set of tasks and the relationship of risk factors are selected, while in the second step, the changes in demands and priorities are changed for the selected tasks.

3. At *stage three*, the two-step MCA is also applied, but only for the remaining and running tasks of running projects.

Figure 1 shows the simulation framework of the SABER.

The simulation framework indicates, which project plans survive. It shows the performances of applied agents and sensitivities of risk effects, and in addition, the interdependency of risk factors and risk effects, see in detail in [10].

5. Computer applications

There are free available matrix-based project planning tools for flexible projects [9]. This plug-in can be applied both for project planning and for risk evaluation. It contains 5 domains, such as LD, TD, CD, QD, and RD. It solves several project scheduling problems (PSPs) for flexible projects, such as Pareto-optimal (multimode) resource-constrained PSPs for a single target function and their Pareto-optimal solutions for multiple targets.

6. Application examples

The application example guides us through matrix-based planning and risk evaluation phases.

The first step is to specify a matrix-based project plan. It can come from the original network-based project plan, but it can be generated. In this case, at least the number of tasks, number of modes, and number of resources have to be specified.

Figure 2(a) shows the logic structure of the flexible project plan. The minimal structures and their demands contain only mandatory tasks and fixed dependencies, while the maximal structure and its demands save all tasks and dependencies, and therefore, all demands. **Figure 2** shows the minimal/maximal demands.

Figure 2 shows that there are significant differences between the minimal and maximal structures and their demands. Flexible and hybrid approaches can

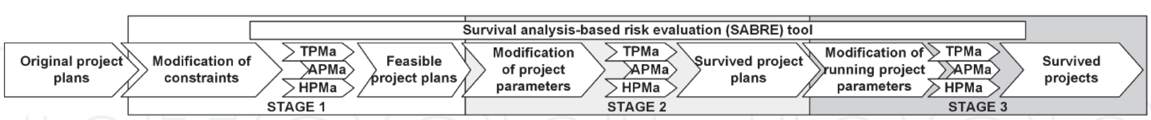
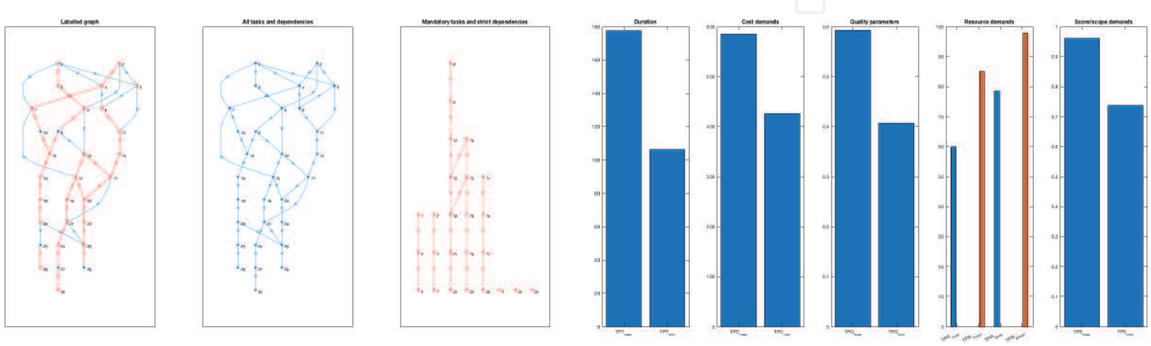


Figure 1.
The simulation framework.



(a) Logic structures (b) Minimal/maximal project demands

Figure 2.
Minimal/maximal structures and demands (the number of tasks was 30, and the number of completion modes and the number of renewable resources was 2).

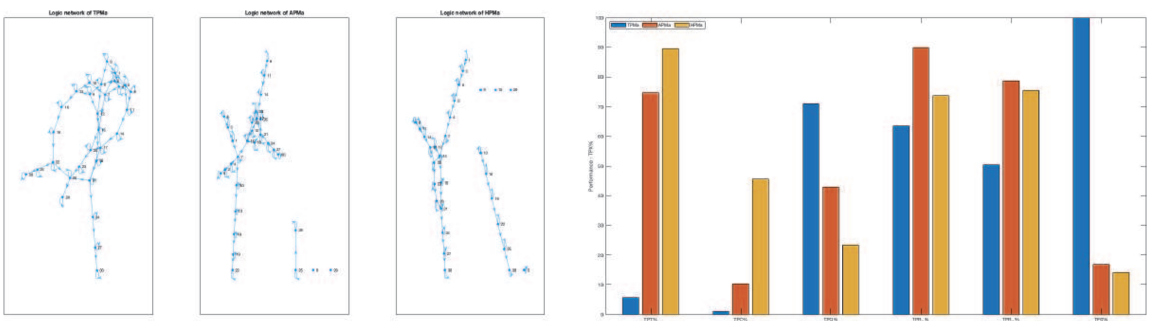
reorganize project structures and reprioritize task completion to save the project plan as feasible. **Figure 3(a)** shows the result project structures of the project management approaches. **Figure 3(b)** compares the project demands. The target function was the minimal distance from the minimal demands, while the constraint was 2/3 of the maximal demands. The vertical axis of 3(b) is the performance ($TPX\%$) of the project plans, which is the relative distance between the minimal demands.

$$TPX\% = \frac{TPX - TPX_{\min}}{TPX_{\max} - TPX_{\min}} \tag{20}$$

If $TPX\% \in [0, 1]$ for all demands, then the project plan is feasible.

Figure 3 shows that the flexible and hybrid project management approaches try to parallelize task completions and exclude low priority tasks, while TPMa tries to reduce demands but keeps all tasks.

Figure 4 shows the comparison of scheduling performances of the project management approaches under risks. **Figure 4** shows that in this case, only the HPMa ensures the survival of the project plan.



(a) Logic networks (b) Project demands

Figure 3.
The comparison of project management approaches.

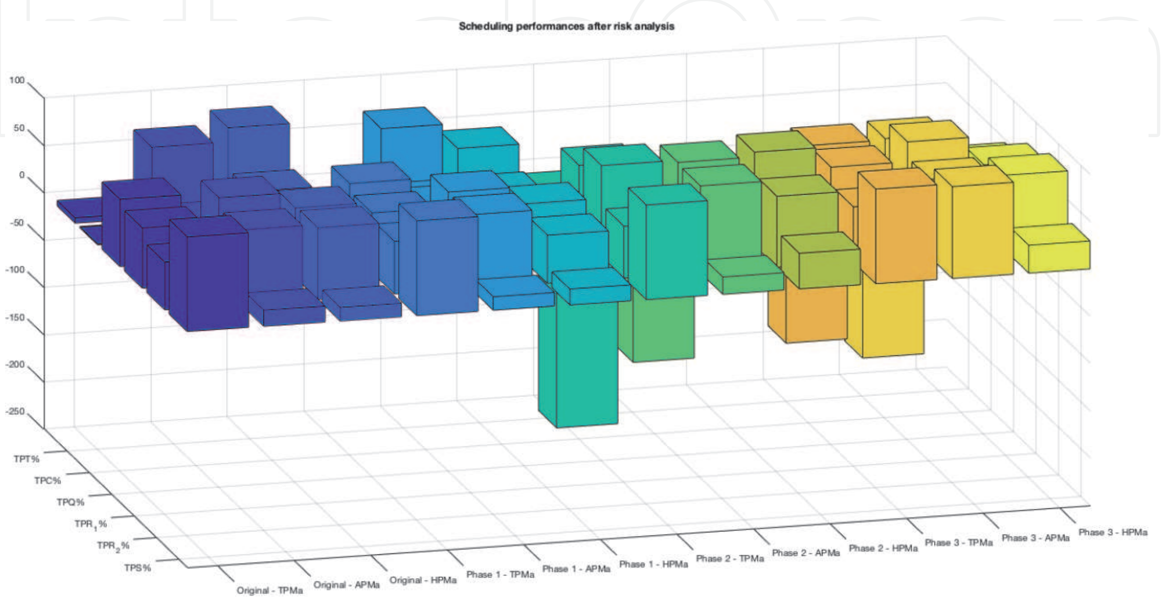


Figure 4.
The scheduling performances of project management approaches under risks.

7. Conclusions

In this chapter, matrix-based approaches are proposed to schedule traditional, flexible, and hybrid project plans. In addition, a risk evaluation tool is proposed to compare project management approaches. It is important to note that there is usually no superior project management approach. If the goal is to complete all tasks, traditional approaches are required; however, flexible projects require flexible project management approaches. Nevertheless, hybrid approaches can better ensure the survival of the project. The study also offers freely available matrix-based project planning applications; therefore, all traditional, flexible, and hybrid project management, planning and scheduling approaches can be supported.

Conflict of interest

The authors declare no conflict of interest.

Nomenclature

CD	cost domain
LD	logic domain
ND	nonrenewable resource domain
QD	quality domain
RD	renewable resource domain
TD	time domain

Variables

n	number of tasks
m	number of completion modes
p	number of projects
η	number of nonrenewable resources
ρ	number of renewable resources
τ	actual time of a running project

Constraints

C_t	time constraint (upper bound, scalar)
C_c	cost constraint (upper bound, scalar)
C_q	quality constraint (lower bound, scalar)
$\mathbf{C}_n$	nonrenewable resource constraint (upper bound, 1 by η vector)
$\mathbf{C}_r$	renewable resource constraint (upper bound, 1 by ρ vector)
C_s	score/scope constraint (lower bound, scalar)

Simple target functions

$TPT \to \min$	minimize project duration
$TPC \to \min$	minimize project cost

$TPQ \to \max$	maximize project quality
$TPN_e \to \min$	minimize the sum of nonrenewable resources e, $e = 1, .., \eta$
$TPR_r \to \min$	minimize the maximum of renewable resources r, $r = 1, .., \rho$
$TPS \to \max$	maximize project score/scope

Abbreviations

APM(a)	agile project management (agent)
DMM	domain mapping matrix
DSM	design/dependency structure matrix
EF	early/earliest finish time
EPM(a)	extreme project management (agent)
ES	early/earliest start time
HPM(a)	hybrid project management (agent)
LF	late/latest finish time
LS	late/latest start time
M^5	matrix-based multimode multilevel (project) management model
MCA	Monte Carlo analysis [MDM] multidomain matrix
MPE	reverse extreme project management
MPR	multilevel project ranking algorithm
NDSM	numerical dependency structure matrix
PDM	project domain matrix
PSP	project scheduling problem
PSM	project schedule matrix
SABER	survival analysis-based risk evaluation
SF	scheduled finish time
SS	scheduled start time
TPC	total project cost
TPM(a)	traditional project management (agent)
TPN	total project nonrenewable resources
TPQ	total project quality
TPR	total project renewable resources
TPS	total project scenario score
TPT	total project time

Author details

Zsolt T. Kosztyán
University of Pannonia, Veszprém, Hungary

*Address all correspondence to: kosztyan.zsolt@gtk.uni-pannon.hu

References

[1] Browning TR. Design structure matrix extensions and innovations: A survey and new opportunities. IEEE Transactions on Engineering Management. 2015;**63**(1):27-52. DOI: 10.1109/TEM.2015.2491283

[2] Chun-Hsien C, Fu LS, Wei C. Project scheduling for collaborative product development using DSM. International Journal of Project Management. 2003;**21**(4):291-299. DOI: 10.1016/S0263-7863(02)00023-6

[3] Creemers S. Minimizing the expected makespan of a project with stochastic activity durations under resource constraints. Journal of Scheduling. 2015;**18**(3):263-273. DOI: 10.1007/s10951-015-0421-5

[4] Danilovic M, Browning TR. Managing complex product development projects with design structure matrices and domain mapping matrices. International Journal of PROJECT MANAGement. 2007;**25**(3):300-314. DOI: 10.1016/j.ijproman.2006.11.003

[5] Kosztyán ZT. Exact algorithm for matrix-based project planning problems. Expert Systems with Applications. 2015;**42**(9):4460-4473. DOI: 10.1016/j.eswa.2015.01.066

[6] Kosztyán ZT. An exact algorithm for the flexible multilevel project scheduling problem. Expert Systems With Applications. 2020;**158**:113485. DOI: 10.1016/j.eswa.2020.113485

[7] Kosztyán ZT, Szalkai I. Multimode resource-constrained project scheduling in flexible projects. Journal of Global Optimization. 2020;**76**(1):211-241. DOI: 10.1007/s10898-019-00832-8

[8] Kosztyán ZT, Szalkai I. Hybrid time-quality-cost trade-off problems. Operations Research Perspectives. 2018;**5**:306-318. DOI: 10.1016/j.orp.2018.09.003

[9] Kosztyán ZT. MFPP: Matrix-based flexible project planning. SoftwareX. 2022;**17**:100973. DOI: 10.1016/j.softx.2022.100973

[10] Kosztyán ZT, Jakab R, Novák G, Hegedüs C. Survive IT! Survival analysis of IT project planning approaches. Operations Research Perspectives. 2020;**7**:100170. DOI: 10.1016/j.orp.2020.100170

[11] Kuhrmann M, Diebold P, Münch J, Tell P, Trektere K, Caffery FM, et al. Hybrid software development approaches in practice: A European perspective. IEEE Software. 2018;**36**(4):20-31. DOI: 10.1109/MS.2018.110161245

[12] Rahimian V, Ramsin R. Designing an agile methodology for mobile software development: A hybrid method engineering approach. 2008 Second International Conference on Research Challenges in Information Science. June 3-6, 2008, Marrakesh, Monaco: IEEE; 2008. pp. 337-342. DOI: 10.1109/RCIS.2008.4632123

[13] Servranckx T, Coelho J, Vanhoucke M. Various extensions in resource-constrained project scheduling with alternative subgraphs. International Journal of Production Research. 2021:1-20. DOI: 10.1080/00207543.2021.1924411

[14] Steward DV. The design structure system: A method for managing the design of complex systems. IEEE Transactions on Engineering Management. 1981;**3**:71-74. DOI: 10.1109/TEM.1981.6448589

[15] Tyagi MK, Munisamy S, Reddy LSS. Traditional and hybrid software project tracking technique formulation: State space approach with initial state

uncertainty. CSI Transactions on ICT. 2014;**2**(2):141-151. DOI: 10.1007/s40012-014-0037-5

[16] Wysocki RK. Effective Project Management: Traditional, Agile, Extreme. Indianapolis, Indiana, USA: John Wiley & Sons; 2019