A matrix-based flexible multi-level project planning library and indicators

Zsolt T. Kosztyán[1], Gergely L. Novák[2], Róbert Jakab[3], István Szalkai[4], Csaba Hegedűs[5]

[1]University of Pannonia, Department of Quantitative Methods, Veszprém, Hungary
[2]Continental Autmotive Hungary Ltd., Veszprém, Hungary
[3]Continental Teves Autmotive AG&Co., Frankfurt am Main, Germany
[4]University of Pannonia, Department of Mathematics, Veszprém, Hungary
[5]University of Pannonia, Department of Supply Chain, Veszprém, Hungary

Abstract: Flexible approaches such as agile, hybrid, and extreme project management from the software project environment are increasingly used in nonsoftware environments. Nevertheless, only a few methods and no topology-, time- or resource-related indicators or (multilevel) project databases can handle projects of a flexible nature. Therefore, this study shows how to extend existing project databases and indicators to examine and handle such projects. For this study, we merged several heterogeneous project databases into a compound matrix-based project database (CMPD). We compared real-life and simulated project databases, and we studied the effects of flexibility on structural and time- and resource-related indicators.

Keywords: Project scheduling; (Multilevel) project database; Flexibility; Topology; Time- and resource-related indicators

1 Introduction

Projects (of all kinds) can contribute almost 20% of a country's GDP (Denizer et al., 2013; World Bank, 2012). A series of studies show that to increase the success of these projects (SGI, 2019), traditional project management approaches are gradually being replaced by flexible approaches (Ciric et al., 2019; Hidalgo, 2019; Özkan and Mishra, 2019; Wysocki, 2019) not only in the field of IT (Stare, 2014) but also in previously unthinkable fields, such as construction projects (Arefazar et al., 2019) or maintenance (Kosztyán et al., 2019). While flexible projects need flexible project plans, allowing, for example, the possibility of project restructuring and/or task reprioritization according to the customer's requirements, most project planning methods assume a fixed (Franco-Duran and Garza, 2019) logic plan or a limited number of scheduling alternatives (Creemers et al., 2015).

In addition, while there are already a few matrix-based methods for scheduling flexible projects (Kosztyán, 2015) and multilevel projects (Kosztyán, 2020), where some of the task realizations and dependency occurrences are handled as variables during the planning phase, there is neither an existing project database that considers project flexibility nor a set of complexity and time- or resource-related indicators that are capable of characterizing flexible project plans. It is important to provide both scholars and practitioners such a database and set of indicators to allow them to examine flexible projects.

Aims of the paper was threefold.

A1. To specify a matrix-based method, which can handle not only single but multi-level projects, not only single mode, but multi-mode projects, and not only traditional but flexible projects too.
A2. To collect existing heterogeneous project databases, including not only simulated but real-life projects too.
A3. To examine the effect of flexibility not only on the project structure but the project demands, too.

In this paper, 7 single project databases and 12 datasets – from sources including Patterson (Patterson, 1976), Kolish's SMCP and SMFF (Kolisch et al., 1995), PSPLIB (Sprecher and Kolisch, 1996), RG300 and RG30 (Debels and Vanhoucke, 2007; Vanhoucke et al., 2008), Boctor (Boctor, 1993), MMLIB (Peteghem and Vanhoucke, 2014), and a real-life project database (Batselier and Vanhoucke, 2015) – are combined into a matrix-based project library. The paper shows how to extend the databases to handle the flexible nature of the projects. The paper gives flexibility-dependent versions of the complexity and time- and resource-related indicators of individual projects. It also examines the effects of project flexibility. In addition, 5 further multilevel project library is included, such as MPSPLIB (Homberger, 2007), BY (Browning and Yassine, 2010), RCMPSPLIB (Vázquez et al., 2015), MPLIB1/MPLIB2 (Van Eynde and Vanhoucke, 2020).

The contributions to the literature and to practice are summarized below.

1. The unified matrix-based (multilevel) project planning model is proposed to unify a set of heterogeneous single-project databases into a compound matrix-based project database (CMPD).
2. The proposed CMPD is complemented by the ability to model flexible dependencies and completion priorities.

3. With the proposed flexible structure generator (FSG), minimal, minimax, maximin and maximal structures are generated to specify minimal and maximal demands.
4. Structural, time-related and resource-related indicators are modified to handle the flexible nature of projects.

2 Matrix-based model

Apart from network planning methods, matrix-based project planning is used to model complex project plans (Chen et al., 2003). Matrix-based project planning methods are often based on the design (or dependency) structure matrix (DSM) (Steward, 1981). The domain mapping matrix (DMM) is an extended version of the DSM, with multiple domains (Danilovic and Browning, 2007). For single projects, a modified project-oriented version of a domain mapping matrix (DMM) is used, which is called the project domain matrix (PDM) (Kosztyán, 2015). 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.

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 called as *completion modes*.

The first mandatory domain is the logic domain. Diagonal values in $\mathbf{LD}$ 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 $[\mathbf{LD}]_{ij} \in \mathbf{LD} = 1$, 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.
- **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.

The project can be organized into a multilevel project. The projects in the applied UMP (unified matrix-based project-planning model) share their domains. Table 1 shows an example of a multilevel project plan. The common logic domain allows us to plan flexible dependencies both within and between projects. It handles the different completion modes; therefore, all the traditional, hybrid and agile project plans can be planned (see Table 1).

Table 1. Unified matrix-based project-planning model (Only two mandatory domains, such as LD and TD, and two supplementary domains, such as CD and RD are represented in this example).

	(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

If the logic domain of the UMP contains supplementary tasks and/or flexible dependencies, then the minimal (maximal) makespan of the (multilevel) project (henceforth, the total project time [TPT]) can be specified. When supplementary tasks and all supplementary dependencies excluded from (included in), the project (Kosztyán, 2015) (the multilevel project (Kosztyán, 2020)) are called *minimal (maximal) project/multilevel project structures*, denoted as $S_{min}(S_{max})$. In the case of an early schedule, the maximal (minimal) resource use occurs when all supplementary tasks are included in (excluded from) the project while all flexible dependencies are excluded from (included in) the project structure. These structures are henceforth called *maximin (minimax) project structures*, denoted as $S_{maximin}(S_{minimax})$ (see Figure 1.)

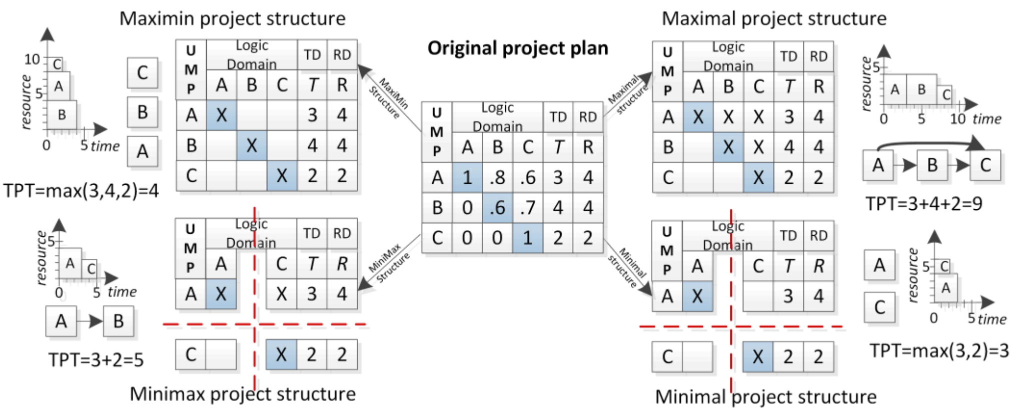


Figure 1. Minimal, maximal, minimax and maximin structures of the flexible project plan.

To indicate that minimal, maximal, minimax and maximin structures are the result of a decision, mandatory tasks and fixed dependencies are represented by X, while omitted tasks and independence are represented by empty cells.

3 Employed indicators

The indicators of project plans can be classified into two groups. The first group of indicators, including I_1 (number of nodes), I_2 (serial or parallel structure), I_3 (task distribution), I_4 (rate of short arcs), I_5 (rate of long arcs), and I_6 (topological float) (Tavares, 1999; Vanhoucke et al., 2008), characterizes the project structure. The other structural operators measure the complexity of the project structures, such as the network complexity (C) indicator of (Sprecher, 1994), the coefficient of network complexity (CNC) of (Davis, 1975), the order strength (OS) of (Mastor, 1970), and the total and average activity density (T-DENSITY and XDENSITY, respectively) of (Patterson, 1976).

The second group of indicators characterizes the project demands. There are time-related indicators, such as the mean and variance of activity durations (XDUR and VA-DUR, respectively), the percent of activities with positive total slack (PCTSLACK), the average total slack per activity (XSLACK), the total and average slack ratios (TOTSLACK-R and

XSLACK-R, respectively), the percent of activities with positive free slack (PCTFREESLK) and the average free slack per activity (XFREESLK) (Patterson, 1976). There are also renewable resource-related indicators, such as the resource factor (RF) (Kolisch et al., 1995) (i.e., the density of the resource domain **RD**), the percent of activities that require the given resource type ($PCTR_j$) (Patterson, 1976), the resource use (RU_i) of the activities (Demeulemeester et al., 2003), the average demand from each resource type, resource constrainedness (RC) (Patterson, 1976), resource strength (RS) (Kolisch et al., 1995), and the following four indicators used by Patterson (1976), which consider the precedence relations of the activities to describe resource needs, the utilization of each type of resource, the constrainedness of the resources, and obstruction and underutilization of the resources. For multilevel projects, in addition to average values, the α-distance (Labro and Vanhoucke, 2008) shows the variation of individual projects' indicators. The Gini coefficient (Van Eynde and Vanhoucke, 2020) measures the inequality of renewable resource demands.

While most of the employed indicators have single and multilevel project versions, none of them considers the structural flexibility of the projects. In addition, there is no published study yet, which analyzes the effect of flexibility for these indicators.

4 Methods

4.1 Parsing heterogeneous project data sources

The different datasets and libraries mentioned in this paper are collected from the project scheduling literature. During our research, we identified suitable data sources that are commonly used and shared by scholars to evaluate their scheduling approaches and find best solutions. The first challenge is usually to access different datasets published by various researchers in the field. One of our intentions was also to review and collect a wide range of available data. The second challenge comes when the data must be handled, as they often have unique formatting and a structure that lacks proper documentation. This might lead to additional reverse engineering efforts that increase the research time and, of course, involve their own risks. Thus, there is a need to harmonize and integrate a wide range of datasets into a library that is accessible and ready to process and that respects the original content.

To overcome limitations such as a lack of standardization and database integration efforts, we decided to write a parser tool (a software that reads inputs, e.g., a text file for further processing) for the most commonly used datasets found in project scheduling research. The parser extracts all information from existing libraries or from the output of project generators in an automated and reproducible way. The resulting data are ready for research and analysis and, if needed, can be further adapted to various formats or platforms. Although our parser covers most of the available formats, the aim is to continuously extend the list of supported extensions. Two main categories of datasets are considered in our study: generated data and empirical data.

Our parser is written in MATLAB and works as follows. It goes through existing project files and looks for network-related data (tasks and their precedence relations); resource- and time-related data, including demands and constraints; and, if present, data on costs and multiple modes of completion. Additional fields are captured from the original data files even if the input is not directly used for scheduling (e.g., the MPM-time field in the case of PSPLIB). The obtained data are then preprocessed into a matrix-based representation and saved to a MAT-file that contains the data as variables. This container file can be easily loaded later into MATLAB's workspace. The parser itself handles renewable resource types, and the tool is designed such that it can be easily extended to use other types (e.g., nonrenewable and doubly constrained resource types). From all the parsed libraries and datasets considered, we selected datasets specifically for this paper. To allow a straightforward comparison of the different indicators, we chose only single-mode examples, and cost-related data are not considered, as they are available in only one library.

4.2 Simulating flexibility

Since none of the project databases considers flexible project structures, in the first step, flexible project structures are generated from fixed structures. According to the specified *flexibility factor* ($ff \in [0,1]$) the rate of mandatory tasks and fixed dependencies made flexible by the flexible structure generator (FSG). Formally, in the ratio *ff*, the values of cells containing 1s decreased from 1 to between 0 to 1. In this way, the rate of supplementary tasks and flexible dependencies can be set up following by the specified ff. After setting flexibility minimal, maximal (which is the original structure in this case), minimax and maximin structures are specified and added to the CMPD databases.

5 Results

Since the limitation of the size of the paper only selected results are showed. First, the results of single projects are represented.

5.1 Effects of flexibility for indicators

Figure 2 shows a comparison of single projects of the structural indicators for the 12 datasets from the 7 databases for 5 different flexibility ratios.

Figure 2. The effects of flexibility for structural indicators.

Figure 2 shows that the considered datasets provide various complexity values. It is important to note that on most complexity measures, such as $I_1 - I_6$, OS, and C, the real-life database covers the greatest intervals of structural and complexity values, while in regards to the CNC, T-DENSITY, and X-DENSITY indicators, the RG300 datasets cover the most possible values. Nevertheless, generally, the flexibility extends the covered intervals of the structural indicators in any of the datasets.

Considering demand-related indicators also inform us that the real-life database and the RG300 dataset cover most of the possible values of time-related measures/indicators. Nevertheless, despite the spread of the time-related value intervals induced by the consideration of flexibility, the real-life database covers significantly more possible values for the time-related indicators. In other words, without considering flexibility, any single simulated database focuses on a narrow interval of time-related indicators that can be very far from the real-life project values. The difference between the simulated and real-life projects on resource-related indicators can also be identified in Figure 3. Nevertheless, in contrast to the time-related indicators, Figure 3 also shows that the MMLIBPLUS dataset provides resource-related indicator values –e.g., the values of resource strength (RS) – that never occur in a real-life project. For example, the number of resources (num_r_resources), resource constrainedness (RC), and underutilization factor (UFACT) values vary in a wider range for the real-life database. In all cases, by introducing flexibility into the project structures and including the generated minimal structures, the interval of the possible values of structural and time-/resource-related indicators can be widened and brought closer to the values of the real-life database.

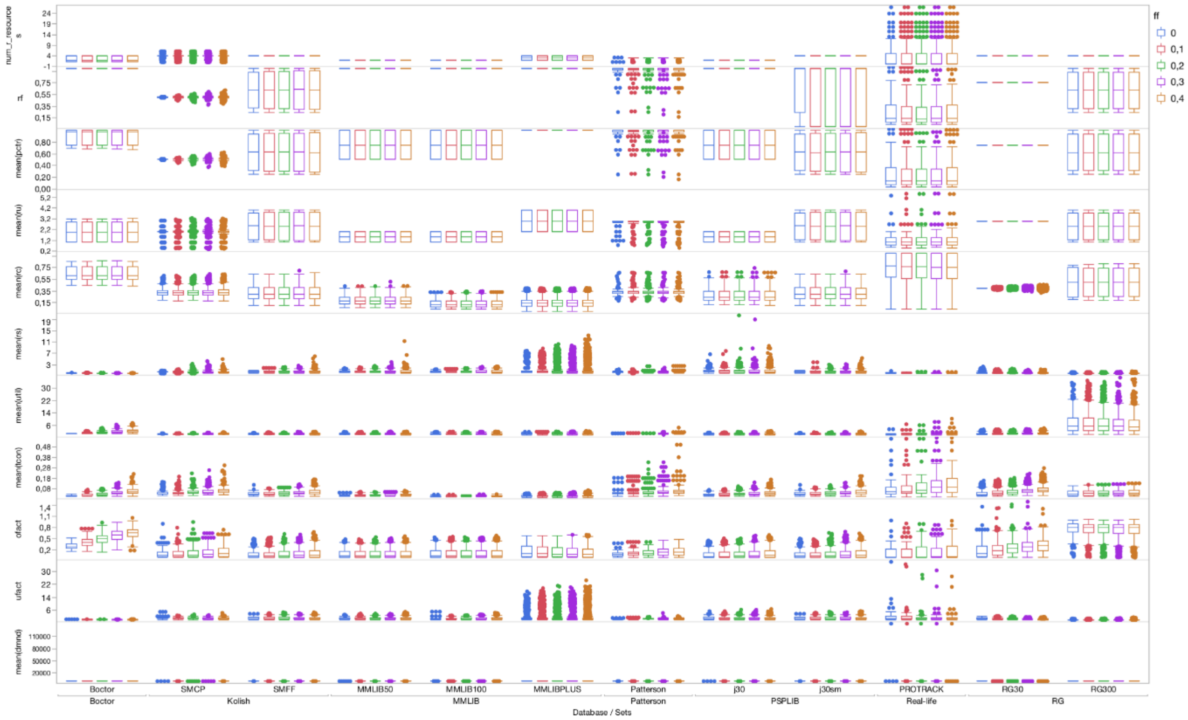
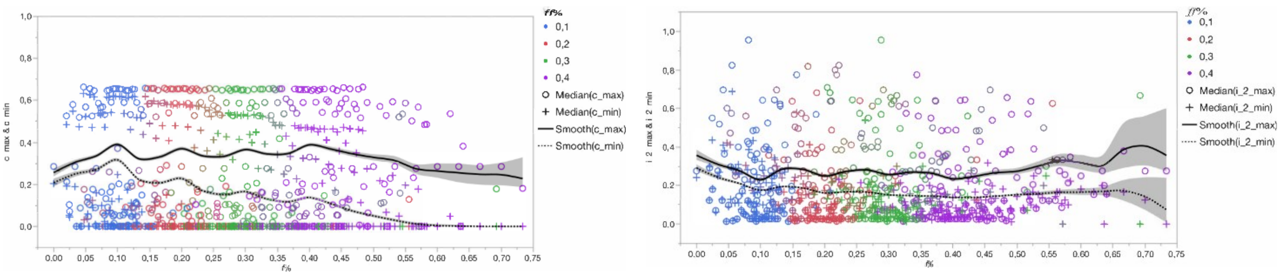


Figure 3. The effects of flexibility for resource-related indicators.

Figure 4 compares the complexity (C) and parallelization (I_2) values of the minimal and maximal structures regarding the ratio of flexible dependencies (ff) (marked on the horizontal axis).



(a) Flexibility vs. complexity.

(b) Flexibility vs. parallelization.

Figure 4. The effects of flexibility for structural-related indicators.

Figure 4 shows that when the flexibility factor is increased, the complexity (C) decreases (see Figure 4(a)), and serial completions also decrease (see Figure 4(b)).

5.2 Flexibility effects for the interdependence of the indicators

The consideration of flexibility expands not only the interval of indicator values but also specifies new value pairs for the coupled indicators.

Figure 5 shows the effect of including minimal structures on the complexity and time-related indicators. In all subfigures, the blue circles and plus signs represent the original pairs of indicator values. Figure 5 shows the pairs of indicator values for the total slack ratio (TOTALSLACK-R) and average slack ratio (XSLACK-R) as time-related indicators on the vertical axis and complexity (C) and parallelization (I_2) as structural parameters on the horizontal axis.

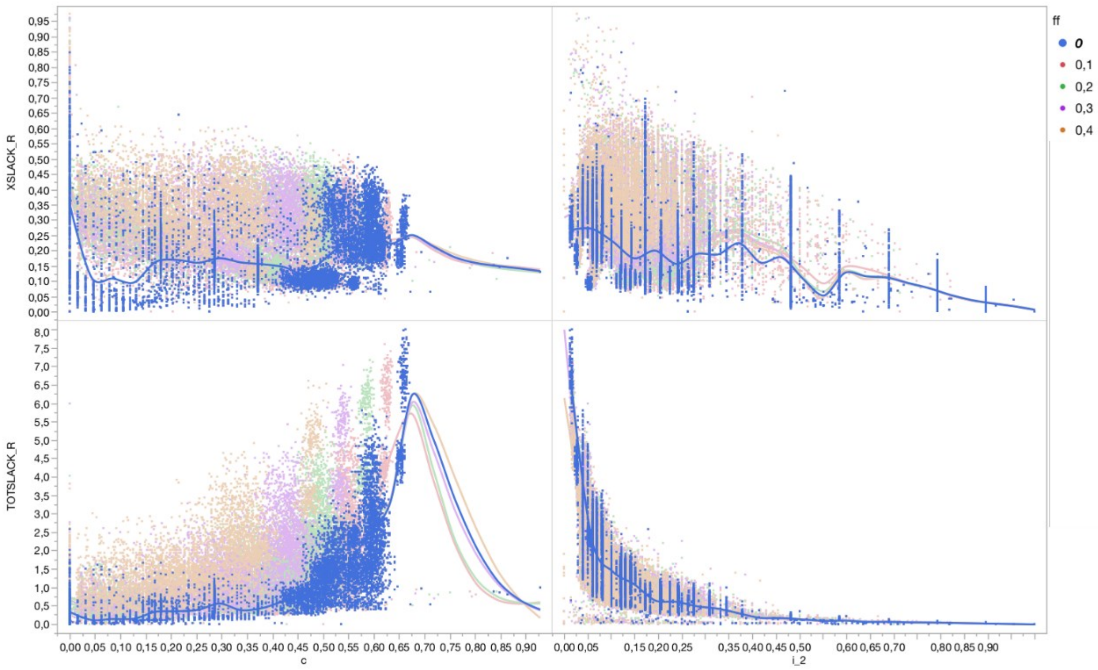


Figure 5. The effect of flexibility for relations between time-related and complexity indicators.

Figure 5 shows that including minimal structures helps to explore new areas on the planes spanned by structural–time-related indicator pairs. These specified combinations better cover the area of possible value pairs. Flexibility can also be expressed here in other ways: the minimal structures of flexible projects have higher average slacks, which can be better utilized in resource allocation.

Figure 6 shows the relations between the slack ratios (TOTSLACK-R, XSLACK-R) and resource-related indicators for the earliest start schedule.

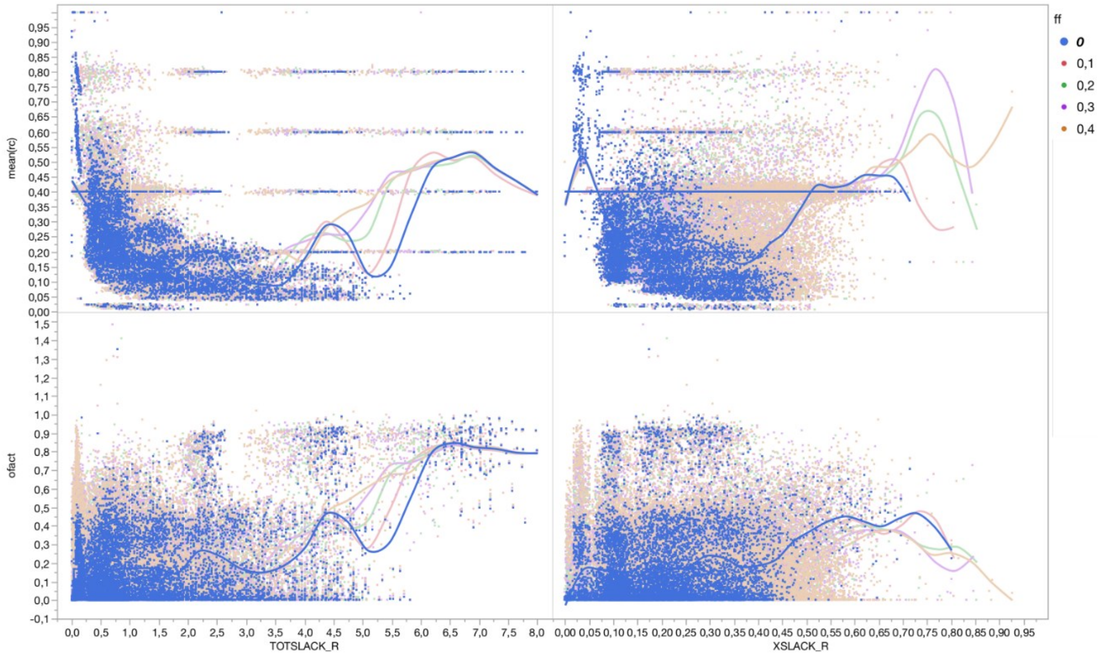


Figure 6. The effect of flexibility for relations between time- and resource-related and indicators.

Figure 6 shows that considering the minimal structures of flexible projects increases the slack ratio and, because of the parallelization, also the resource constrainedness and the obstruction factor. These combinations of resource- and time-related indicator values occur only for flexible project plans.

5.3 Flexibility effects for multilevel projects

In the context of multilevel projects, Figure 7 shows that by introducing minimal structures, the flexible projects become more parallel and slack times are increasing while their overall complexity is reduced. As a result, total project time is also reduced, and resources get more constrained. The interval of indicator values for parallelity and complexity shrinks and shifts to lower values considering minimal structures.

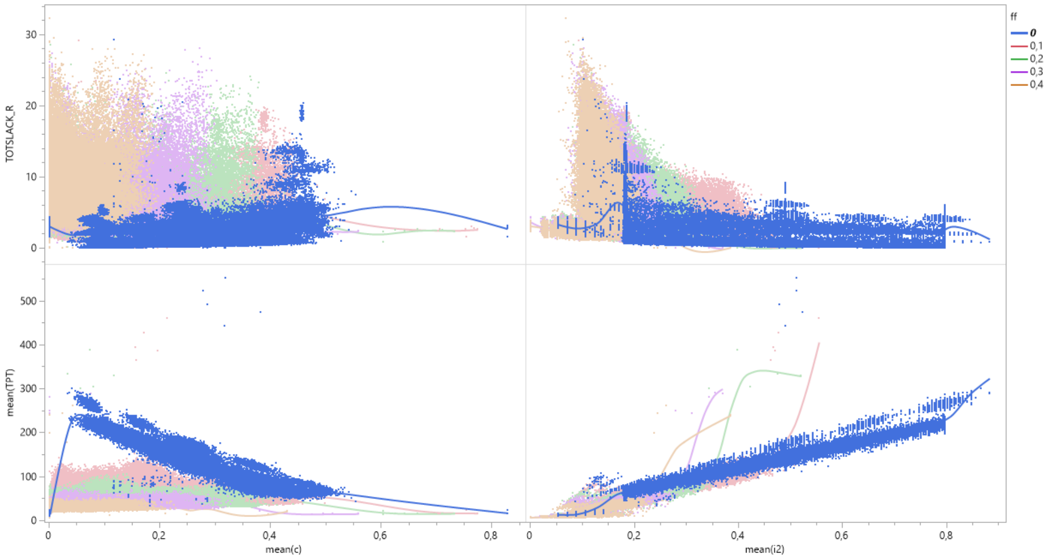


Figure 7. The effect of flexibility for relations between time-related and complexity indicators.

Figure 8 depicts structural and resource-related indicators and gives insight into their variation on multiple levels: tasks, connected components, and projects. Variation in parallelity described by α(I2) gets closer to one (less variation) when flexibility is present and varies more when only a few projects exist.

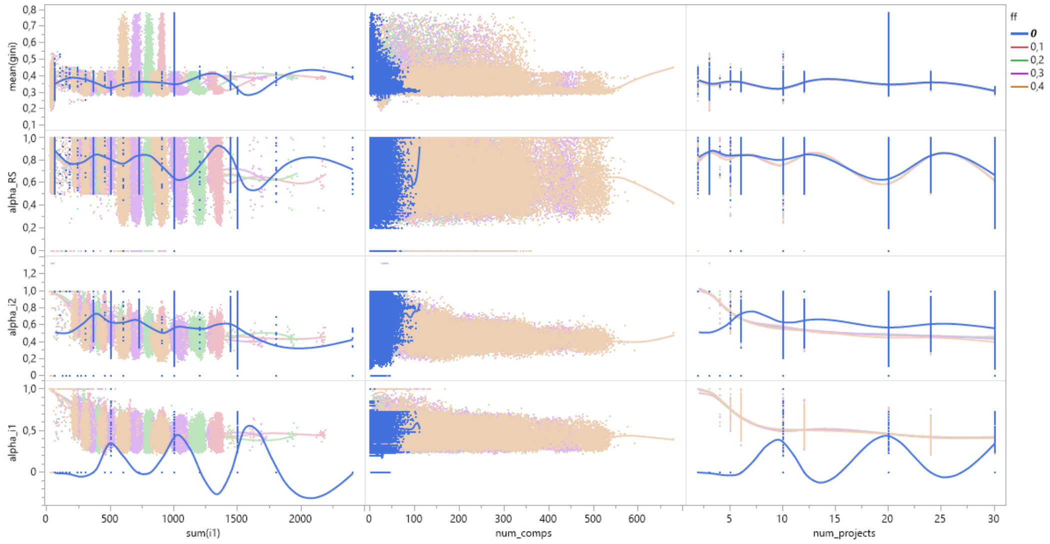


Figure 8. The effect of flexibility for distributions of structural and resource-related indicators on different levels.

With increasing flexibility for minimal structures, the values of the Gini index become smaller as the work demand for resources gets more equally distributed (closer to zero) amongst projects. However, the interval of Gini indices also widens, which means a higher potential inequality in some cases. α(RS) shows less variation (closer to zero) in the resource demand and availability relation when flexibility is higher. α(I1) shows that the variation in the number of tasks is decreasing with more projects present.

6 Discussion

When testing project scheduling and resource allocation algorithms only on simulation databases, two types of errors can be made. The first problem is whether new algorithms are applied on real-life projects that have different kinds of complexity-, time- or resource-related indicator values than the simulated projects in the (benchmark) databases. Even if scheduling the simulated projects is harder for the current objectives and algorithms, these algorithms may be not prepared for the challenges of new objectives for real-life projects. Creating specified database tailored to one kind of problem can cause discrepancies in the real-life usage, because of indirect constraints routed in not considered properties. Secondly, if algorithms are optimized to properties of simulated projects that never appear in a real-life case, that is misspent of resources. Simulation datasets should also be combined because an individual dataset usually covers only a small interval of an indicator. Results also show that including minimal structures widens the intervals of indicators; therefore, even if flexible structures are not studied, the extended dataset may cover larger intervals of indicators.

By considering flexibility and generating minimal structures, the interval of indicators can be widened; therefore, this operation should be covered in the testing of project scheduling or a resource allocation algorithm to widen the scope of the application of that algorithm. Nevertheless, considering minimal structures does not solve the problem that most complexity-, time- and resource-related measures are still significantly different between the real-life and simulated databases. Results show that the increase in flexibility reduces the complexity and increases the parallelization (decreases the task sequence length). These results are in line with the requirements of flexible project management approaches for reducing project complexity (Williams, 2010).

7 Summary and conclusion

This study stated three aims (A1-A3). In this study, a unified matrix-based project-planning model (UMP) is proposed (A1) to model heterogeneous project plans (A2). To combine heterogeneous project databases, a compound matrix-based project database (CMPD) is proposed (A2). In addition, a flexible structure generator (FSG) is proposed to extend the existing project databases to handle possible structures of flexible project plans (A3). The proposed minimal and maximal structures specify new combinations of the structural and demand indicator values to test algorithms in flexible project management environments.

The UMP handles both individual and multiple projects, single and multimodal completions, renewable and nonrenewable resources, cost and quality parameters and traditional and flexible project plans (A1). The unified database contains both simulated and real-life data sources (A2). The proposed parsers are prepared for single and multimode completion modes. Therefore, the proposed CMPD provides a wider range of test project schedules and resource allocation algorithms (A3).

Acknowledgement

This work has been implemented by the TKP2021-NVA-10 project with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the 2021 Thematic Excellence Programme funding scheme.

References

Arefazar, Y., Nazari, A., Hafezi, M.R., Maghool, S.A.H., 2019. Prioritizing agile project management strategies as a change management tool in construction projects. Int. J. Constr. Manag. 0, 1–12. https://doi.org/10.1080/15623599.2019.1644757

Batselier, J., Vanhoucke, M., 2015. Construction and evaluation framework for a real-life project database. Int. J. Proj. Manag. 33, 697–710.

Boctor, F.F., 1993. Heuristics for scheduling projects with resource restrictions and several resource-duration modes. Int. J. Prod. Res. 31, 2547–2558.

Browning, T.R., Yassine, A.A., 2010. A random generator of resource-constrained multi-project network problems. J. Sched. 13, 143–161.

Chen, C.-H., Ling, S.F., Chen, W., 2003. Project scheduling for collaborative product development using {DSM}. Int. J. Proj. Manag. 21, 291–299. http://dx.doi.org/10.1016/S0263-7863(02)00023-6

Ciric, D., Lalic, B., Gracanin, D., Tasic, N., Delic, M., Medic, N., 2019. Agile vs. Traditional Approach in Project Management: Strategies, Challenges and Reasons to Introduce Agile. Procedia Manuf. 39, 1407–1414. https://doi.org/10.1016/j.promfg.2020.01.314

Creemers, S., Reyck, B.D., Leus, R., 2015. Project planning with alternative technologies in uncertain environments. Eur. J. Oper. Res. 242, 465–476. http://dx.doi.org/10.1016/j.ejor.2014.11.014

Danilovic, M., Browning, T.R., 2007. Managing complex product development projects with design structure matrices and domain mapping matrices. Int. J. Proj. Manag. 25, 300–314. http://dx.doi.org/10.1016/j.ijproman.2006.11.003

Davis, E.W., 1975. Project network summary measures constrained-resource scheduling. AIIE Trans. 7, 132–142.

Debels, D., Vanhoucke, M., 2007. A decomposition-based genetic algorithm for the resource-constrained project-scheduling problem. Oper. Res. 55, 457–469.

Demeulemeester, E., Vanhoucke, M., Herroelen, W., 2003. RanGen: A random network generator for activity-on-the-node networks. J. Sched. 6, 17–38.
Denizer, C., Kaufmann, D., Kraay, A., 2013. Good countries or good projects? Macro and micro correlates of World Bank project performance. J. Dev. Econ. 105, 288–302. https://doi.org/10.1016/j.jdeveco.2013.06.003
Franco-Duran, D.M., Garza, J.M. de la, 2019. Review of Resource-Constrained Scheduling Algorithms. J. Constr. Eng. Manag. 145, 03119006.
Hidalgo, E.S., 2019. Adapting the scrum framework for agile project management in science: case study of a distributed research initiative. Heliyon 5, e01447. https://doi.org/10.1016/j.heliyon.2019.e01447
Homberger, J., 2007. A multi-agent system for the decentralized resource-constrained multi-project scheduling problem. Int. Trans. Oper. Res. 14, 565–589.
Kolisch, R., Sprecher, A., Drexl, A., 1995. Characterization and generation of a general class of resource-constrained project scheduling problems. Manag. Sci. 41, 1693–1703.
Kosztyán, Z.T., 2020. An Exact Algorithm for the Flexible Multilevel Project Scheduling Problem. Expert Syst. Appl. 113485.
Kosztyán, Z.T., 2015. Exact algorithm for matrix-based project planning problems. Expert Syst. Appl. 42, 4460–4473. http://dx.doi.org/10.1016/j.eswa.2015.01.066
Kosztyán, Z.T., Pribojszki-Németh, A., Szalkai, I., 2019. Hybrid multimode resource-constrained maintenance project scheduling problem. Oper. Res. Perspect. 6, 100129.
Labro, E. and Vanhoucke, M., 2008. Diversity in resource consumption patterns and robustness of costing systems to errors. Management Science, 54(10), pp.1715-1730.
Mastor, A.A., 1970. An experimental investigation and comparative evaluation of production line balancing techniques. Manag. Sci. 16, 728–746.
Özkan, D., Mishra, A., 2019. Agile Project Management Tools: A Brief Comprative View. Cybern. Inf. Technol. 19, 17–25.
Patterson, J.H., 1976. Project scheduling: The effects of problem structure on heuristic performance. Nav. Res. Logist. Q. 23, 95–123.
Peteghem, V.V., Vanhoucke, M., 2014. An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances. Eur. J. Oper. Res. 235, 62–72. https://doi.org/10.1016/j.ejor.2013.10.012
SGI, 2019. CHAOS Manifesto.
Sprecher, A., 1994. Generation of Instances by ProGen, in: Resource-Constrained Project Scheduling. Springer, pp. 70–90.
Sprecher, A., Kolisch, R., 1996. PSPLIB—a project scheduling problem library. Eur J Oper Res 96, 205–216.
Stare, A., 2014. Agile Project Management in Product Development Projects. Procedia - Soc. Behav. Sci. 119, 295–304. https://doi.org/10.1016/j.sbspro.2014.03.034
Steward, D.V., 1981. The design structure system- A method for managing the design of complex systems. IEEE Trans. Eng. Manag. 28, 71–74.
Tavares, L.V., 1999. Advanced models for project management. Springer Science & Business Media.
Van Eynde, R., Vanhoucke, M., 2020. Resource-constrained multi-project scheduling: benchmark datasets and decoupled scheduling. J. Sched. 23, 301–325.
Van Eynde, R. and Vanhoucke, M., 2022. New summary measures and datasets for the multi-project scheduling problem. European Journal of Operational Research, 299(3), pp.853-868.
Vanhoucke, M., Coelho, J., Debels, D., Maenhout, B., Tavares, L.V., 2008. An evaluation of the adequacy of project network generators with systematically sampled networks. Eur. J. Oper. Res. 187, 511–524.
Vázquez, E.P., Calvo, M.P., Ordóñez, P.M., 2015. Learning process on priority rules to solve the RCMPSP. J. Intell. Manuf. 26, 123–138.
Williams, L., 2010. Agile Software Development Methodologies and Practices, in: Zelkowitz, M.V. (Ed.), Advances in Computers, Advances in Computers. Elsevier, pp. 1–44. https://doi.org/10.1016/S0065-2458(10)80001-4
World Bank, D.R.G., 2012. The little data book on financial inclusion 2012. World Bank Publications.
Wysocki, R.K., 2019. Effective project management: traditional, agile, extreme, hybrid. John Wiley & Sons.

Contact: Zsolt T. Kosztyán, University of Pannonia, Department of Quantitative Methods, Egyetem str.10, H-8200, Veszprm, Hungary, +36202085840, kzst@gtk.uni-pannon.hu, https://www.gtk.uni-pannon.hu/zsoltkosztyan/