A matrix-based flexible project-planning library and indicators

Zsolt T. Kosztyán[a], Gergely Novák[b], Róbert Jakab[c], István Szalkai[d], Csaba Hegedűs[e]

[a] *Professor, Department of Quantitative Methods, University of Pannonia, Egyetem str. 10, Veszprém, H-8200, email: kzst@gtk.uni-pannon.hu*

[b] *Lead Software Project Manager, Continental Automotive Hungary Ltd., Házgyári str. 6-8., Veszprém, H-8200, Hungary, email: gergely.novak@continental-corporation.com*

[c] *Department Leader, Customer Lead Engineer, Continental Teves AG & Co. oHG, Guerickestraße 7, Frankfurt am Main, 60488, Germany, email: robert.jakab@continental-corporation.com*

[d] *Associate Professor, Department of Mathematics, University of Pannonia, Egyetem str. 10, Veszprém, H-8200, email: szalkai.istvan@mik.uni-pannon.hu*

[e] *Associate Professor, Department of Supply Chain Management, University of Pannonia, Egyetem str. 10, Veszprém, H-8200, Hungary, email: hegedus.csaba@gtk.uni-pannon.hu*

Abstract

Flexible approaches, such as agile, hybrid, and extreme project management methods in the software project environment, are increasingly being used in nonsoftware environments. Nevertheless, only a few methods and no topological, time-related or resource-related indicators or project databases can address projects of a flexible nature. This study proposes a unified matrix-based model (UMP) to consider single-mode, multimode, individual and multiple projects. Based on the UMP, a CMPD is specified to collect 11 existing project databases. In addition, a flexible structure generator (FSG) is proposed to generate flexible tasks and dependencies for analyzing the effects of flexibility on topology and project demand-related indicators. By correlation graphs, the relations between structure-related and demand-related indicators are analyzed. The comparison of the simulated and real-life databases shows that the interpretation range of the indicators is different; however, considering the flexibility, this problem can be resolved.

Keywords: Project scheduling; Project database; Flexibility; Topology; Time- and resource-related indicators

1. Introduction

Projects (of all types) can contribute almost 20% of a country's GDP (Denizer et al., 2013; World Bank, 2012). Several studies have shown that to increase the success of these projects (SGI, 2015), traditional project management approaches are gradually being replaced by flexible approaches (Wysocki, 2019; Ciric et al., 2019; Özkan and Mishra, 2019; Hidalgo, 2019) in not only the IT field (see, e.g., in Stare, 2014) but also previously unconsidered fields, such as construction (Yasaman et al., 2022) and maintenance projects (Kosztyán et al., 2019).

Broadly defined, flexibility is the magnitude of the room for scheduling decisions. (Multiple) project scheduling is open to several flexibility types; time-related or scheduling flexibility can result from slacks or topological floats (see Tavares (1999); Vanhoucke et al. (2008)) in the project plan. This type is the most obvious, and it frequently occurs even in traditional projects. In this case, the precedence relations and the implementation modes remain the same, and only the scheduled start and finish times of the tasks change. Hauder et al. (2020) shows how this flexibility can change the logistical (storing or conveying) task duration, but it can be implemented by defining the minimal and maximal time lags of an activity-on-node project network (Ren et al., 2021).

The second type is activity (i.e., task) or modal flexibility in which a task can be performed in several modes or the same result can be achieved by carrying out one of different sets of tasks and utilizing different resource combinations. These alternative (sets of) tasks are modeled by Petri nets in Čapek et al. (2012), by mandatory and optional choices in the project network (Kellenbrink and Helber, 2015), or by the AND/OR network in Tao and Dong (2018). These works extended the resource constrained (multiple) project scheduling problems (RCPSP or RCMPSP) with alternative activity chains (RCPSP-AC or RCMPSP-AC). Combined with time-related flexibility, Hauder et al. (2020) defined the problem set of the resource-constrained multiple project-scheduling problem with alternative activity chains and time-related flexibility (RCMPSP-ACTF).

The third type is dependency flexibility. If the project task technology does not require a strict sequence, some logical dependencies can be omitted. Omitting a dependency lifts the restriction of sequential execution and allows the associated tasks to be performed in parallel or in an arbitrary, relative order.

The fourth type is scope flexibility in which some low priority tasks can be omitted or postponed to a later project. This situation reduces the resource demand and can shorten the project duration by sacrificing quality or fulfilment level. The latter two flexibility types appear typically but not exclusively in agile projects (Kosztyán, 2015). Since these flexibilities affect the logical structure of a project, i.e., which tasks are performed and according to which logical dependency are they performed, hereinafter, dependency and scope flexibility are together called structural flexibility.

While structurally flexible projects require flexible project plans, allowing 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; Servranckx and Vanhoucke, 2019; Čapek et al., 2012; Kellenbrink and Helber, 2015; Tao and Dong, 2018; Hauder et al., 2020). In addition, a few matrix-based methods are available for scheduling structurally flexible projects (Kosztyán, 2015; Kosztyán and Szalkai, 2020); among these, some task realizations and dependency occurrences are treated as variables during the planning phase.

In flexible, such as in agile and extreme, project management, the entire project is usually split into smaller parts, called iterations or sprints in the SCRUM, KANBAN, or SCRUMBAN methodology (Wysocki, 2019). These iterations usually take 2-6 weeks. However, every iteration should be closed with a business value, called *minimal viable production* (MVP). . Therefore, an iteration can be considered a mini-project. Based on the customer's requirements tasks have to be prioritized. In addition, most tasks can be parallelized (see Fig. 1).

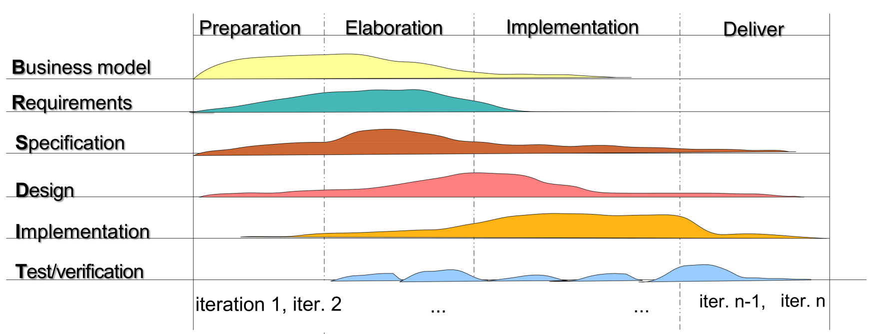


Figure 1: Splitting a waterfall software development project into iterations by an agile project management approach

Fig. 1 shows an example of splitting a waterfall software development project into smaller iterations. The colorful distributions indicate that every iteration contains tasks of most phases of the software development project. It is important to note that without reconsidering precedences between phases and functions of the development, parallelization cannot be completed. It is also important to note that even though agile approaches came from the field of software development, nowadays, primarily not only software development is managed with agile techniques (Wysocki, 2019; Yasaman et al., 2022). Therefore, we believe that studying flexible projects will be further emphasized. In this context, it is surprising that no databases are currently available to help design and schedule (structurally) flexible projects. We believe that our study fills this gap.

Within an iteration, agile project management approaches do not allow new, so-called unplanned tasks. They have to be planned into the next iterations, while extreme project management approaches allow new tasks, which also have to be scheduled within the iterations. (Wysocki, 2019). Kosztyán and Szalkai (2020) showed that in terms of scheduling flexible tasks and dependencies, there is no difference in flexible, such as agile and extreme project scheduling approaches, which means the same scheduling algorithm can be used for planned and unplanned tasks. Nevertheless, the rate of flexible tasks and dependencies primarily influences the scheduling performances.

The contributions to the literature and practice are summarized below.

1. A unified matrix-based project-planning model (UMP) 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. Minimal, minimax, maximin and maximal structures are generated to specify the minimal and maximal demands with the proposed flexible structure generator (FSG).
4. Structure-related, time-related and resource-related indicators are modified to address the flexible nature of projects.

In this paper, 10 project databases, 22 datasets, and the real-life project database reported by Batselier and Vanhoucke (2015) were combined into a matrix-based project library. This paper provides a way to extend the databases to address the flexible nature of projects. This paper gives flexibility-dependent versions of the complexity and the time-related and resource-related indicators of individual projects. It also examines the effects of project flexibility.

The remainder of the paper is organized as follows. In the background section (Section 2), related works and databases are reviewed. In Section 3, first, the applied project databases and the considered complexity, time-related and resource-related indicators are introduced. Then, the flexibility-dependent indicators are specified. In Section 4, the applied project databases are compared, and the flexibility effects are examined. Section 5 discusses the results. Finally, in Section 6, we provide a summary, the limitations of this study and directions for future work.

2. Background

Project databases play a key role in rendering different scheduling and resource allocation methods (Brucker et al., 1999; Hartmann and Briskorn, 2010) comparable and developing new methods (Franco-Duran and Garza, 2019). Individual projects are available in various databases, such as Patterson (Patterson, 1976), 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 the real-life project database by (Batselier and Vanhoucke, 2015), or sets of individual or multiple projects, including MPSPLIB (Homberger, 2007), BY (Browning and Yassine, 2010), RCMPSPLIB (Vázquez et al., 2015), and MPLIB (Van Eynde and Vanhoucke, 2020). All these databases contain tasks and dependencies between tasks and renewable resources. However, most databases do not include costs, quality or nonrenewable resources, and none of them account for flexibility issues. In addition, the source file format is heterogeneous; therefore, if a scholar wants to test a new method in multiple databases, different parsers must first be written for each project database. Heterogeneity is not simply a matter of format; tasks may be assigned different requirements, such as duration, cost, or renewable and nonrenewable resource demands. In addition, several databases contain only one completion mode (namely, those of Patterson (Patterson, 1976), SMCP and SMFF (Kolisch et al., 1995), PSPLIB (Sprecher and Kolisch, 1996), RG300 and RG30 (Debels and Vanhoucke, 2007; Vanhoucke et al., 2008), and the real-life database (Batselier and Vanhoucke, 2015)), while others contain multiple completion modes (namely, PSPLIB (Sprecher and Kolisch, 1996), Boctor (Boctor, 1993), and MMLIB (Peteghem and Vanhoucke, 2014)). Table 1 summarizes the properties of the existing project databases.

Table 1: Applied project databases

Name	Project Plan	Completion Modes	Projects	Demands	Cited as
Patterson	Generated	Single	Single	Time, renewable resources	Patterson (1976)
PSPLIB	Generated	Single, Multiple	Single	Time, re/nonrenewable resources	Sprecher and Kolisch (1996)
RG30, RG300	Generated	Single	Single	Time, renewable resources	Vanhoucke et al. (2008)
SMCP, SMFF	Generated	Single	Single	Time, renewable resources	Kolisch et al. (1995)
Boctor	Generated	Multiple	Single	Time, renewable resources	Boctor (1993)
MMLIB	Generated	Multiple	Single	Time, re/nonrenewable resources	Peteghem and Vanhoucke (2014)
Real-life	Collected	Single	Single	Time, cost, renewable resources	Batselier and Vanhoucke (2015)
MPSPLIB	Generated	Single	Multiple	Time, renewable resources	Homberger (2007)
BY	Generated	Single	Multiple	Time, cost, renewable resources	Browning and Yassine (2010)
RCMPSPLIB	Generated	Single	Multiple	Time, renewable resources	Vázquez et al. (2015)
MPLIB1, MPLIB2	Generated	Single	Multiple	Time, renewable resources	Van Eynde and Vanhoucke (2020)

To the best of our knowledge, no study comprehensively assessed how well the simulated databases describe the plurality of real projects. Some criticism has arisen regarding these simulated project databases. Peteghem and Vanhoucke (2014) reported 4 shortcomings of the widely used PSPLIB. One limitation is the low diversity in the complexity of topology networks indicated by the order strength (OS) values. The authors also found that Boctor's dataset contains mainly serial projects, and the renewable resources are hardly restricted by the constraints. In the results, the instances of the real project dataset (Protrack) are compared to the simulated ones, and the effects are evaluated by introducing flexibility to the implementation priority or precedence relations on the project properties.

The following datasets were not applied in our study as only a part of their data could be used. However, they can still be useful for further research. The MT dataset (Vanhoucke, 2010b) is mainly used for schedule risk analysis and earned value management, contains project structures that can be combined with ResSet, have additional resource data, and result in the NetRes dataset (Vanhoucke and Coelho, 2018). DC1 (Vanhoucke et al., 2001) and DC2 (Vanhoucke, 2010a) are studied within the context of the RCPSP with discounted cash flows. The CV set (Coelho and Vanhoucke, 2020) contains RCPSP instances that are difficult to solve. MISTA2013 (Wauters et al., 2016) is a dataset and generator for the multimode resource-constrained multiple project scheduling problem (MRCMPSP) and combines instances from PSPLIB.

Other sources of projects are project generators, such as ProGen (Kolisch et al., 1995), Progen/max (Schwindt and Schwindt, 1995) and Progen/πx (Drexl et al., 2000), RanGen1 and RanGen2 (Demeulemeester et al., 2003; Vanhoucke et al., 2008), RiskNet (Tavares, 1999), and the random generator by Browning and Yassine (2010). These project generators have been used to generate several project databases, such as the PSPLIB (Sprecher and Kolisch, 1996), the RG300 and RG30 (Debels and Vanhoucke, 2007; Vanhoucke et al., 2008), the MMLIB (Peteghem and Vanhoucke, 2014), and BY (Browning and Yassine, 2010). When generating a new project, only very few structure-related, time-related and resource-related indicators can be set. Therefore, numerous undiscovered and untested project structures cannot be generated by the existing project generators. Although this study does not aim to develop a new project generator, our consideration of flexibility extends the domain of the indicator values.

To unify the heterogeneous project databases, in this study, we propose a matrix-based model that can accommodate both individual and multiple projects, both single- and multimodal completions and both renewable and nonrenewable resources. Although the proposed unified matrix-based model can consider different types of projects, to ensure the comparison between the existing real-life and simulated databases, which consist mainly of single projects, only single modes are considered.

Projects are usually represented as graphs in which activities (i.e., tasks) are in depicted either arcs (activity-on-arrow [AoA] networks) (Demeulemeester et al., 1996) or nodes (activity-on-node [AoN] networks) (Ren et al., 2021). The matrix representation of projects usually describes an AoN network (Minogue, 2011). Kosztyán (2015) suggested a project domain matrix (PDM) that can be used for both single and multimodal project plans. PDMs allow mandatory and supplementary tasks with priorities and flexible dependencies between tasks. Kosztyán (2020) subsequently extended this matrix-based model to address multiple projects, programs and project portfolios. This matrix-based multiple project management model is denoted as M^4. In this study, we unified and extended M^4 and PDM to consider nonrenewable resources. The proposed matrix-based method is called the *unified matrix-based project-planning model* (UMP).

The UMP contains two mandatory and four supplementary domains (see Figure 2).

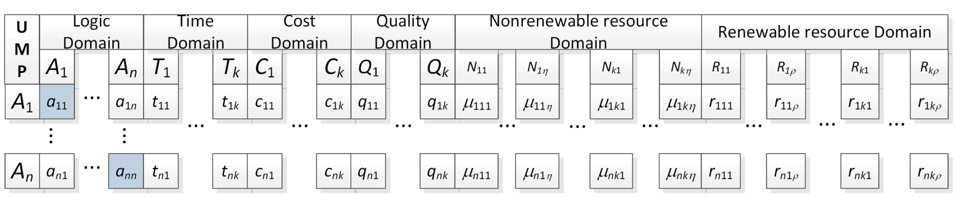


Figure 2: Structure of the unified matrix-based project-planning model (UMP)

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 k matrix with positive real values, where

k is the number of completion modes.

The first mandatory domain is the logic domain, $\mathbf{LD} \in [0,1]^{n\times n}$. The diagonal values in **LD** represent the task priority values. If a diagonal value is 0, the task will not be completed, and if the diagonal value is 1, the task is mandatory. If the diagonal value is between 0 and 1, the task is supplementary, indicating that depending on the decision, it will be either completed or omitted/postponed. In the case of flexible projects, tasks are prioritized by the product owner according to their business value and the risks involved in their development (Abad et al., 2010). To help decision-makers prioritize task completion, several methods, such as MoSCoW rules, are available, and the requirements are prioritized based on their importance by sorting them into the four groups of must-have, should-have, could-have, and will-not-have features. In addition to the categories, tasks can be ranked by their importance, or the importance/priority values can be calculated by the Analytic Hierarchy Process (AHP) method (Srivastava et al., 2021). The prioritization of task completions is the essential part of all flexible, such as agile, hybrid, and extreme project management methods. Nevertheless, @inproceedingsgovil2021information, in this study, only the rate of the existing supplementary (i.e. lower priority) tasks was analyzed; therefore, priority rankings were not studied. Fig. 3 shows an example of MoSCoW prioritization of requirements applied by an agile, namely, the Dynamic System Development Method (DSDM). DSDM method was one of the first, which suggests MoSCoW method prioritize task completion (Stapleton, 1997). This technique indicates that the rate of the mandatory tasks should be approximately 60%. Nevertheless, the concept of task prioritization is generally applied in most agile techniques (Dingsøyr et al., 2012; Govil and Sharma, 2021).

A task can fulfill more than one requirement (see T13); however, usually, to fulfill requirements, more than one task should be completed. In an agile project, only 'MUST (called Maximum Usable SubseT) have' tasks (appr. 60% of tasks and efforts) will be completed necessarily; the other tasks (appr. 40%) are supplementary tasks with a different class of priorities.

The out-diagonal values represent the dependencies between the tasks. If an out-diagonal value $a_{ij} = l_{ij} = [\mathbf{LD}]_{ij}$ $(i \neq j)$ is 1, task i precedes task j. In the case of $l_{ij} = 0$, no precedence relation exists from task i to task j. If $0 < l_{ij} < 1$, a flexible dependency exists between task i and task j, indicating that task i may precede or follow task j depending on the manager's (algorithm's) decisions. All flexible techniques, such as agile,

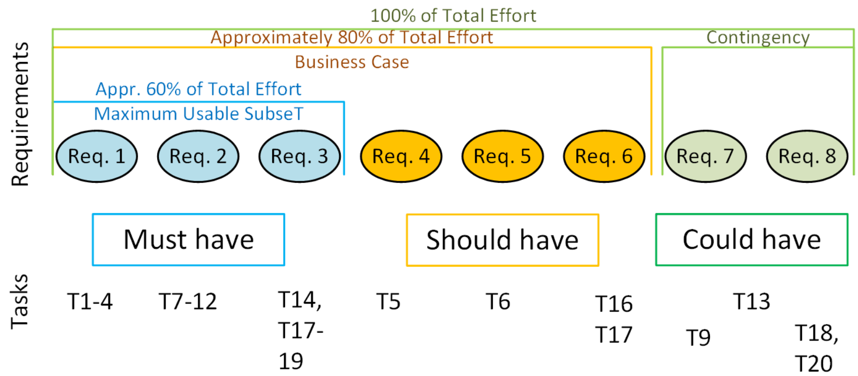


Figure 3: An example of MoSCoW of the prioritization of requirements and tasks (based on the guide of DSDM (Stapleton, 1997))

hybrid, or extreme techniques, require flexible dependencies between tasks (Fernandez and Fernandez, 2008; Ciriello et al., 2022).

Since none of the project networks from the considered databases contains any cycles, they can be ordered topologically, and the logic domain of the topologically ordered project networks is an upper triangular matrix (formally, $l_{ij} = 0$ if $i > j$). Although the matrix-based representation does not require acyclic structures, and feedback can be resolved (see, e.g., in Kosztyán, 2015) since most indicators are defined for acyclic project structures, the upper triangular logic domain is considered for the topologically ordered tasks in the rest of this study. Flexible project management allows iterations; however, the databases lack cycles; thus, we can investigate only one iteration at a time. The Fig. 4 shows how to schedule prioritized tasks using a SCRUMBAN method. SCRUMBAN is a combination of SCRUM, which is the first agile method suggesting sprints (=iterations) (Hidalgo, 2019), and the KANBAN, which limits parallel work-in-progress (WIP) tasks (Williams, 2010). These two techniques, such as SCRUM and KANBAN is most widely used in agile project management (Wysocki, 2019).

The other mandatory UMP domain is the time-related domain. The positive values of the time domain represent the possible task durations. For each task, k types of durations can be assigned; the duration values may also match each other.

Matrix-based methods can also address general precedence relations (GPRs) (Minogue, 2011); however, most databases allow only finish-to-start (F-S) relations between tasks. F-S relations indicate that a successor task

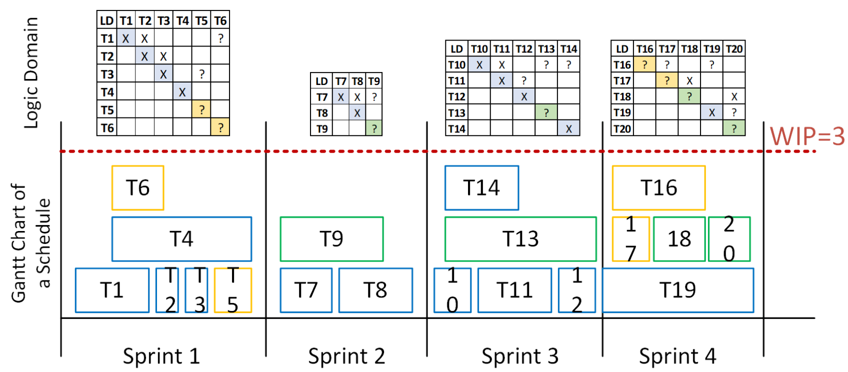


Figure 4: Example of a schedule of prioritized tasks with the SCRUMBAN method ('X'=1 represents mandatory (MUST HAVE) tasks in diagonal, fixed dependencies in out-diagonal; ~~0¡'?'¡1~~ 0 <'?'< 1 represents supplementary (either SHOULD HAVE or COULD HAVE) tasks in diagonal, or flexible dependencies in out-diagonal.

can be started only if all predecessor tasks have been finished. In this study, we assume that tasks can only have F-S relations.

The additional supplementary domains are as follows:

CD The cost domain is an n by k nonnegative matrix of the task costs

QD The quality domain is an n by k, nonnegative matrix of the task quality parameters, where the quality parameters are between [0,1]

ND The nonrenewable resource domain is an n by $k \cdot \eta$ 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 $k \cdot \rho$ nonnegative matrix of renewable resource demands, where ρ is the number of types of renewable resources

The optional domains can be either ignored or filled in with zero values. In this study, we always used the **LD**, **TD** and **CD** domains, and if there are renewable resources, the **RD** was also filled in, but if there is no information regarding resources, the **RD** was ignored. The applied database does not contain quality data; therefore, **QD** was omitted. The study focuses only on the structure, time-related and (renewable) resource demands; therefore, a nonrenewable domain was not used. Since the real-life database counts of the task and resource costs can also be calculated from the multiplication of resource and time demands, **CD** was not ignored. However, in this study, the costs were not analyzed.

If the logic domain of the UMP contains supplementary tasks and/or flexible dependencies, the minimal (maximal) makespan of the project (henceforth, the total project time [TPT]) can be specified. When the supplementary tasks and all supplementary dependencies are excluded from (included), project Kosztyán (2015) (see the example in Figure 5) are called *minimal (maximal) project structures*, denoted as $\mathcal{S}_{\min}$ ($\mathcal{S}_{\max}$).

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 *maxi-min (minimax) project structures* denoted as $\mathcal{S}_{\text{maximin}}$ ($\mathcal{S}_{\text{minimax}}$) (see the left side of Figure 5 and Equations (2) through (5)).

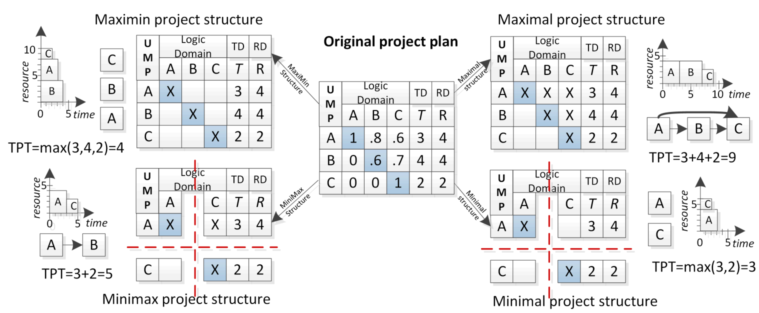


Figure 5: Minimal, maximal, minimax and maximin structures of the flexible project plan

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

The project plan indicators can be classified into two groups. The first group characterizes the project structure, including measures of its complexity, and the second group characterizes the project time-related and resource demands.

Table 2 summarizes the indicators of the project plans (exact definitions are given in subsections Appendix B.1 - Appendix B.3).

Table 2: Applied indicators

Name	Short description	Adapted from	Besides single mode single project, applicable for single mode multi-project	multi-mode single project	multi-mode multi-project	For the results see
Structural indicators						
I1	number of nodes (i.e., tasks)	Tavares (1999); Vanhoucke et al. (2008)	X	X	X	F8
I2	serial or parallel structure	Tavares (1999); Vanhoucke et al. (2008)	X	X	X	F8, F11a, F14
I3	task distribution	Tavares (1999); Vanhoucke et al. (2008)	X	X	X	F8
I4	rate of short arcs	Tavares (1999); Vanhoucke et al. (2008)	X	X	X	F8
I5	rate of long arcs	Tavares (1999); Vanhoucke et al. (2008)	X	X	X	F8
I6	topological float	Tavares (1999); Vanhoucke et al. (2008)	X	X	X	F8
T-DENSITY	total activity density	Patterson (1976)	X	X	X	F8
XDENSITY	average activity density	Patterson (1976)	X	X	X	F8
C	network complexity	Sprecher (1994)	X	X	X	F8, F11b, F14,F16
CNC	coefficient of network complexity	Davis (1975)	X	X	X	F8
OS	order strength	Mastor (1970)	X	X	X	F8
Time related indicators						
TPT	total project time		X	X	X	F9
XDUR	average activity duration	Patterson (1976)	X	X	X	F9
VA-DUR	variance in activity duration	Patterson (1976)	X	X	X	F9
PCTSLACK	percent of activities possessing positive total slack	Patterson (1976)	X			F9
XSLACK	average total slack per activity	Patterson (1976)	X			F9
TOTSLACK-R	total slack ratio	Patterson (1976)	X			F9, F13, F15, F16
XSLACK-R	average slack ratio	Patterson (1976)	X			F9, F13, F15
PCTFREESLACK	percent of activities possessing positive free slack	Patterson (1976)	X			F9
XFREESLACK	average free slack per activity	Patterson (1976)	X			F9
Renewable resource-related indicators						
RF	resource factor (i.e., density of **RD**)	Kolisch et al. (1995)	X			F10
$PCTR_j$	percent of activities that require resource type j	Patterson (1976)	X			F10
RU	resource use	Demeulemeester et al. (2003)	X			F10
$DMND_j$	the average demand resource type j	Patterson (1976)	X			F10
RC	resource constrainedness	Patterson (1976)	X			F10, F14, F15, F16
RS	resource strength	Kolisch et al. (1995)	X			F10
UTIL	utilization of resources	Patterson (1976)	X			F10
$TCON_j$	constraints of resource j over time	Patterson (1976)	X			F10
$OFACT_j$	obstruction of resource j	Patterson (1976)	X			F10, F14, F15
$UFACT_j$	underutilization of resource j	Patterson (1976)	X			F10

Table 2 shows that the characterization of both the project structure and demands has several indicators. However, flexibility has no indicators, and quality and cost demands have very few indicators. None of the indicators are interval indicators. This result indicates that the result of each indicator is a scalar or, in the case of multimode completions, a vector. However, in the case of flexible projects, several possible projects have different project demands; therefore, the indicators should be specified as an interval. Appendix B contains the detailed formal description of employed and extended indicators.

3. Research methods

3.1. Data

The different datasets and libraries mentioned in this paper were collected from the project-scheduling literature. During our research, we identified suitable data sources that are commonly used and shared by scholars to evaluate scheduling approaches and find the best solutions.

The first challenge is usually accessing 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 arises when the data must be addressed as they often have unique formatting and a structure that lacks proper documentation. This situation 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, is ready to process, and respects the original content. To overcome limitations, such as a lack of standardization and database integration efforts, we wrote a parser tool (a software program that reads inputs, e.g., a text file for further processing) for the most commonly used datasets found in the project-scheduling literature. The parser extracts all information from the existing libraries or 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 available formats, the aim is to continually extend the list of supported extensions. The following two main dataset categories are considered in our study: generated and empirical (see also Table 1). Our parser was written in MATLAB and works as follows. It reviews the existing project files in search of network-related data (tasks and their precedence relations); time-related and resource-related data, including demands and constraints, and if present, data of the costs and multiple modes of completion. Additional fields are captured from the original data files even if the input is not used directly for scheduling (e.g., the MPM-time field in the case of PSPLIB). The obtained data were 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 into MATLAB's workspace. The parser addresses renewable resource types, and the tool is designed such that it can be extended easily to use other types (e.g., nonrenewable and doubly constrained resource types). Since real-life projects and most simulated project databases contain neither these resource types nor quality parameters, these domains are not used in this study. From all 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 were not considered as these data were available in only one library.

3.2. Methods

Two indicator types are examined. The first group is *structural indicators*, such as complexity and flexibility indicators, which consider only the logic domain of the project domain matrices. The second group of indicators consists of *demand indicators*, which consider other domains, such as time domains (time-related indicators) and renewable resource indicators (such as renewable resource-related indicators).

An original logic structure of a project yields an activity-on-node network, which is denoted as a $G = (N, \mathcal{A})$ directed graph, where $N = \{A_1, ..., A_n\}$ (A_i is often shortened to i) is the set of nodes (i.e., tasks), and $\mathcal{A} \subset N \times N$ is the set of arcs (i.e., dependencies). $n = |N|$ is the number of tasks, and $|\mathcal{A}|$ is the number of dependencies. Furthermore, the matrix representation of the logic plan is the logic domain (LD) of the UMP matrix, where $\mathbf{LD} \in \{0,1\}^{n\times n}$, for each $i \leq n$ $[\mathbf{LD}]_{ii} = 1$, and for each $i \neq j$, we have $(A_i, A_j) \in \mathcal{A}$ if and only if $[\mathbf{LD}]_{i,j} = 1$ (otherwise $[\mathbf{LD}]_{i,j} = 0$).

Since none of the project databases considers flexible project structures, in the first step, the flexible project structures are generated. Let $\mathbf{LD} \in \{0,1\}^{n\times n}$ and $\mathbf{LD'} \in [0,1]^{n\times n}$ the modified logic domain as follows:

$$l'_{ij} = [\mathbf{LD'}]_{ij} := \begin{cases} u_{ij} \text{ if } l_{ij} = 1 \text{ and } v_{ij} \leq fp \\ l_{ij} \text{ otherwise} \end{cases} \tag{1}$$

where $l_{ij} = [\mathbf{LD}]_{ij}$, $u_{ij}, v_{ij} \sim U[0,1]$ are uniformly distributed random probability variables (r.v.), and $fp \in [0,1]$ is a fixed flexibility parameter we set for computer runs. We want the ratio of the number of (supplementary tasks + flexible dependencies) w.r.t. the total number of LD elements is approximately fp, which is ensured by "$v_{ij} \leq fp$". The *weights* of these flexible objects are set by the r.v. u_{ij}. Note that $\mathbf{LD'}$ already contains flexible dependencies ($i \neq j$) and supplementary tasks ($i = j$). However, complexity and time-related and resource-related indicators address only fixed project structures.

The modified logic domain is used to specify only the minimal, maximal, minimax and maximin structures as follows:

$$l_{ij}^{\min} = \left\lfloor l'_{ij} \right\rfloor, \tag{2}$$

$$l_{ij}^{\max} = \left\lceil l'_{ij} \right\rceil, \tag{3}$$

$$l_{ij}^{\text{minimax}} = \begin{cases} \left\lfloor l'_{ij} \right\rfloor & \text{if } i = j \\ \left\lceil l'_{ij} \right\rceil & \text{if } i \neq j \text{ and } \lfloor l'_{ii} \rfloor = \left\lfloor l'_{jj} \right\rfloor = 1 \\ 0 \text{ otherwise} \end{cases}, \tag{4}$$

$$l_{ij}^{\text{maximin}} = \begin{cases} \left\lceil l'_{ij} \right\rceil & \text{if } i = j \\ \left\lfloor l'_{ij} \right\rfloor & \text{if } i \neq j \end{cases}, \tag{5}$$

where $l_{ij}^{\min}, l_{ij}^{\max}, l_{ij}^{\text{minimax}}, l_{ij}^{\text{maximin}}$ are the (i,j) cells of the logic domains of the minimal, maximal, minimax and maximin structures, respectively, with $i, j = 1, 2, .., n$ (see Figure 5).[1)]

[1)] The $\lceil\cdot\rceil$ ($\lfloor\cdot\rfloor$) operators denote the rounding up (rounding down) of real numbers.

Minimal, maximal, minimax and maximin structures are also included in the databases. Of course, any other possible implementation structure can be specified by rounding up or down the cell values of the logic domain. However, in the case of single completion modes and the early schedule, the minimal structure provides the minimal task duration and minimal project budget, while a maximal structure provides the highest project score (widest project scope). In addition, the minimax (maximin) structure provides the highest (lowest) renewable resource demands.

To ensure comparability between the real and simulated databases, we examined primarily individual projects; however, both individual and multiple projects can be compared by calculating the average of the indicators per project. However, the calculation of most indicators differs in the single and multimode cases.

3.3. Structural indicators

Two structural indicator types are investigated in detail. The first group describes the rates of the flexible dependencies and supplementary tasks, and the second group describes the project structure complexity.

3.3.1. Structural flexibility

First, we set

$$S\text{-SET} := \{l'_{ii} | l'_{ii} \sim P(0,1), 0 < l'_{ii}\} \tag{6}$$

$$F\text{-SET} := \{l'_{ij} | l'_{ij} \sim P(0,1), i \neq j, 0 < l'_{ij}\} \tag{7}$$

where $P(0,1)$ is an arbitrary continuous distribution on interval]0,1[. Then, we let

fp = *flexibility parameter*, shows the total number of flexible dependencies and supplementary tasks across all tasks and dependencies as follows:

$$fp = \frac{|F\text{-SET} \cup S\text{-SET}|}{n(n+1)/2} \tag{8}$$

We set fp before the computer runs as the *approximate* ratio of flexible objects in Equation (1), while Equation (8) calculates the *exact* value of this ratio. Hereafter, we use this latter value of fp.

$f\%$ = *rate of flexible dependencies* shows the sum of flexible dependencies across all dependencies as follows:

$$f\% = \frac{|F\text{-SET}|}{n(n-1)/2} \tag{9}$$

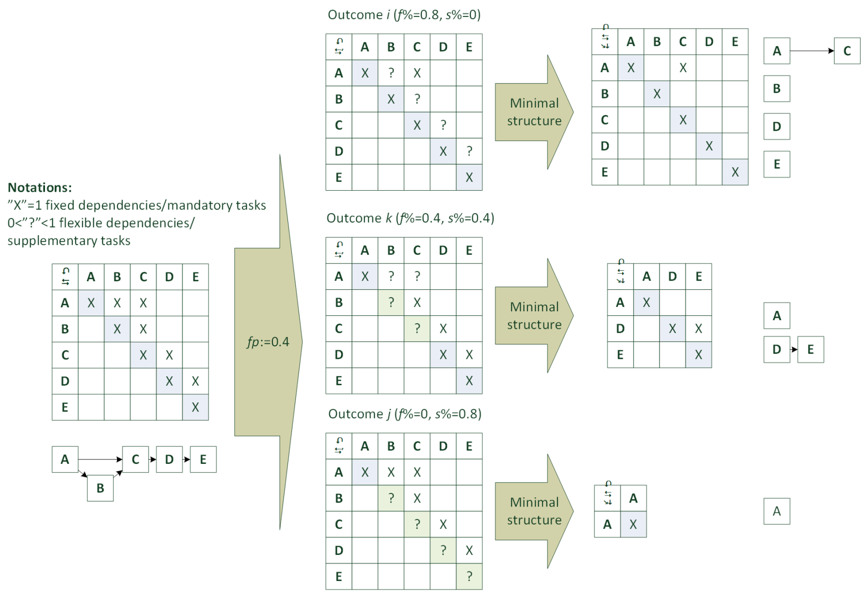


Figure 6: Example of generating flexibility

$s\% = \textit{rate of supplementary tasks}$ shows the sum of supplementary (prioritized) tasks across all tasks as follows:

$$s\% = \frac{|S\text{-SET}|}{n} \tag{10}$$

Observe that $fp = \frac{a+b}{c+d}$ if $f\% = \frac{a}{c}$ and $s\% = \frac{b}{d}$, which has the notation $\frac{a}{c} \boxplus \frac{b}{d} = \frac{a+b}{c+d}$. For a, b, c, d positive (which is our case) $\frac{a}{c} \boxplus \frac{b}{d}$ is always between $\frac{a}{c}$ and $\frac{b}{d}$. Thus, fp is always between $f\%$ and $s\%$, and all three depend only on Equation (1). Fig. 6 shows the mechanism of generating flexibility. The left side of the Fig. 6 shows the original logic domain, where the flexibility parameter is set to be 0.4. In the first step, fixed dependencies/mandatory tasks (denoted by the "X" symbol) become flexible (denoted by "?", where "?" indicates a number between 0 to 1). The right side of the Fig. 6 shows the minimal structure of the project. The center of Fig. 6 shows three possible outcomes from $\binom{10}{4}$. Because the number of "X" symbols is 10, we have $fp = 0.4$.

Outcome i retains all tasks, but cuts almost all dependencies, while outcome j retains only one task from the original project. In the general case, several dependencies are cut, and several tasks are omitted, see, e.g., in outcome k.

In subsections Appendix B.1 - Appendix B.3 below, we give the exact mathematical definitions of the indicators listed in Table 2. Using these indicators, we can compare the databases in Section 4, and the reader can decide based on these indicators which databases provide the closest match to the real case or problem they intend to study.

3.4. Applied multivariate analysis

In addition to the descriptive statistics, multivariate and network analyses were used to explore the relationships between the indicators. First, a correlation graph is specified between the indicators, represented by nodes, where the arcs represent the strength of the correlation between these nodes (i.e., indicators). The clustered correlation graph collects subsets of highly correlated indicators and groups them into a module by the Leiden method (Traag et al., 2019). In addition, the Force Atlas II (FA2) algorithm (Jacomy et al., 2014) arranges central indicators, which have many correlations between other variables, to the center of the module, and peripheric indicators are arranged at the edge of the correlation graph.

4. Results

4.1. Descriptive statistics–Data source comparison

Table 3 shows the number of projects in the 12 datasets of the 7 project databases.

Table 3: Descriptive statistics of the applied project database

(a) Descriptive statistics of the single project database

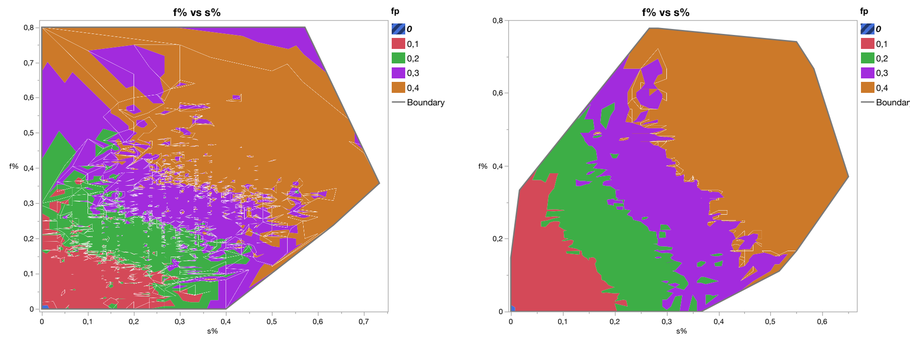
Database	**Set**	N	**original** fp**=0**	**0.1**	**0.2**	**0.3**	**0.4**
			Task number mean ($\overline{I}_1$)				
				minimal structures			
Boctor	Boctor	2160	75.00	67.38	60.09	52.40	44.81
Kolisch	SMCP	1800	29.00	26.16	23.29	20.51	17.40
	SMFF	4320	30.00	26.97	23.84	21.08	17.77
MMLIB	MMLIB50	4860	50.00	45.05	40.14	35.18	29.86
	MMLIB100	4860	100.00	89.94	80.00	70.10	59.97
	MMLIBPLUS	29160	75.00	67.50	60.05	52.54	44.85
Patterson	Patterson	990	24.02	21.73	19.51	16.85	14.91
PSPLIB	j30	5760	30.00	27.14	24.08	20.86	17.91
	j30sm	4320	30.00	27.06	24.02	21.11	17.78
Real-life	PROTRACK	1125	65.56	58.83	52.09	45.50	39.78
RG	RG30	16200	30.00	26.96	24.07	21.08	18.01
	RG300	4320	300.00	270.16	240.11	210.25	180.31

(b) Descriptive statistics of the multiple project database

Database	**Set**	N	**original** fp**=0**	**0.1**	**0.2**	**0.3**	**0.4**
			Mean of task numbers (by projects) ($\overline{I}_1$)				
				minimal structures			
BY	BY	110880	60.00	53.99	47.98	42.04	35.94
MPLIB1	Set 1	7497	360.00	324.08	287.95	251.59	216.39
	Set 2	13167	720.00	648.13	576.42	503.87	431.86
	Set 3	20286	1440.00	1296.34	1151.50	1007.89	863.79
MPLIB2	Set 1	91125	1000.00	900.19	800.08	700.10	599.91
	Set 2	77760	1000.00	900.40	800.39	700.30	600.16
	Set 3	77760	1000.00	900.01	799.77	700.09	599.89
	Set 4	69120	1000.00	899.88	800.25	700.17	600.29
MPSPLIB	MPSPLIB	1260	872.14	785.12	698.91	610.79	522.21
RCMPSPLIB	RCMPSPLIB	234	164.62	149.00	131.65	117.15	98.38

The total considered project number in a single project database was $79,875$. This value was nine times more than the original 8,875 projects. This result is due to the inclusion of both minimum and maximum structures in the database with four different flexibility parameter (fp) values. Most projects were derived from the MMLIBPLUS dataset (29,160) from the MMLIB database and the RG30 dataset from the RG database (16,200). The average task number within a project in the original databases was between 24 and 300 (see column $fp = 0$ in Table 3); this value decreased for minimal structures when the flexibility parameter (fp) was increased. The considered multiple project database contains 5 databases and 10 datasets. Considering demands by projects shows the same effects of increasing flexibility. Nevertheless, this database does not contain any real-life data; therefore, only simulated projects can be compared.

Figure 7 shows the relationship between the specified rate of constraints and the observed rates of the supplementary tasks and flexible dependencies.



(a) Single projects

(b) Multiple projects

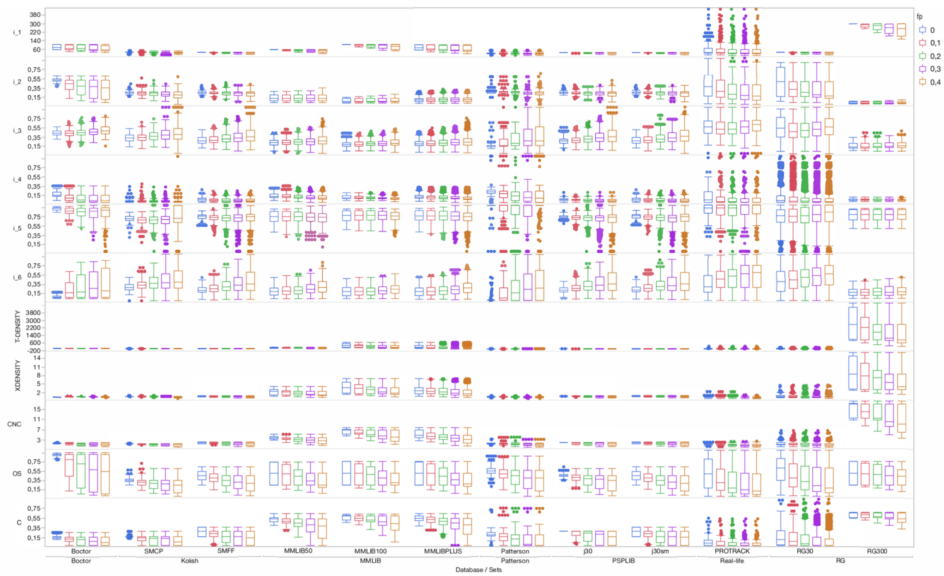
Figure 7: Observed rate of the supplementary tasks ($s\%$) and the flexible dependencies ($f\%$) by the flexibility parameter rate (fp)

fp is maximized to 40% for both theoretical and practical reasons. However, the expected value of f% and s% is 40% if fp% is 40%, which is in line with the guide of the DSDM (see Fig. 3), Fig 6. indicates that a further increase in the fp% above 40% might cause all tasks to be flexible and could be omitted or postponed in the minimal structure in which only mandatory tasks are completed. In addition, since we consider an iteration (sprint) as a logic plan, the number of flexible tasks may be higher than 40%. However, on average, this number should not be greater than 40%. In the case of hybrid projects, the number of flexible tasks is less than that in agile ones; therefore, fp between 0.0 to 0.4 well simulates the traditional-hybrid-agile transitions.

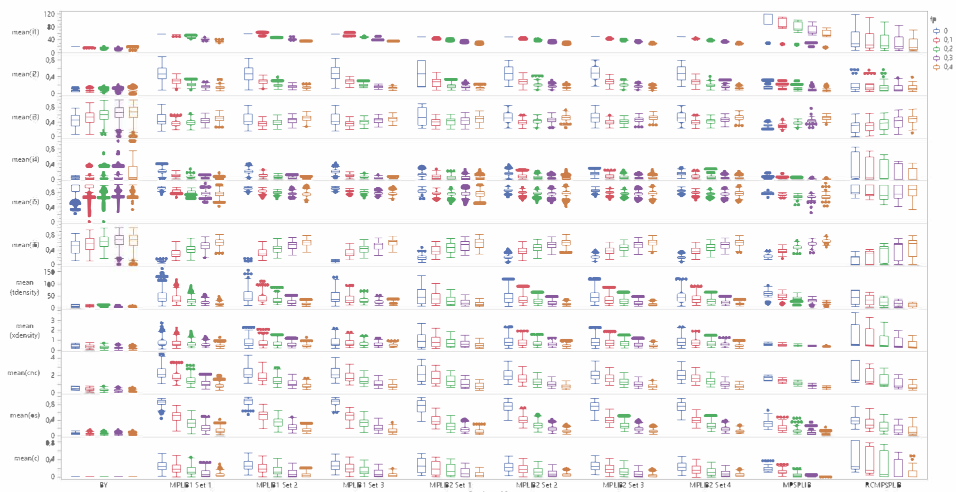
Figure 7 shows that the observed rates of the supplementary tasks and those of flexibility dependencies covered the most combinations of the flexibility parameters.

4.2. Flexibility effects on the indicators

Figure 8 shows a comparison of the structural indicators in the 22 datasets with 5 different flexibility parameters.



(a) Single project database



(b) Multiple project database

Figure 8: Flexibility effects on the structural indicators

Figure 8 shows that the considered datasets provide various complexity values. Regarding most complexity measures, such as $I_1 - I_6$, OS, and C, the real-life database covers the greatest intervals of the structure-related and complexity related values, while regarding the CNC, T-DENSITY, and X-DENSITY indicators, the RG300 datasets cover the most possible values. Nevertheless, generally, the flexibility extends to the covered intervals of the structural indicators in all datasets. Nevertheless, the multiple project databases do not contain real-life datasets. Thus, the comparison between the simulated and real-life database can only be analyzed in a single project database. Therefore, we focused on a single project database; however, all results based on a multiple project database can be found in the Appendix A.

Figure 9 compares the time-related indicators of projects from the 7 single project databases and 12 datasets.

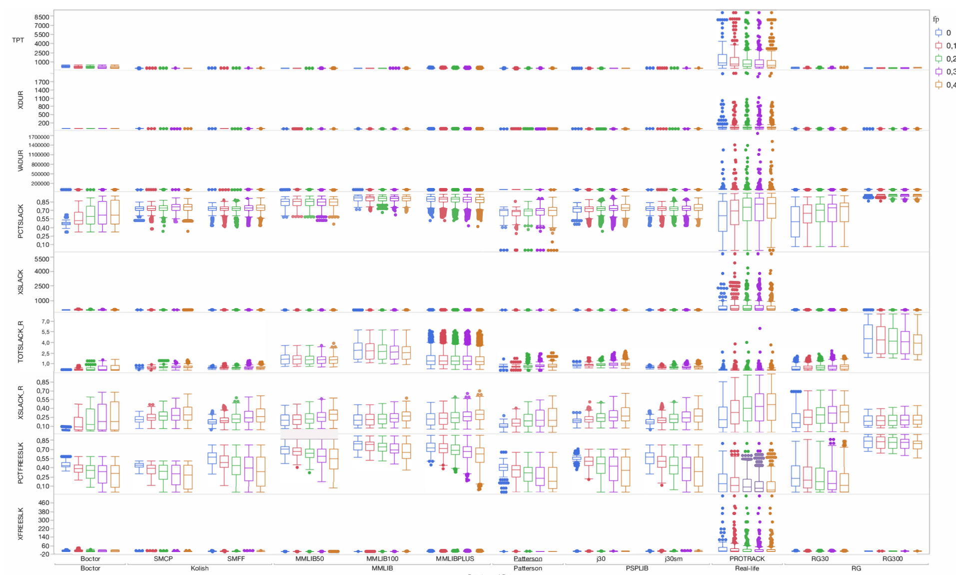


Figure 9: Flexibility effects on the time-related indicators

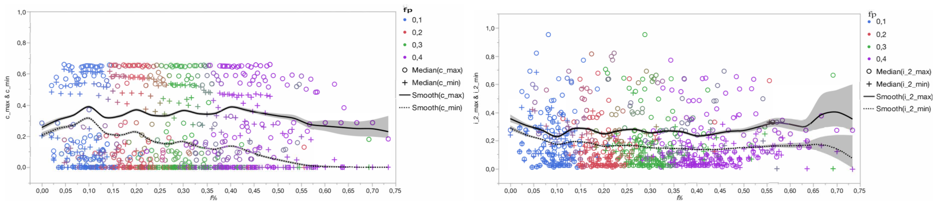
Figure 9 also shows that the real-life database and the RG300 dataset covered most possible values of the time-related measures/indicators. Nevertheless, despite the spread of the time-related value intervals induced by considering flexibility, the real-life database covered significantly more possible values of the time-related indicators. Without considering flexibility, any single simulated database focuses on a narrow interval of time-related indicators that can be very far from real-life project values.

Figure 10 shows a comparison of the resource-related indicators of the projects from the 7 databases and their 12 datasets.

Figure 10: Flexibility effects on the resource-related indicators

The difference between the simulated and real-life projects based on the resource-related indicators can also be identified in Figure 10. Nevertheless, in contrast to the time-related indicators, Figure 10 shows that the MMLIBPLUS dataset provided resource-related indicator values, e.g., the resource strength (RS) values, 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 varied in a wider range in the real-life database. In all cases, by introducing flexibility to the project structures and including the generated minimal structures, the interval of the possible values of the structure-related, time-related, and resource-related indicators can be widened and brought closer to the values of the real-life database. We broadened the interpretation ranges of the indicators of multiple projects; see Figure A.17 in the Appendix A.

Figure 11 compares the complexity (C) and parallelization (I_2) values of the minimal and maximal structures regarding the ratio of flexible dependencies ($f\%$) (marked on the horizontal axis).



(a) Flexibility vs complexity

(b) Flexibility vs parallelization

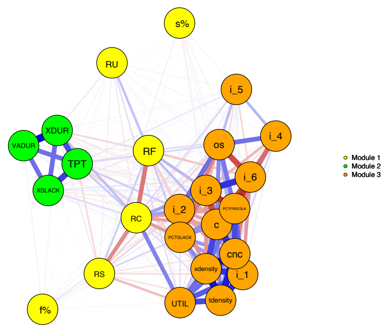
Figure 11: Structural changes in complexity and parallelization

Figure 11 shows that when the flexibility parameter (fp) was increased via an increase in the rate of flexibility dependencies ($f\%$) in the minimal structures, the complexity (C) decreased (see Figure 11(a)) as did the serial completions (see Figure 11(b)).

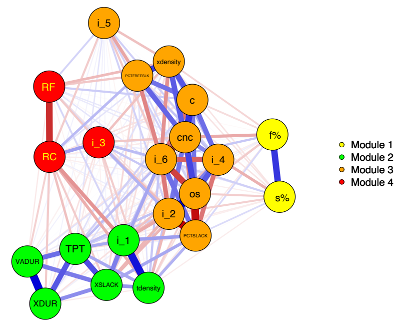
4.3. Flexibility effects on indicator interdependence

Figure 12 shows the clustered correlation graph between the indicators in the single-project database detailed in Section 3.4.

One interpretation of Figure 12 is that several redundant indicators were highly correlated with each other. This was especially true among the topological indicators (Module 3). In comparison, the proposed ($s\%, f\%$) flexibility indicators were located on the periphery and in another module (i.e., in Module 2), suggesting that although they are related to the other indicators, they should not be merged with them. Another finding is that the modules in the simulated datasets were quite well provided with the structure-related, time-related and resource-related indicators, where the complexity (C), resource constrainedness (RC), and project duration (TPT)) played central roles. Furthermore, the real-life dataset provided more mixed modules. Thus, the correlation direction did not change, four modules were specified, and at least one structural indicator was included in all modules, indicating the greater importance of the structural indicators in the description of real-life projects. The separation of the three modules can also be considered in the case of multiple projects (see Figure A.18). When Figures 12 and A.18 are compared, more significant differences can be observed between the simulated and real-life indicators than between the single and multiple project indicators. The multiple project database also produced three modules. Nevertheless, these modules were more mixed than those in the single-project cases.



(a) For the simulated projects



(b) For the real-life database (RS,$UTIL$ cannot be used in correlation graph)

Figure 12: Clustered correlation graph between the indicators (Notes: Only significant correlations are represented. The correlation strengths are proportional to the tightness of the arcs between the nodes. The blue (red) arcs indicate positive (negative) correlations. Applied grouping was accomplished using the Leiden modularity-based community detection method. The nodes are represented only where there is variance).

Flexibility considerations not only expand the interval of the indicator values but also specify new value pairs for the coupled indicators. Figure 13 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 the indicator values. Figure 13 shows the pairs of the indicator values of the total slack ratio (TOTSLACK-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.

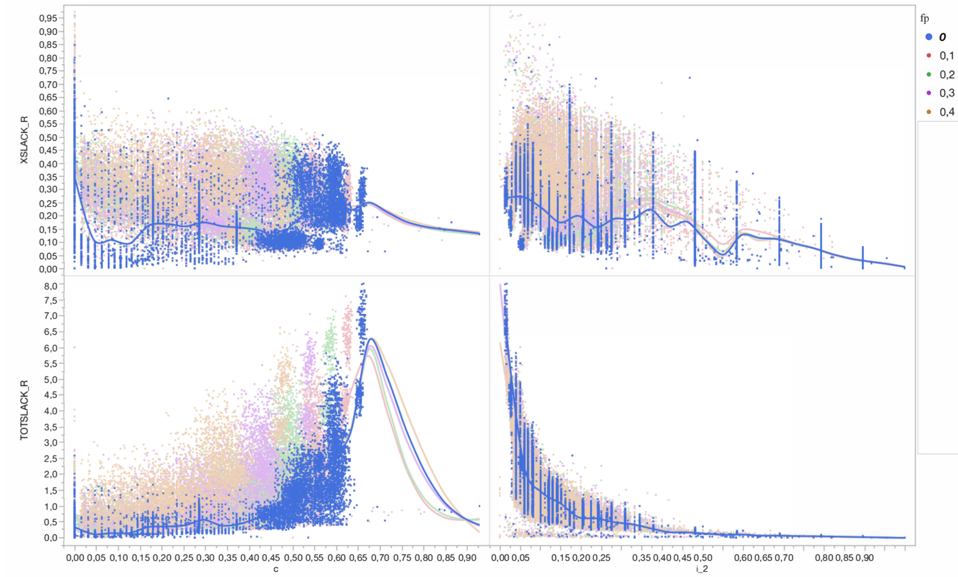


Figure 13: Flexibility effects on the relations between the time-related and complexity indicators

Figure 13 shows that including minimal structures helps explore new areas on the planes spanned by the structure-related and time-related indicator pairs. These combinations better cover the area of the possible value pairs. Flexibility can also be expressed in other ways as follows: the minimal structures of flexible projects have higher average slacks, which can be better utilized in resource allocation. When the minimal structures of flexible projects are included, the domain is better covered if a combination of (1) resource-related indicators, such as the mean of resource constrainedness ($\overline{\mathrm{RC}}$)/the mean of the obstruction factor ($\overline{\mathrm{OFACT}}$, and (2) a structural indicator, such as complexity (C)/parallelization (I_2), is studied (see Figure 14).

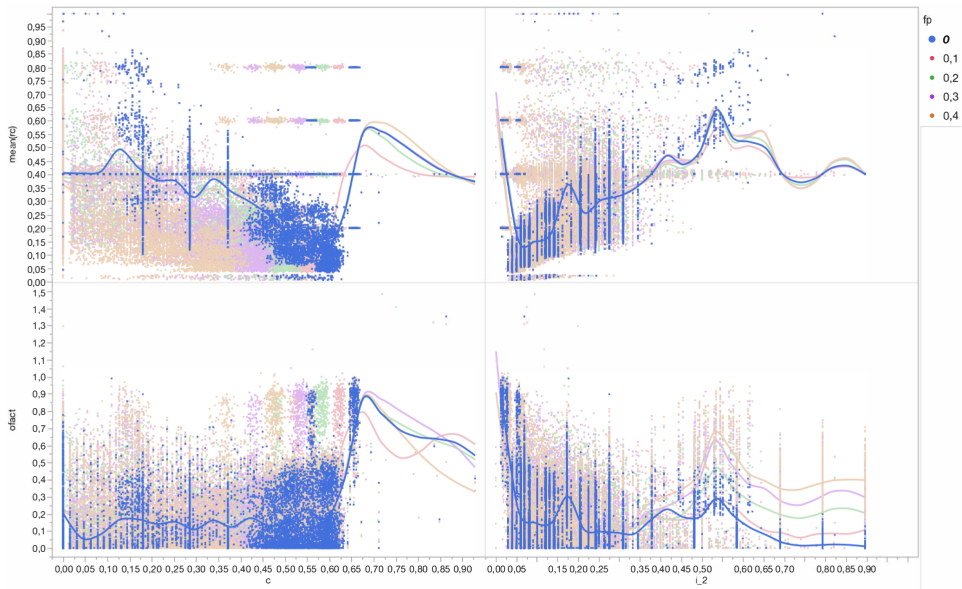


Figure 14: Flexibility effects on the relations between the resource-related and complexity indicators

Figure 14 shows that while minimal structures decreased the complexity (C) and increased the parallelization (i.e., decrease serialization) (I_2), they also increased the obstruction factor and the resource constrainedness.

Figure 15 shows the relations between the slack ratios (TOTSLACK-R, XSLACK-R) and the resource-related indicators in the earliest start schedule. Considering the minimal structures of flexible projects increases the slack ratio the resource constrainedness, and the obstruction factor because of the parallelization. These combinations of time-related and resource-related indicator values occurred only in flexible project plans.

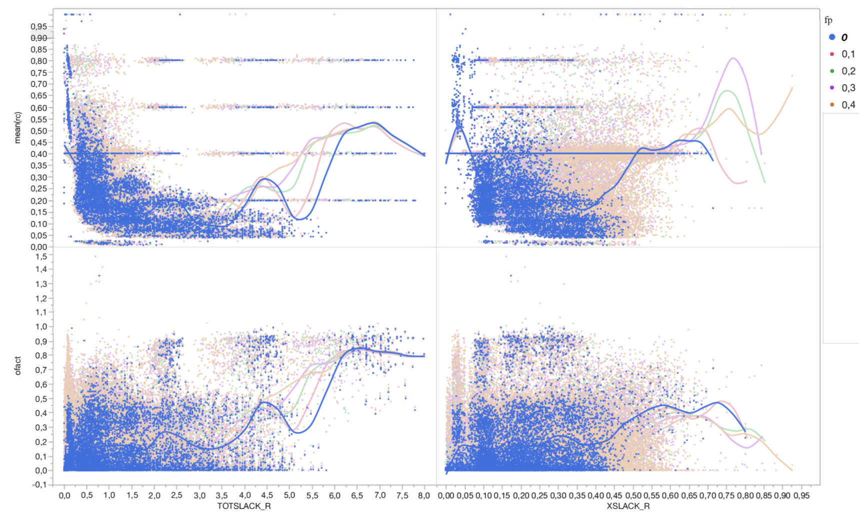
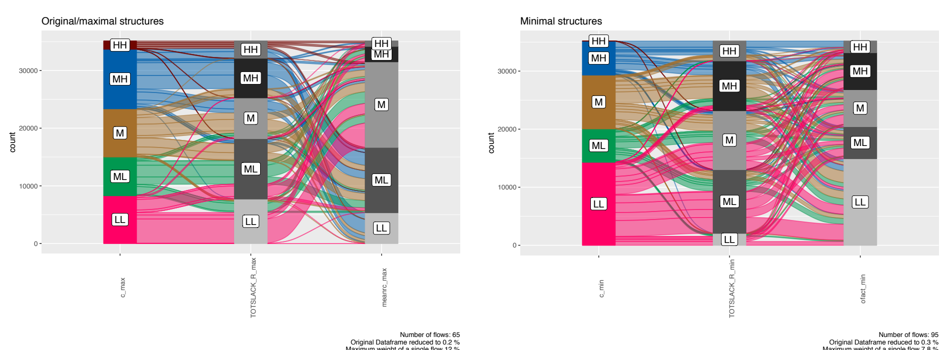


Figure 15: Flexibility effects on the relations between the time-related and the resource-related indicators

Figure 16 shows the mutual effect of flexibility on a structure-related (C), a time-related (TOTSLACK-R) and a resource-related (RC) indicator. Considering flexibility can reduce complexity (compare Figures 16(a) and (b)) while it increases the slack ratios and reduces the resource constrainedness (RC).



(a) Original/maximal structures

(b) Minimal structures

Figure 16: Alluvial diagrams of the complexity, time-related and resource-related indicators

5. Discussion

5.1. Evaluation of the project library comparison

When testing project-scheduling and resource allocation algorithms only in simulated databases, two error types can be made. The first problem is whether new algorithms are applied to real-life projects that have different types of complexity (see Figure 8, time-related (see Figure 9) or resource-related (see Figure 10) indicator values than simulated projects in (benchmark) databases. Even if scheduling simulated projects is more difficult for the current objectives and algorithms, these algorithms may not be prepared for the challenges of the new objectives often found in real-life projects. Creating a specified database tailored to one type of problem can cause discrepancies in real-life usage because of indirect constraints rooted in unconsidered properties. Second, if the algorithms are optimized to properties of simulated projects that never appear in real life, resources are squandered. An interesting result is that the differences in the indicator values are much larger between simulated and real-life projects than they are between individual and multiple projects (compare Figures 8-10 and Figures A.17(a-b)). The relationship between the indicators illustrated by the clustered correlation graph (see Figures 12 and A.18) also shows significantly different results, mainly between the simulated and traditional projects. We could not include a real-life multiple project database in our study as we have not yet found one. Therefore, we must focus on the individual project databases, but we consider it essential to examine real-life projects, when possible, to study schedules. The simulated datasets should also be combined because an individual dataset usually covers only a small range of an indicator (see Figures 8-10).

Figures 8-10, A.17 also show that including minimal structures (see Figure 5) widened the indicator intervals; therefore, even if flexible structures are not studied, the extended dataset may cover larger indicator intervals.

Table 4 compares the simulated and real-life databases. The indicators from the two groups, i.e., (1) a real-life database and (2) simulated datasets, were compared by an ANOVA. Table 4 shows the number of indicators that had significantly different values between these groups.

Table 4: Number of significantly different indicators between the simulated and the real-life databases ($pvalue = 0.01$)

Indicators	$fp = 0$	$fp = 0.1$	$fp = 0.2$	$fp = 0.3$	$fp = 0.4$	All
Structural	11/13	11/13	10/13	10/13	11/13	11/13
Time-related	9/9	9/9	9/9	9/9	9/9	9/9
Resource-related	9/11	8/11	8/11	8/11	8/11	8/11

When flexibility and generating minimal structures are considered, the indicator interval 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-related and resource-related measures remain significantly different between the real-life and simulated databases.

Figure 11 shows that an increase in flexibility reduces complexity and increases 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). However, Figure 12(a) shows that especially in the simulated databases where resource constraints are prespecified, structural flexibility correlates with the resource-related indicators. For real-life projects, structural flexibility forms a separated module. In contrast to the simulated projects, the structural flexibility indicators mainly correlated with the other structural and topological indicators; because of the lack of resource constraints, indicators RS and UTIL could not be calculated.

5.2. Flexibility effects on demands

Considering flexibility not only widens the indicator intervals but also specifies new demand combinations. Figures 13-15 indicate that including minimal structures of flexible projects covered more of the domain. The new combination of indicators specified new structures that have never been tested by project-scheduling and resource allocation algorithms. However, the fact that flexible projects are becoming increasingly popular implies that tasks must be prioritized and technological dependencies must be rethought. Minimal structures have the advantage of eliminating the need to use algorithms. Existing algorithms can be tested in new structures generated by the FSG. Nevertheless, maintaining flexibility values, flexible project-planning and scheduling algorithms can also be tested in a large set of project databases.

6. Summary and conclusion

In this study, a unified matrix-based project-planning model (UMP) is proposed to model heterogeneous project plans. To combine heterogeneous project databases, a compound matrix-based project database (CMPD) is proposed. In addition, a flexible structure generator (FSG) is proposed to extend the existing project databases to address possible structures of flexible project plans (see Table 5). The proposed minimal and maximal structures specify new combinations of the structural and demand indicator values to test algorithms in flexible project management environments.

As Table 5 highlights the applicability of the proposed models and methods, the UMP addresses both individual and multiple projects, single and multimodal completions. In addition, it handles renewable and nonrenewable resources, cost, and quality parameters, which are essential in real-life projects. In addition, The unified database contains both simulated and real-life data sources. 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. However, to our best knowledge, there is no existing real-life database for multi-projects and multimodal completion modes. Therefore, the proposed model and methods cannot be tested in these types of real-life projects. The proposed parsers and generators are freely available at https://github.com/novakge/parsers.

The proposed matrix-based model addresses cost and nonrenewable demands and quality parameters and manages multiple completion modes and multilevel projects. Nevertheless, to ensure the comparison between simulated and real-life projects, this study examined mainly a single-project, single-mode environment with time and renewable resource demands. The proposed matrix-based model not only unifies heterogeneous databases but also allows the user to test both traditional and flexible project-scheduling algorithms.

Table 5: Summary table of employed models, generators and databases, and limitations

Data source	**Type of projects**	**Comletion modes**	**UMP Traditional**	**UMP Flexible**	**CMPD Traditional**	**CMPD Flexible**	**FSG**	**Analyzed? Traditional**	**Analyzed? Flexible**
Simulated	Single project	Single-mode	X	X	X	X	X	X	X
Simulated	Single project	Multi-mode	X	X	X	X	X	-	-
Simulated	Multi-project	Single-mode	X	X	X	X	X	?	?
Simulated	Multi-project	Multi-mode	X	X	-	-	X	-	-
Real-life	Single project	Single-mode	X	X	X	X	X	X	X
Real-life	Single project	Multi-mode	-	-	-	-	-	-	-
Real-life	Multi-project	Single-mode	-	-	-	-	-	-	-
Real-life	Multi-project	Multi-mode	-	-	-	-	-	-	-

Notations: '+' addressed, '-' not addressed, '?' partly addressed.

Nomenclature

Latin symbols

$a_i(T)$ scheduled execution time interval of task a_i
a_i task i
D maximal number of short arcs
EF_i, LF_i early and late finish for task i
ES_i, LS_i early and late start to task i
$f\%$ rate of flexible dependencies
fp flexibility parameter, the ratio of flexible dependencies and prioritized tasks to all tasks and dependencies
FS_i free slack of task i
k number of task completion modes
$L(i_1, i_2)$ length of an arc between tasks i_1 and i_2, i.e., the difference between their progressive level numbers
$\overrightarrow{L}$ longest (critical) path
$l_{ij} = [\mathbf{LD}]_{ij}$ element of the logic domain, task occurrence if $i = j$, and arc that represent the precedence relation between tasks i and $j \neq i$ (in this case, $l_{ij} = 1$ means task i precedes task j)
n number of tasks
n'_L number of arcs with length L
m maximal number of progressive levels
$\overrightarrow{P}$ task path (sequence)
P_i set of immediate predecessors of task i
r_{ij} demand of task i for renewable resource type j
$r_{ij}(\tau)$ demand of task i for renewable resource type j at time τ
$s\%$ rate of supplementary tasks
S_i set of immediate successors of task i
t_i duration of task a_i
TS_i total slack of task i
w_i width of progressive level i, $i = 1, ..., m$

Greek symbols

α_j availability of renewable resource type j
α_w total absolute deviation from the average width
η number of types of nonrenewable resources
ρ number of types of renewable resources

Calligraphic symbols

$\mathcal{A}$ set of arcs (dependencies)
$|\mathcal{A}|$ number of dependencies in a project structure
$\mathcal{S}$ project structure, set of (to-be-) realized tasks
$\overrightarrow{\mathcal{S}}$ project schedule of project structure $\mathcal{S}$

Abbreviations

CD cost domain of the UMP
CMPD compound matrix-based project database
FSG flexible structure generator
LD logic domain of the UMP
TD time domain of the UMP
QD quality domain of the UMP
ND nonrenewable resource domain of the UMP
PDM project domain matrix
PL progressive level
RD renewable resource domain of the UMP
RL regressive level
UMP unified matrix-based project-planning model

CRediT authorship statement

Zsolt T. Kosztyán: Conceptualization, Methodology, Software, Writing-Original draft preparation, Writing-Reviewing and Editing **Gergely Novák**: Data curation, Software. Writing-Reviewing and Editing **Róbert Jakab**: Investigation, Validation. **István Szalkai**: Methodology, Writing-Reviewing and Editing. **Csaba Hegedűs**: Writing-Reviewing and Editing.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

Funding

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. Gergely Novák's cooperation was supported of the Doctoral Student Scholarship Program of the Co-Operative Doctoral Program of the Ministry of Innovation and Technology financed by the National Research, Development, and Innovation Fund. Csaba Hegedűs's research contribution supported by Project No. PD 123915 provided from the National Research, Development and Innovation Fund of Hungary, financed under the PD_17 postdoctoral funding scheme, while Zsolt T. Kosztyán's research contribution supported by Project no. K 142395 has been implemented with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the K_22 "OTKA" funding scheme.

References

Abad, Z.S.H., Sadi, M.H., Ramsin, R., 2010. Towards tool support for situational engineering of agile methodologies, in: 2010 Asia Pacific Software Engineering Conference, IEEE. pp. 326–335. doi:10.1109/APSEC.2010.45.

Batselier, J., Vanhoucke, M., 2015. Construction and evaluation framework for a real-life project database. International Journal of Project Management 33, 697–710. doi:10.1016/j.ijproman.2014.09.004.

Boctor, F.F., 1993. Heuristics for scheduling projects with resource restrictions and several resource-duration modes. The international journal of production research 31, 2547–2558. doi:10.1080/00207549308956882.

Browning, T.R., Yassine, A.A., 2010. A random generator of resource-constrained multi-project network problems. Journal of Scheduling 13, 143–161. doi:10.1007/s10951-009-0131-y.

Brucker, P., Drexl, A., Möhring, R., Neumann, K., Pesch, E., 1999. Resource-constrained project scheduling: Notation, classification, models, and methods. European journal of operational research 112, 3–41. doi:10.1016/S0377-2217(98)00204-5.

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 Manufacturing 39, 1407–1414. doi:10.1016/j.promfg.2020.01.314.

Ciriello, R.F., Glud, J.A., Hansen-Schwartz, K.H., 2022. Becoming agile together: Customer influence on agile adoption within commissioned software teams. Information & Management 59, 103645. doi:10.1016/j.im.2022.103645.

Coelho, J., Vanhoucke, M., 2020. Going to the core of hard resource-constrained project scheduling instances. Computers & Operations Research 121, 104976. doi:10.1016/j.cor.2020.104976.

Creemers, S., Reyck, B.D., Leus, R., 2015. Project planning with alternative technologies in uncertain environments. European Journal of Operational Research 242, 465 – 476. doi:10.1016/j.ejor.2014.11.014.

Davis, E.W., 1975. Project network summary measures constrained-resource scheduling. AIIE Transactions 7, 132–142. doi:10.1080/05695557508974995.

Debels, D., Vanhoucke, M., 2007. A decomposition-based genetic algorithm for the resource-constrained project-scheduling problem. Operations Research 55, 457–469. doi:10.1287/opre.1060.0358.

Demeulemeester, E.L., Herroelen, W.S., Elmaghraby, S.E., 1996. Optimal procedures for the discrete time/cost trade-off problem in project networks. European Journal of Operational Research 88, 50–68. doi:10.1016/0377-2217(94)00181-2.

Demeulemeester, E.L., Vanhoucke, M., Herroelen, W., 2003. Rangen: A random network generator for activity-on-the-node networks. Journal of Scheduling 6, 17–38. doi:10.1023/A:1022283403119.

Denizer, C., Kaufmann, D., Kraay, A., 2013. Good countries or good projects? macro- and microcorrelates of world bank project performance. Journal of Development Economics 105, 288–302. doi:10.1016/j.jdeveco.2013.06.003.

Dingsøyr, T., Nerur, S., Balijepally, V., Moe, N.B., 2012. A decade of agile methodologies: Towards explaining agile software development. doi:10.1016/j.jss.2012.02.033.

Drexl, A., Nissen, R., Patterson, J.H., Salewski, F., 2000. Progen/πx–an instance generator for resource-constrained project scheduling problems with partially renewable resources and further extensions. European Journal of Operational Research 125, 59–72.

Fernandez, D.J., Fernandez, J.D., 2008. Agile project management: agilism versus traditional approaches. Journal of Computer Information Systems 49, 10–17. doi:10.1080/08874417.2009.11646044.

Franco-Duran, D.M., Garza, J.M.d.l., 2019. Review of resource-constrained scheduling algorithms. Journal of Construction Engineering and Management 145, 03119006. doi:10.1061/(ASCE)CO.1943-7862.0001698.

Govil, N., Sharma, A., 2021. Information extraction on requirement prioritization approaches in agile software development processes, in: 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), IEEE. pp. 1097–1100. doi:10.1109/ICCMC51019.2021.9418285.

Hartmann, S., Briskorn, D., 2010. A survey of variants and extensions of the resource-constrained project scheduling problem. European Journal of operational research 207, 1–14. doi:10.1016/j.ejor.2009.11.005.

Hauder, V.A., Beham, A., Raggl, S., Parragh, S.N., Affenzeller, M., 2020. Resource-constrained multi-project scheduling with activity and time flexibility. Computers & Industrial Engineering 150, 106857. doi:10.1016/j.cie.2020.106857.

Hidalgo, E.S., 2019. Adapting the scrum framework for agile project management in science: case study of a distributed research initiative. Heliyon 5, e01447. doi:10.1016/j.heliyon.2019.e01447.

Homberger, J., 2007. A multi-agent system for the decentralized resource-constrained multi-project scheduling problem. International Transactions in Operational Research 14, 565–589. doi:10.1111/j.1475-3995.2007.00614.x.

Jacomy, M., Venturini, T., Heymann, S., Bastian, M., 2014. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. Plos One 9, 1–12. doi:10.1371/journal.pone.0098679.

Kellenbrink, C., Helber, S., 2015. Scheduling resource-constrained projects with a flexible project structure. European Journal of Operational Research 246, 379–391. doi:10.1016/j.ejor.2015.05.003.

Kolisch, R., Sprecher, A., Drexl, A., 1995. Characterization and generation of a general class of resource-constrained project scheduling problems. Management Science 41, 1693–1703. doi:10.1287/mnsc.41.10.1693.

Kosztyán, Z.T., 2015. Exact algorithm for matrix-based project planning problems. Expert Systems with Applications 42, 4460 – 4473. doi:10.1016/j.eswa.2015.01.066.

Kosztyán, Z.T., 2020. An exact algorithm for the flexible multilevel project scheduling problem. Expert Systems with Applications , 113485doi:10.1016/j.eswa.2020.113485.

Kosztyán, Z.T., Pribojszki-Németh, A., Szalkai, I., 2019. Hybrid multimode resource-constrained maintenance project scheduling problem. Operations Research Perspectives 6, 100129. doi:10.1016/j.orp.2019.100129.

Kosztyán, Z.T., Szalkai, I., 2020. Multimode resource-constrained project scheduling in flexible projects. Journal of Global Optimization 76, 211–241. doi:10.1007/s10898-019-00832-8.

Mastor, A.A., 1970. An experimental investigation and comparative evaluation of production line balancing techniques. Management Science 16, 728–746. doi:10.1287/mnsc.16.11.728.

Minogue, P., 2011. "Gantt-Like" DSMs, in: DSM 2011: Proceedings of the 13th International DSM Conference, pp. 259–271. URL: https://www.designsociety.org/publication/30838/%22Gantt-Like%22+DSMs.

Özkan, D., Mishra, A., 2019. Agile project management tools: A brief comprative view. Cybernetics and Information Technologies 19, 17–25. doi:10.2478/cait-2019-0033.

Patterson, J.H., 1976. Project scheduling: The effects of problem structure on heuristic performance. Naval Research Logistics Quarterly 23, 95–123. doi:10.1002/nav.3800230110.

Peteghem, V.V., Vanhoucke, M., 2014. An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances. European Journal of Operational Research 235, 62 – 72. doi:10.1016/j.ejor.2013.10.012.

Ren, R., Zhang, J., Jiang, Y., 2021. New automated activity-on-node calculation grading method for construction management education innovation. Journal of Civil Engineering Education 147, 04021004. doi:10.1061/(ASCE)EI.2643-9115.0000043.

Schwindt, C., Schwindt, C., 1995. ProGen/max: A New Problem Generator for Different Resource-Constrained Project Scheduling Problems with Minimal and Maximal Time Lags. Technical Report. CiteSeer. URL: https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.45.5804.

Servranckx, T., Vanhoucke, M., 2019. Strategies for project scheduling with alternative subgraphs under uncertainty: similar and dissimilar sets of schedules. European Journal of Operational Research 279, 38–53. doi:10.1016/j.ejor.2019.05.023.

SGI, 2015. CHAOS Manifesto. Standish Group International. URL: https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf.

Sprecher, A., 1994. Generation of Instances by ProGen, in: Resource-Constrained Project Scheduling. Springer, pp. 70–90. doi:10.1007/978-3-642-48397-4_6.

Sprecher, A., Kolisch, R., 1996. PSPLIB-a project scheduling problem library. European Journal of Operational Research 96, 205–216. doi:10.1016/S0377-2217(96)00170-1.

Srivastava, A., Mehrotra, D., Kapur, P., Aggarwal, A.G., 2021. Measuring and evaluating best practices in agile testing environment using ahp, in: Advances in Interdisciplinary Research in Engineering and Business Management. Springer, pp. 475–495. doi:10.1007/978-981-16-0037-1_37.

Stapleton, J., 1997. DSDM, dynamic systems development method: the method in practice. Cambridge University Press.

Stare, A., 2014. Agile project management in product development projects. Procedia - Social and Behavioral Sciences 119, 295–304. doi:10.1016/j.sbspro.2014.03.034.

Tao, S., Dong, Z.S., 2018. Multi-mode resource-constrained project scheduling problem with alternative project structures. Computers & Industrial Engineering 125, 333–347. doi:10.1016/j.cie.2018.08.027.

Tavares, L.V., 1999. Advanced models for project management. volume 16. Springer Science & Business Media. URL: 10.1007/978-1-4419-8626-9, doi:10.1007/978-1-4419-8626-9.

Traag, V.A., Waltman, L., van Eck, N.J., 2019. From louvain to leiden: guaranteeing well-connected communities. Scientific Reports 9, 5233. doi:10.1038/s41598-019-41695-z.

Van Eynde, R., Vanhoucke, M., 2020. Resource-constrained multi-project scheduling: benchmark datasets and decoupled scheduling. Journal of Scheduling 23, 301–325. doi:10.1007/s10951-020-00651-w.

Vanhoucke, M., 2010a. A scatter search heuristic for maximising the net present value of a resource-constrained project with fixed activity cash flows. International Journal of Production Research 48, 1983–2001. doi:10.1080/00207540802010781.

Vanhoucke, M., 2010b. Using activity sensitivity and network topology information to monitor project time performance. Omega 38, 359–370. doi:10.1016/j.omega.2009.10.001.

Vanhoucke, M., Coelho, J., 2018. A tool to test and validate algorithms for the resource-constrained project scheduling problem. Computers & Industrial Engineering 118, 251–265. doi:10.1016/j.cie.2018.02.001.

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. European Journal of Operational Research 187, 511–524. doi:doi.org/10.1016/j.ejor.2007.03.032.

Vanhoucke, M., Demeulemeester, E.L., Herroelen, W., 2001. On maximizing the net present value of a project under renewable resource constraints. Management Science 47, 1113–1121. doi:10.1287/mnsc.47.8.1113.10226.

Vázquez, E.P., Calvo, M.P., Ordóñez, P.M., 2015. Learning process on priority rules to solve the RCMPSP. Journal of Intelligent Manufacturing 26, 123–138. doi:10.1007/s10845-013-0767-5.

Čapek, R., Šůcha, P., Hanzálek, Z., 2012. Production scheduling with alternative process plans. European Journal of Operational Research 217, 300–311. doi:10.1016/j.ejor.2011.09.018.

Wauters, T., Kinable, J., Smet, P., Vancroonenburg, W., Vanden Berghe, G., Verstichel, J., 2016. The multi-mode resource-constrained multi-project scheduling problem. Journal of Scheduling 19, 271–283. doi:10.1007/s10951-014-0402-0.

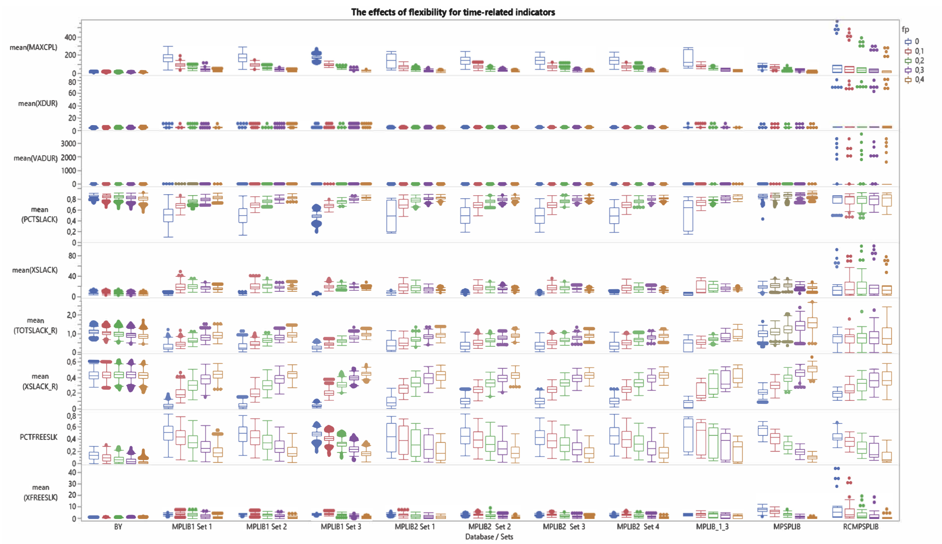
Williams, L., 2010. Agile software development methodologies and practices, in: Zelkowitz, M.V. (Ed.), Advances in Computers. Elsevier. volume 80 of *Advances in Computers*, pp. 1–44. doi:10.1016/S0065-2458(10)80001-4.

World Bank, 2012. The little data book on financial inclusion 2012. World Bank Publications. doi:10.1596/978-0-8213-9509-7.

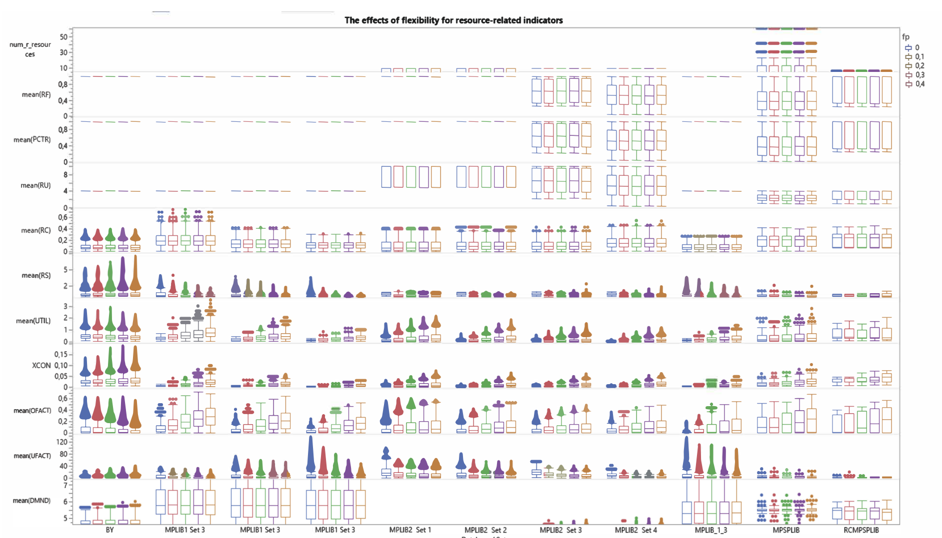
Wysocki, R.K., 2019. Effective project management. 8 ed., John Wiley & Sons, Nashville, TN.

Yasaman, A., Nazari, A., Hafezi, M.R., Maghool, S.A.H., 2022. Prioritizing agile project management strategies as a change management tool in construction projects. International Journal of Construction Management 22, 678–689. doi:10.1080/15623599.2019.1644757.

Appendix A.



(a) The effects of flexibility for time-related indicators



(b) The effects of flexibility for resource-related indicators

Figure A.17: Flexibility effects on the demand-related indicators among the multiple projects

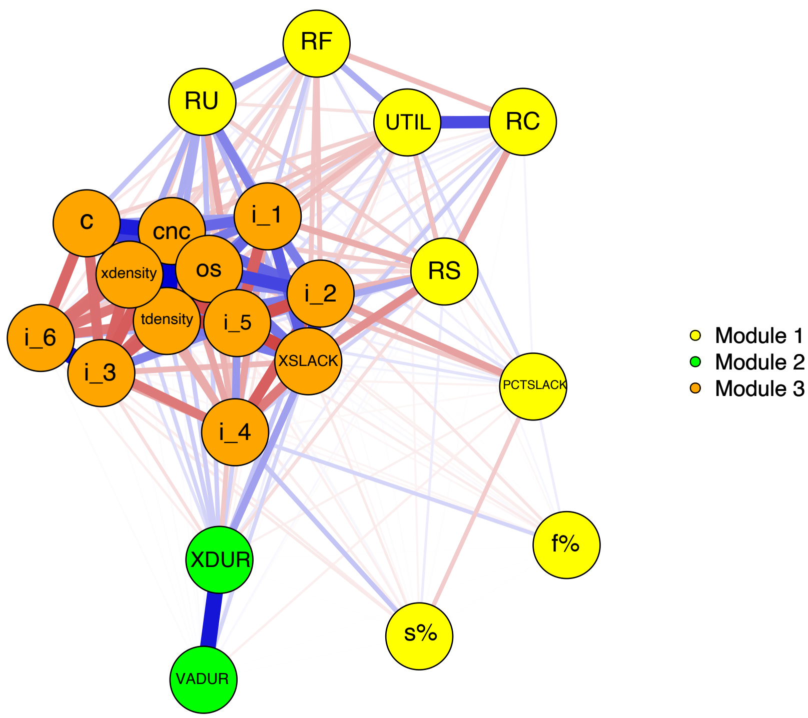


Figure A.18: Clustered correlation graph of the multiple project database indicators. (Note: Indicators indicate the average values for a project.

Appendix B. Formal description of employed indicators

Appendix B.1. Structural complexity

We denote $\mathcal{S}$ a realized project structure, $\mathbf{LD} \in \{0,1\}^{n\times n}$ of $\mathcal{S}$, $|\mathcal{A}| = \sum_{i\neq j} l_{ij}$ ($l_{ij} = [\mathbf{LD}]_{ij}$) is the total number of dependencies (arcs) between tasks.

I_1, the number of tasks (nodes), is calculated as follows:

$$I_1 := n \tag{B.1}$$

I_2, the serial-parallel structure, measures the closeness to a serial or parallel completion. For I_2, we need the following notations: S_i (P_i) denotes the *set* of immediate successors (predecessors) of task i. For topologically ordered, acyclic project networks, $|S_i| = \sum\limits_{j=i+1}^{n} l_{ij}$, $|P_i| = \sum\limits_{j=1}^{i-1} l_{ji}$. The progressive ($PL_i$) and regressive ($RL_i$) level *numbers* of each task i can be calculated as follows:

$$PL_i := \begin{cases} 1 & \text{if } P_i = \emptyset \\ \max\limits_{j\in P_i} PL_j + 1 & \text{if } P_i \neq \emptyset \end{cases} \tag{B.2}$$

and

$$RL_i := \begin{cases} m & \text{if } S_i = \emptyset \\ \min\limits_{j\in S_i} RL_j - 1 & \text{if } S_i \neq \emptyset \end{cases} \tag{B.3}$$

where $m = \max\limits_{i} PL_i$. Next, we have the following:

$$I_2 := \begin{cases} 1 & \text{if } n = 1 \\ \frac{m-1}{n-1} & \text{if } n > 1 \end{cases} \tag{B.4}$$

I_3, the task distribution, measures the distribution of tasks over the progressive levels by calculating the total absolute deviations.

First, we define the jth progressive *level* of $j = 1, ..., m$ as follows: $\mathbf{PL}_j := \{i \leq n : PL_i = j\}$, i.e., the *set* of all tasks having progressive level number j. Then,

$$I_3 := \begin{cases} 0 & \text{if } m = 1 \text{ or } m = n \\ \frac{\alpha_w}{\alpha_{\max}} = \frac{\sum\limits_{j=1}^{m} |w_j - \overline{w}|}{2(m-1)(\overline{w}-1)} & \text{if } 1 < m < n \end{cases} \tag{B.5}$$

where $w_j = |\mathbf{PL}_j|$ is the width (size) of progressive level $j = 1, ..., m$, $w = (w_1, w_2, ..., w_m)$ is the vector containing the widths of each progressive level, and $\overline{w} = n/m$, α_w is the total absolute deviation from the average width. Then, $\alpha_{\max}$ is the maximal value of α_w of a network (ranging for all possible $\mathcal{A}$); thus,[2)] $\alpha_{\max} = (m-1)(\overline{w}-1) + (n-m+1-\overline{w}) = 2(m-1)(\overline{w}-1)$.

I_4, the ratio of *short* arcs. The length of an "arc" (called a path in graph theory) between tasks i_1 and i_2 is defined as $L(i_1, i_2) := |PL_{i_1} - PL_{i_2}|$, the difference between their progressive level numbers. Arcs of length 1 are called *short*, and $D := \sum_{j=1}^{m-1} w_j \cdot w_{j+1}$ is the *maximal* number of short arcs. n'_L denotes the number of arcs of length L for $1 \leq L \leq m-1$. Then, I_4 is calculated as follows:

$$I_4 := \begin{cases} 1 & \text{if } D = n - w_1 \\ \frac{n'_1 - n + w_1}{D - n + w_1} & \text{if } D > n - w_1 \end{cases} \tag{B.6}$$

I_5, the ratio of *long* arcs ($L > 1$), is calculated as follows:

$$I_5 := \begin{cases} 1 & \text{if } |\mathcal{A}| = n - w_1 \\ \frac{\left(\sum\limits_{L=2}^{m-1} n'_L \frac{m-L-1}{m-2}\right) + n'_1 - n + w_1}{|\mathcal{A}| - n + w_1} & \text{if } |\mathcal{A}| > n - w_1 \end{cases} \tag{B.7}$$

I_6, the topological float, considers the differences between the regressive and progressive level numbers of task i, i.e., $|RL_i - PL_i|$, as follows:

$$I_6 := \begin{cases} 0 & \text{if } m \in \{1, n\} \\ \frac{\sum\limits_{i=1}^{n} |RL_i - PL_i|}{(m-1)(n-m)} & \text{if } m \notin \{1, n\} \end{cases} \tag{B.8}$$

CNC, the coefficient of network complexity, is calculated as follows:

$$\text{CNC} = \frac{|\mathcal{A}|}{n} \tag{B.9}$$

OS, the order strength, is calculated as follows:

$$\text{OS} = \frac{|\mathcal{A}|}{n(n-1)/2} \tag{B.10}$$

2) The maximal value of α_w is achieved (n and m are fixed, $\sum_{j=1}^{m} w_j = n$) when all levels are singletons, except for one with $n-(m-1)$ tasks; repetitive use of the inequality $|a-\overline{w}| + |b-\overline{w}| < |a-1-\overline{w}| + |b+1-\overline{w}|$ for $1 < a \leq \overline{w} \leq b < n$ proves this extrema.

C, the network complexity, is calculated as follows:

$$\mathrm{C} = \begin{cases} \dfrac{\log \frac{|\mathcal{A}|}{n-1}}{\log \frac{n^2-1}{4(n-1)}} & \text{if } n \text{ is odd} \\ \\ \dfrac{\log \frac{|\mathcal{A}|}{n-1}}{\log \frac{n^2}{4(n-1)}} & \text{if } n \text{ is even} \end{cases} \tag{B.11}$$

T-DENSITY, the total activity density, is calculated as follows:

$$\text{T-DENSITY} := \sum_{i:=1}^{n} \max\{0, |P_i| - |S_i|\} \tag{B.12}$$

(S_i and P_i were defined immediately before I_2.)

XDENSITY, the average activity density, is calculated as follows:

$$\text{XDENSITY} := \frac{\text{T-DENSITY}}{n} \tag{B.13}$$

Flexibility-related structural indicators. All structural indicators depend on the realized structure ($\mathcal{S}$), i.e., on the set of the included flexible dependencies and supplementary tasks from $\mathbf{LD'} \in [0,1]^{n \times n}$. I_1 = number of tasks; therefore, $I_1(\mathcal{S}_{\text{min}}) = I_1(\mathcal{S}_{\text{minimax}}) \leq I_1(\mathcal{S}) \leq I_1(\mathcal{S}_{\text{max}}) = I_1(\mathcal{S}_{\text{maximin}})$. Nevertheless, since the fixed dependencies between the supplementary tasks must be excluded if the supplementary tasks are excluded, the minimal (maximal) structures are the lower (upper) bounds of C. The CNC and OS indicators of these cases are those in which only mandatory tasks exist. Regarding the other structural indicators, the connection between them and the maximal-minimal structures are not obvious, and no such rules can be defined.

Appendix B.2. Time-related indicators

To ensure the validity of the comparison of the simulated and real-life datasets, only networks with single modes are considered. Therefore, only the single-mode version of the indicators was considered. We denote $\mathcal{S}$ a realized project structure that decides the non-mandatory tasks and dependencies *from* $\mathbf{LD'} \in [0,1]^{n \times n}$. In the following, all quantities depend on $\mathcal{S}$, but we omit indicating $\mathcal{S}$ everywhere. For example, $\mathcal{S}$ determines $\mathbf{LD''} \in \{0,1\}^{n'' \times n''}$ from $\mathbf{LD'} \in [0,1]^{n \times n}$, However, we simply denote $\mathbf{LD''}$ and n'' by $\mathbf{LD}$ and n, similarly for $\mathbf{TD}$, and $|\mathcal{A}| = \sum_{i<j} l_{ij}$ ($l_{ij} = [\mathbf{LD}]_{ij}$). We denote $t_i := [\mathbf{TD}]_{ii}$ the duration of task i and $\overrightarrow{P} =$ "$a_1 \prec a_2 \prec \ldots \prec a_N$"

a path of preceding tasks, where $a_j \prec a_{j+1}$ indicates $l_{a_j,a_{j+1}} = 1$ for $1 \leq j < N$ ($N \leq n$). $\ell(\overrightarrow{P}) := N$ is the *length* of the path, and $d(\overrightarrow{P}) := \sum_{i \in \overrightarrow{P}} t_i$ is the *duration* of path $\overrightarrow{P}$. A path $\overrightarrow{L}$ is called the *longest* or *critical* path if $d(\overrightarrow{L})$ is maximal among all paths. Next, the TPT, the total project time, is calculated as follows:

$$\text{TPT} := d(\overrightarrow{L}) \tag{B.14}$$

for any longest path $\overrightarrow{L}$. $\overline{X}$DUR, the average task duration, is calculated as follows:

$$\overline{X}\text{DUR} := \frac{1}{n}\sum_{i=1}^{n} t_i \tag{B.15}$$

VA-DUR, the variance in task duration, is calculated as follows:

$$\text{VA-DUR} := \frac{1}{n-1}\sum_{i=1}^{n} \left(t_i - \overline{X}\text{DUR}\right)^2 \tag{B.16}$$

PCTSLACK, the percent of tasks with positive total slack, is calculated as follows:

$$\text{PCTSLACK} := \frac{1}{n}\sum_{i=1}^{n} \begin{cases} 1 & \text{if } LS_i - ES_i > 0 \\ 0 & \text{if } LS_i - ES_i = 0 \end{cases} \tag{B.17}$$

where LS_i (ES_i) is the latest (earliest) start time, and $TS_i := LS_i - ES_i$ is the total slack of task i.

XSLACK, the average total slack per task, is calculated as follows:

$$\text{XSLACK} := \frac{1}{n}\sum_{i=1}^{n} TS_i \tag{B.18}$$

TOTSLACK-R, the total slack ratio, is calculated as follows:

$$\text{TOTSLACK-R} := \frac{\sum_{i=1}^{n} TS_i}{TPT} \tag{B.19}$$

XSLACK-R, the average slack ratio, is calculated as follows:

$$\text{XSLACK-R} := \frac{XSLACK}{TPT} \tag{B.20}$$

PCTFREESLK is the percent of tasks with positive free slack. First, the earliest finishing time of task j is $EF_j = ES_j + t_j$; then, we denote $FS_i := \min_{l_{ij}=1} ES_j - EF_i$ the free slack of task i (lowest early start of successors - early finish). Here, we have the following:

$$\text{PCTFREESLK} := \frac{1}{n} \sum_{i=1}^{n} \begin{cases} 1 & \text{if } FS_i > 0 \\ 0 & \text{if } FS_i = 0 \end{cases} \tag{B.21}$$

XFREESLK, the average free slack per task, is calculated as follows:

$$\text{XFREESLK} := \frac{1}{n} \sum_{i=1}^{n} FS_i \tag{B.22}$$

Flexibility impacts of the time-related indicators. Since the average task duration and variance in activity duration depend on the inclusion/exclusion of tasks but not on their dependencies (see (B.16) and (B.15)), the following equations are easy to verify:

$$\overline{X}\text{DUR}(\mathcal{S}_{\max}) = \overline{X}\text{DUR}(\mathcal{S}_{\text{maximin}}) \tag{B.23}$$

$$\overline{X}\text{DUR}(\mathcal{S}_{\min}) = \overline{X}\text{DUR}(\mathcal{S}_{\text{minimax}}) \tag{B.24}$$

$$\text{VA-DUR}(\mathcal{S}_{\max}) = \text{VA-DUR}(\mathcal{S}_{\text{maximin}}) \tag{B.25}$$

$$\text{VA-DUR}(\mathcal{S}_{\min}) = \text{VA-DUR}(\mathcal{S}_{\text{minimax}}) \tag{B.26}$$

Large samples. Large samples refer to large n for which we can use the central limit theorem (CLT). Here, we offer some mathematical results regarding $\overline{X}\text{DUR}(\mathcal{S})$. Similar results are also used for resource indicators, such as RF, PCTR, RU, DMND, and RC in Equation (B.38).

$\overline{X}DUR(\mathcal{S})$ contains (finally) mandatory tasks only; thus, we may consider $\mathcal{S} \subseteq \mathbb{I}_n$, where we denote $\mathbb{I}_n := \{1, 2, ..., n\}$, and we let $s = |\mathcal{S}|$.

In the following, we assume that n and s are large numbers, $t_i \sim U(a, b)$ (for $i \in \mathbb{I}_n$) are uniform random variables (**r.v.**) on the *fixed* finite interval $[a, b] \subset \mathbb{R}$, and t_i are independent and identically distributed (**i.i.d.**) r.v.

STEP ONE: n and $\mathcal{S}$ are fixed. Next, $\overline{X}DUR(\mathcal{S})$ is the mean of s i.i.d. uniform r.v., and thus, the CLT yields the following:

$$\frac{\overline{X}DUR(\mathcal{S}) - \mu}{\frac{\sigma}{\sqrt{s}}} \sim \Phi(0, 1) \tag{B.27}$$

where:

$$\mu = E\left(\overline{X}DUR(\mathcal{S})\right) = \frac{a+b}{2} \quad , \quad \sigma = D(\overline{X}DUR(\mathcal{S})) = \frac{|b-a|}{\sqrt{12}} \tag{B.28}$$

and $\Phi(0,1)$ is the standard normal distribution[3)].

STEP TWO: n is fixed, but $\mathcal{S}$ may be any nonempty subset of $\mathbb{I}_n$, i.e., the event space is currently the power set of $\mathbb{I}_n: \quad \Omega = \mathcal{P}(\mathbb{I}_n)$. Next, we consider $\overline{X}DUR(\mathcal{S})$ on Ω and use the notation $\overline{\mathcal{X}}_{DUR}$ instead of $\overline{X}DUR(\mathcal{S})$. The probability of *any* $\mathcal{S}$ is $\frac{1}{n!}$, $E(\overline{\mathcal{X}}_{DUR}[\mathcal{S}]) = \mu$ and $D\left(\overline{\mathcal{X}}_{DUR}[\mathcal{S}]\right) = \frac{\sigma}{\sqrt{s}}$ when $s = |\mathcal{S}|$, which has the probability $\binom{n}{s}/2^n$; thus, we have the following:

$$D\left(\overline{\mathcal{X}}_{DUR}\right) = \sqrt{\frac{1}{2^n}\sum_{s=1}^{n}\binom{n}{s}\left(\frac{\sigma}{\sqrt{s}}\right)^2} = \sigma\sqrt{\frac{1}{2^n}\sum_{s=1}^{n}\frac{\binom{n}{s}}{s}} \tag{B.29}$$

Finally, by the CLT, we obtain the following:

$$\frac{\overline{X}DUR(\mathcal{S}) - \mu}{D\left(\overline{\mathcal{X}}_{DUR}\right)} \sim \Phi\left(0,1\right) \tag{B.30}$$

In the case $|\mathcal{S}|$ is limited, i.e., $c \leq |\mathcal{S}| \leq d$ is required for some fixed $c \leq d \leq n$, (B.29) becomes the following:

$$D\left(\overline{\mathcal{X}}_{DUR}\right) = \sigma\sqrt{\frac{1}{2^n}\sum_{s=c}^{d}\frac{\binom{n}{s}}{s}} \tag{B.31}$$

Appendix B.3. Resource-related indicators

We denote $\mathcal{S}$ a realized project structure and $\mathbf{LD} \in \{0,1\}^{n\times n}$, $\mathbf{T} \in \mathbb{R}_+^n$ and $\mathbf{RD} \in \mathbb{R}_+^{n\times\rho}$ domains of the matrix representation of $\mathcal{S}$, where n is the number of tasks and $|\mathcal{A}| = \sum_{ij,i\neq j} l_{ij}$ ($l_{ij} = [\mathbf{LD}]_{ij}$) We denote $t_i = [\mathbf{TD}]_{ii}$ the duration of task i and TPT the duration of the project, and $r_{ij} = [\mathbf{RD}]_{ij}$ the resource demand of task i of resource j.

$\overrightarrow{\mathcal{S}}$ is a project *schedule* of project structure $\mathcal{S}$ if for each realized task $a_i \in \mathcal{S}$, the interval $T_i \subseteq [0, \mathrm{TPT}]$ is determined when a_i is addressed (scheduled). To ensure compatibility with other papers, we use the redundant notation $a_i(T) \in \overrightarrow{\mathcal{S}}$.

3). In the denominator of (B.27), one may write $\sqrt{\text{VA-DUR}\,(\mathcal{S})}$ instead of $\frac{\sigma}{\sqrt{s}}$.

We denote $S(a_i(T)) \in [0, TPT - t_i]$ the start and $F(a_i(T)) \in [t_i, TPT]$ the finish time of task i. The early schedule, denoted as $\overrightarrow{\mathcal{S}}_{\min}$, satisfies $\forall a_i(T) \in \overrightarrow{\mathcal{S}}_{\min}\, S(a_i(T)) = \mathrm{ES}_i$ and $F(a_i(T)) = \mathrm{EF}_i$. We denote the *resource demand* j of task i at time τ as follows:

$$r_{ij}(\tau) := \begin{cases} r_{ij} \text{ if } a_i(T) \in \overrightarrow{\mathcal{S}}, \tau \in T_i \\ 0 \text{ otherwise} \end{cases} \tag{B.32}$$

Furthermore, we denote the total (renewable) resource *demand* of j at time τ as $r_j(\tau) = \sum_i r_{ij}(\tau), \quad \tau \in [0, \mathrm{TPT}]$.

Appendix B.3.1. Nonscheduled

RF, the resource factor, is the density of **RD**, the resource matrix from a domain mapping matrix (DMM). RF gives the rate of how often resources required are from all possible resource type-activity pairings. Higher RF values indicate a more complex scheduling problem.

$$\mathrm{RF} := \frac{1}{n\rho} \sum_{i=1}^{n} \sum_{j=1}^{\rho} \begin{cases} 1 & \text{if } r_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} = \frac{1}{\rho} \sum_{j=1}^{\rho} \mathrm{PCTR}_j \tag{B.33}$$

where r_{ik} denotes the amount of resource type j required by task i, and PCTR_j denotes the percent of activities that require the given resource type, which gives a columnwise view of RF as follows:

$$\mathrm{PCTR}_j := \frac{1}{n} \sum_{i:=1}^{n} \begin{cases} 1 & \text{if } r_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{B.34}$$

RU, the resource use, represents the resource use for each activity, i.e., the number of resource types used. RU varies between 0 and r (the number of resource types). It is a rowwise view of RF ($i = 1, ..., n$) as follows:

$$\mathrm{RU}_i := \sum_{j=1}^{\rho} \begin{cases} 1 & \text{if } r_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{B.35}$$

DMND_j is the average quantity of resource j demanded when required by an activity ($j = 1, ..., \rho$) as follows:

$$\mathrm{DMND}_j := \frac{\sum_{i=1}^{n} r_{ij}}{\sum_{i=1}^{n} \begin{cases} 1 & \text{if } r_{ij} > 0 \\ 0 & \text{if } r_{ij} = 0 \end{cases}} \tag{B.36}$$

RC is the resource constrainedness of each resource type and is calculated as follows:

$$\mathrm{RC}_j := \frac{\mathrm{DMND}_j}{\alpha_j} \tag{B.37}$$

where α_j is the *availability* of renewable resource type j.

Flexibility impacts on the nonscheduled renewable resource indicators. The nonscheduled resource-related indicators are independent of the schedule. Therefore, they are independent of the rate of flexible dependencies.

All possible structures can be considered a random sample from the maximal structure if the elements of S-SET follow a uniform distribution. In this case, the following formula can be specified:

$$\frac{\mathrm{NRI}(\mathcal{S}) - Exp(\mathrm{NRI}(\mathcal{S}))}{\sqrt{Var(\mathrm{NRI}(\mathcal{S}))}} \sim \Phi(0,1) \tag{B.38}$$

where $\mathrm{NRI}(\mathcal{S})$ denotes any mean of the nonscheduled resource indicators, such as RF, PCTR, RU, DMND, and RC for project structure $\mathcal{S}$.

Appendix B.3.2. Resource-related indicators for the early schedule

The following indicators from Patterson (1976) require early scheduling ($\overrightarrow{\mathcal{S}}_{\min}$) of the activities regarding the precedence relations but not the resource constraints.

RS is the resource strength of each renewable resource type and is calculated as follows:

$$\mathrm{RS}_j := \frac{\alpha_j - r_j^{\min}}{r_j^{\max} - r_j^{\min}} \tag{B.39}$$

where α_j denotes the total availability of renewable resource type j, $r_j^{\min} := \max_{i=1,\dots,n}(r_{ij})$ is the highest *individual* resource demand, and $r_j^{\max}$ denotes the peak total demand at any moment for resource type j in the precedence preserving the earliest start schedule.

UTIL_j is the utilization (rate) of resources and is measured based on the critical path length. Higher values indicate more constraints, less room for scheduling, and less possibility of changing the task starting times without increasing the TPT.

$$\mathrm{UTIL}_j := \frac{\sum_{i=1}^{n} r_{ij} t_i}{\alpha_j \cdot \mathrm{TPT}} \tag{B.40}$$

TCON_j is the constrainedness of (renewable) resource type j over time. In practice, it is the average utilization (UTIL_j) considering only those tasks that use that particular resource type as follows:

$$\mathrm{TCON}_j := \frac{\sum_{i=1}^{n} r_{ij} t_i}{\alpha_j \cdot \mathrm{TPT} \cdot \sum_{i=1}^{n} \begin{cases} 1 & \text{if } r_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}} \tag{B.41}$$

OFACT_j is the obstruction factor of (renewable) resource type j and is calculated as follows:

$$\mathrm{OFACT}_j := \frac{\int_0^{\mathrm{TPT}} \max\{0; r_j(\tau) - \alpha_j\} d\tau}{\sum_{i=1}^{n} r_{ij} t_i} \tag{B.42}$$

UFACT_j is the underutilization factor and is calculated as follows:

$$\mathrm{UFACT}_j := \frac{\int_0^{\mathrm{TPT}} \max\{0; \alpha_j - r_j(\tau)\} d\tau}{\sum_{i=1}^{n} r_{ij} t_i} \tag{B.43}$$

Interval of the scheduled resource indicators. Since the minimax (maximin) structure requires minimal (maximal) resource demands, the following equations can be specified.

$$\mathrm{SRI}_j(\mathcal{S}_{\text{minimax}}) \leq \mathrm{SRI}_j(\mathcal{S}) \leq \mathrm{SRI}_j(\mathcal{S}_{\text{maximin}}) \tag{B.44}$$

$$\mathrm{SRI}(\mathcal{S}_{\text{minimax}}) \leq \mathrm{SRI}(\mathcal{S}) \leq \mathrm{SRI}(\mathcal{S}_{\text{maximin}}) \tag{B.45}$$

where SRI_j denotes the scheduled resource indicators, such as RS, UTIL, TCON, OFACT, and UFACT, of resource j, and SNI denotes the mean of a scheduled resource indicator of all resource types.

Resource indicator mean. Since the number of resource demands is very heterogeneous, in this study, the mean of the resource indicators was considered instead of calculating these values of all resources. Moreover, to ensure the comparability of the resource indicators, when the resource numbers differ across projects, we must use the means of these indicators. In Subsection Appendix B.2, we also focus on the means. In the following, the means are denoted without indexing.