Journal of Computational Science 93 (2026) 102745

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Exploring network-level indicators for project analysis — A comparison of real and synthetic projects

Zsolt T. Kosztyán

University of Pannonia, Department of Quantitative Methods, Veszprém, H-8200 Egyetem str. 10, Hungary
Institute of Advanced Studies, Kőszeg, H-9730 Chernel str. 14, Hungary

ARTICLE INFO

Keywords:
Network indicators
Project database
Synthetic projects
Real project structures

ABSTRACT

This study introduces a novel perspective on project network analysis by incorporating network theory and graph analysis to identify network-level indicators for activity-on-node project networks. A key contribution lies in the comparison between real and synthetic projects on the basis of project and network-level indicators, revealing distinct variations. The findings underscore that real projects demonstrate lower flexibility and efficiency than synthetic counterparts, impacting project scheduling and resource allocation. Moreover, the study suggests employing supervised and unsupervised learning classification methods to categorize projects on the basis of indicator values, enhancing project selection and prioritization processes. By bridging the gap between network-level indicators and project aspects, this research enriches the literature by offering fresh insights and resources for project managers to optimize project network structure and performance.

1. Introduction

The significance of projects is crucial and includes several aspects, such as corporate operations, research, and national economies [1]. Within the project management, the project planning problem is a critical challenge in management science and operations research, focusing on efficient scheduling and resource allocation of project activities [2]. This complex task involves organizing interconnected activities, managing resources, and optimizing project timelines to achieve the desired outcomes. The significance of project planning extends across various domains, including corporate operations, research initiatives, and national economies. Researchers in Management Science and Operations Research have thoroughly examined project scheduling and resource allocation for more than sixty years [3]. Project databases were suggested as a means to evaluate algorithms, encompassing both simulated [4] and real scenarios [5], as well as individual [6] or multiple projects [7], and single-mode [8] or multimode projects [9]. Nevertheless, the numerous project databases exhibit significant heterogeneity in terms of both the file structure and the characteristics of the represented projects. Kosztyán et al. [10] proposed a unified matrix-based project planning (UMP) model, which offers several advantages over existing network-based project planning methods and heterogeneous network-based sources of project plans. UMP provides a unified approach that can seamlessly represent both individual projects and multiprojects portfolios. It is scalable in terms of complexity and is capable of handling both single-mode and multimode project completions. Moreover, UMP's power lies in its ability to collect and integrate most existing projects databases, allowing for comprehensive analysis and comparison across diverse project structures and datasets. This unified framework enables researchers and project managers to conduct more robust and standardized analyses, facilitating better understanding and management of complex project networks. Kosztyán [11] proposed a MATLAB model, and Kosztyán and Saghir [12] proposed an R package that can plan different kinds of project scheduling problems. Kosztyán and Saghir [12] stores [4]'s synthetic and Batselier and Vanhoucke [5]'s real-life projects. The logic structure of the project is stored in a logic domain, which is an adjacency matrix of the activity-on-node network of the project plan. Further domains (i.e., submatrices) store the task duration and resource demands by completion mode and resource type.

Owing to the UMP, different types of project databases can be compared to each other. In addition, project networks can be analyzed by network-level indicators. In this paper, we present the following research questions.

- **RQ**$_1$ What are the new insights from the proposed network-level indicators for project planning and scheduling?
- **RQ**$_2$ What is the relationship between the proposed network level and the existing project indicators?
- **RQ**$_3$ What is the difference between real and synthetic project plans in the case of the project and network-level indicators?

E-mail address: kosztyan.zsolt@gtk.uni-pannon.hu.

https://doi.org/10.1016/j.jocs.2025.102745
Received 16 December 2024; Received in revised form 13 May 2025; Accepted 14 November 2025
Available online 26 November 2025

The investigation of $\mathbf{RQ}_1$ can facilitate the identification of the prospective advantages and difficulties associated with the utilization of network-level indicators for project planning and scheduling. Network-level indicators refer to metrics that quantify the qualities and attributes of the project network; these metrics include density, diameter, modularity, centrality, resilience, efficiency, and transitivity. Moreover, these metrics offer novel perspectives on the composition and intricacy of the project network, as well as their impact on project performance, risk, and quality. The examination of $\mathbf{RQ}_2$ can provide insights into the relationships in the network-level indicators and the current project metrics. The existing project indicators encompass many measurements that reflect key features of project management, including duration, resource allocation, and network complexity, order strength, activity distribution, and topological float. These indicators can offer insights into the scope, timing, cost, and logic of the project. An examination of the correlation between network-level indicators and current project indicators, the impact of network properties on project elements can be understood along with the reciprocal influence. Additionally, methods for integrating and for These factors can be aligned to enhance project management. The use of $\mathbf{RQ}_3$ allows direct comparison and analysis of the project and network-level indicators between real and synthetic project plans. Real project plans are derived from factual data obtained from companies in various industries, whereas synthetic project plans are produced by random network generators. The research goal is to assess the accuracy and usefulness of the simulation model by comparing the actual project plans with synthetic plans. Additionally, we seek to uncover the commonalities and disparities between the real and synthetic project networks.

The research questions were addressed as follows. The background of the study, and the contributions of our study to the literature are as follows: Section 2. Methods and application materials are as follows: introduced in Section 3. In Section 4, We report the results of the study. In Section 5, we discuss the results of the study. Section 6 provides the summary and conclusions, and finally, Section 7 shows the limitations and potential further directions.

2. Background of the study

A multitude of indicators are presented in the literature, as there are continuous endeavors to create novel indicators. Vanhoucke provides a comprehensive summary in Vanhoucke et al. [13], and Browning and Yassine [14] offer a comprehensive analysis of different projects. Van Eynde and Vanhoucke [15] and Van Eynde and Vanhoucke [16] have expanded their research to incorporate indicators for both projects generation and classification.

The existing indicators can be separated into three large groups: (1) *time-related*, (2) *resource-related* and (3) *structural* (i.e.,. *logic*) indicators (see the summary in Van Eynde et al. [17]). The *time-related indicators* measure the duration, schedule, and flexibility of the project activities and the project as a whole. These measures include variables such as average activity duration, critical path length, number and percentage of activities with positive slack, average slack and the slack ratio, and variance in activity duration. They are important because they help to monitor and control project progress, performance, and quality. (2) The *resource-related indicators* measure the allocation, utilization, and diversity of project resources, such as human, material, or financial resources. These variables include the number of renewable resources, alpha distribution of resource strength, the normalized average resource loading factor, and the resource leveling index. Such measures are important because they help optimize project cost, efficiency, and reliability. Finally, (3) *structural (logical) indicators* measure the complexity, order, and density of the project network, which represents the activities and their dependencies in a project. These structural indicators include variables such as the network complexity coefficient, the order strength coefficient, the activity distribution index, the serial/parallel index, the length of the long arcs index, the short arcs index, the topological float index, the total and average activity density, the number of articulation points, and the number of components. These variables are important because they help to understand the structure and logic of the project network while improving project planning and scheduling.

Since structural indicators already incorporate the basic properties of the project network, several network-level indicators are not interpreted in the project network and are not used to describe the logic structure of the project network.

A project network is also a network that is a set of nodes and a set of edges that connect some pairs of nodes. In the case of an Activity-on-Node (AoN) project network, the nodes represent tasks (activities), and the edges represent dependencies. A dependency is a logical relationship that specifies the order or sequence of the tasks. An AoN project network is a directed network in which the edges have a direction that indicates the precedence of the tasks. Such a network is usually a directed acyclic graph (DAG), where there are no cycles or loops in the network; however, our analysis does not require this property.

Project indicators are crucial for delineating project frameworks, for assessing structural intricacy, for estimating length, for identifying idle time, and for determining resource requirements. This information has the potential to significantly impact the performance of the scheduling or resource allocation process [17]. Nevertheless, while these indicators are focused on time and resource demands, and there are several indicators that focus on the project structure of the project, to our knowledge, most network-level indicators, such as centralization, modularity, and transitivity are not considered. However, these indicators may provide new insights for the project managers.

The main difference between the proposed network-level and structural indicators is that the network-level indicators are on the basis of network theory and graph analysis, while the structural indicators are on the basis of project management theory and logic analysis. This means that network-level indicators offer a more holistic and dynamic perspective on the project network, whereas the structural indicators offer a greater Detailed and static perspective on the project network. Another The difference between the network-level and structural indicators is that the network-level indicators capture the emergent and nonlinear properties of the project network; these properties include modularity, centrality, resilience, efficiency, and transitivity. The structural indicators capture the deterministic and linear properties of the project network, such as complexity, order, and density. This means that the network-level indicators can reveal the hidden and unexpected patterns and behaviors of the project network, whereas the structural indicators can reveal the explicit and predictable rules and constraints of the project network. For the structural indicators, it is assumed that the project network is homogeneous, independent, and static; this assumption may not reflect the reality of a complex and dynamic project environment. Network-level indicators can address these limitations by incorporating heterogeneity, interactivity, and adaptability of the project network, which can lead to more realistic and robust project network analysis.

To compare different kinds of project databases, we used the UMP model. However, because of the focus on the project indicators, we consider only logic (LD), time (TD) and (renewable) resource domains (RD), which store the AoN project network, the time and the resource demands.

This study makes three contributions to the literature.

- $\mathbf{C}_1$ We interpret network-level indicators to explain the logical structure of an activity-on-node network, and we demonstrate how these indications can be understood within the framework of a project.
- $\mathbf{C}_2$ We show the effects of the time-related, the resource-related, and the structural project metrics on the suggested network-level indicators.

C$_3$ We show that synthetic and real projects exhibit distinct variations in their project and network indicator values. These variables can therefore be employed to categorize projects with supervised and unsupervised learning classification techniques.

According to the contribution of **C$_1$**, we propose a new perspective on project network analysis that is on the basis of network theory and graph analysis. Network-level indicators are measures that capture the properties and characteristics of the project network, such as density, diameter, modularity, centrality, resilience, efficiency, and transitivity. These indicators can provide novel insights into how connected, diverse, modular, centralized, resilient, efficient, and cohesive the project network is, and how these factors affect project performance, risk, and quality. Interpreting network-level indicators in context of an activity-on-node network, we show how these indicators can be applied and understood within the framework of a project. Contribution **C$_2$** regards the relationship between the network-level indicators and the existing project indicators. Existing project indicators are measures that capture the aspects of the project time, resources, and logic. These indicators can provide information about the project scope, schedule, and logic. By showing the effects between the network-level indicators and the existing project indicators, we reveal how the network properties influence and are influenced by the project aspects and how they can be integrated and aligned for better project management. Finally, according to contribution **C$_3$**, we compare and contrast the real and synthetic project plans in terms of the project and network-level indicators. Real project plans are on the basis of actual data, e.g., from a construction or a service-oriented company, while synthetic project plans are generated by a random network generator. By demonstrating that synthetic and real projects exhibit distinct variations in their project and network indicators values, we evaluate the validity and the applicability of the simulation model. Moreover, we identify the similarities and differences between the real and synthetic project networks. We also demonstrate how these variables can be used to categorize projects with supervised and unsupervised learning classification techniques, which can improve the project selection and prioritization processes. This classification step highlights what are the most different indicators between a real-life and synthetic projects.

To achieve maximal reproducibility, this study used the R programming language. The **rticles** [18] package was utilized to generate the LaTeX code and the corresponding PDF document. All the data and tables are derived from computations. The entire code is provided alongside the publication to enable anyone to replicate the findings. The following R packages are used in the paper: To read the descriptive files of the indicators we used the **readxl** package [19]. For the data preparation and manipulation, we used the **xfun** [20], **stringr** [21] packages. For descriptive statistics, we used the **DescTools** package [22] and for analysis of variance (ANOVA), we applied the **car** package [23]. For the cluster analysis and unsupervised learning, we used the package **nda** [24] and to calculate the multidimensional scaling, we used the **MASS** package [25]. To measure the average of the silhouette measure, we used **cluster** package [26]. In the case of supervised learning, we used **Boruta** [27] to calculate relevant indicators, and we used the **randomForest** package [28] to calculate variable importance. To calculate accuracy measures on the basis of the confusion matrix, we used **caret** package [29]. To construct the data tables, we used the **knitr** package [30] and the **kableExtra** package [31]. To manipulate the figures, we used the **scales** packages [32], and to present figures, we used the **ggplot2** [33], **ggrepel** [34], and **plotly** [35] packages. For the remaining functions, we used **base** [36] and **stats** [37] packages. Matrix-based project plans for network analysis were produced by the **mfpp** [12] package and analyzed by the **igraph** [38] and **brainGraph** [39] packages.

3. Material and methods

In this section we introduce the applied datasets (see Section 3.1) and we provide the applied indicators (see Section 3.2) and methods (see Section 3.3).

Table 1
Applied matrix-based model.

UMP	LD				TD			RD					
	a_1	a_2	$\cdots$	a_n	$\mathbf{t}_1$	$\cdots$	$\mathbf{t}_m$	$\mathbf{r}_{11}$	$\cdots$	$\mathbf{r}_{1\rho}$	$\mathbf{r}_{21}$	$\cdots$	$\mathbf{r}_{m\rho}$
a_1	1	l_{12}	$\cdots$	l_{1n}	t_{11}	$\cdots$	t_{1m}	r_{111}	$\cdots$	$r_{1\rho1}$	r_{211}	$\cdots$	$r_{m\rho1}$
a_2	0	1	$\cdots$	l_{2n}	t_{21}	$\cdots$	t_{2m}	r_{112}	$\cdots$	$r_{1\rho2}$	r_{212}	$\cdots$	$r_{m\rho2}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
a_n	0	0	$\cdots$	1	t_{n1}	$\cdots$	t_{nm}	r_{11n}	$\cdots$	$r_{1\rho n}$	r_{21n}	$\cdots$	$r_{m\rho n}$

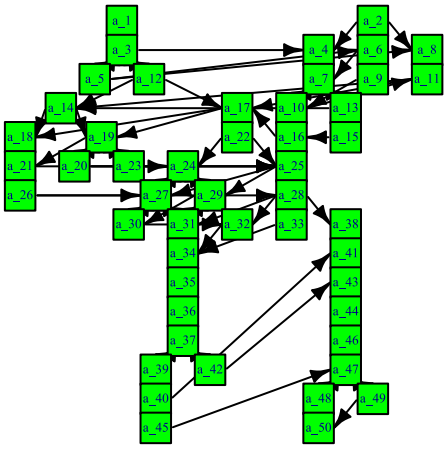
3.1. Employed datasets

Kosztyán et al. [10] proposed a unified matrix-based project planning (UMP) model to collect heterogeneous project datasets, which are used in the resource-constrained project scheduling problem. The UMP is a multidomain mapping matrix (MDM) model [40], which considers two mandatory (logic and time) and four supplementary (cost, quality, renewable resources and nonrenewable resources) domains. Existing indicators focus on durations, resource demands and logic structures; Therefore, only three domains are employed, such as logic domain (LD), time domain (TD) and the renewable resource domain (RD). The Table 1 shows the realized model of the UMP. We used Boctor [4]'s datasets, in which there were 240 original instances. This is a synthetic project dataset, and all project plans contain four possible completion modes. Batselier and Vanhoucke [5]'s datasets are on the basis of real-life Projects, which contains 133 real project networks with only one completion mode. Despite Batselier and Vanhoucke [5]'s dataset including task costs, synthetic project networks usually neglect costs or consider costs as a nonrenewable resource. Since Boctor [4]'s dataset did not specify nonrenewable resources and since the existing project indicators focus only on time-related, resource-related and structural indicators, we used only three domains for the UMP; see Table 1.

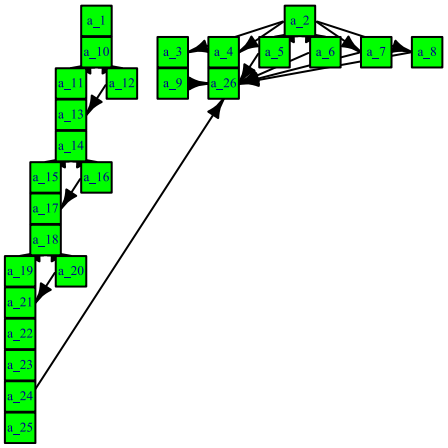
LD is a mandatory domain that specifies an individual logic structure (i.e., single) or, in a multiproject environment, a set of individual projects. In the case of fixed project structures, the **LD** is binary, where diagonal values are always one, and the binary outdiagonal values specify the adjacency matrix of the given project structure. Nevertheless, UMP enables (1) $0 < l_{ii} < 1$ relative priority values of the task completion in the diagonal and (2) $0 < l_{ij} < 1$ flexible dependency values in the outdiagonal. However, in this study we looked at the original project networks, which did not include priorities and flexible dependencies, and even loops and cycles were not included in the networks. Therefore, all $l_{ii} = 1$, and $l_{ij} = 0$, if $j > i$. **TD** is an n by m matrix domain (submatrix) of positive real values for durations, where $n = =$ number of tasks and m is the number of completion modes. When $m = 1$, the project plan is called single-modal, whereas in the case of $m > 1$, the project plan is called a multimodal project plan. **RD** is an n by $m \cdot \rho$ domain for the renewable resources of each task, where ρ is the number of renewable resources.

Fig. 1 shows synthetic (a) and real (b) activity-on-node project structure.

The two project databases – Boctor's synthetic dataset and Batselier and Vanhoucke's real project dataset – are worth choosing for comparison because they provide complementary perspectives on project network structures. Synthetic projects offer a controlled environment with known completion modes and a randomized network properties, serving as a baseline for understanding theoretical project scheduling behaviors. In contrast, real projects reflect practical complexities, constraints, and resource patterns encountered in actual project management. Comparing these datasets reveals significant differences in flexibility, efficiency, connectedness, and resilience, thereby validating the utility of network-level indicators in distinguishing real project characteristics from synthetic approximations. This comparison enables the calibration and improvement of simulation models and enhances project selection and prioritization methods that are on the basis of more realistic network features. To further improve this argument,



(a) boct1 logic (AoN) network



(b) C2011-08 Sports Center Tielt logic (AoN) network

Fig. 1. Logical structures of synthetic (a) and real (b) project networks.

one could explicitly highlight how the combined use of These datasets help in identifying specific network-level indicators that are robust and generalizable across both synthetic and real contexts, thus reinforcing the empirical and practical relevance of the research contributions. Further databases can be obtained from Kosztyán and Novák [41].

3.2. Applied methods and indicators

The existing project indicators can be distinguished into three clusters: (1) time-related indicators (see Section 3.2.1), (2) (renewable) resource-related indicators (see Section 3.2.2), and structural or logic indicators (see Section 3.2.3). The proposed network indicators (see Section 3.2.4) are closest to the structural indicators, but at the same time complement them in many areas.

3.2.1. Time-related indicators

The time-related indicators can be classified into three groups: (1) *Duration-related indicators*: These indicators measure the average and the variance of the activity durations in a project. They include average activity duration (XDUR) and variance in activity durations (VADUR). These indicators are important because they help to estimate the total project time, to identify the critical path, and to assess the uncertainty and risk of the project schedule. (2–3) *Slack-related indicators* and *Free-slack related indicators*: The slack-related indicators measure the amount and the proportion of slack or float in a project. Slack or float is the amount of time an activity can be delayed without affecting the project completion date. These include number of activities with positive (nonzero) total float (NSLACK), percentage of activities with positive (nonzero) total float (PCTSLACK), total slack ratio (TOTSLACK_R), average slack (XSLACK), and average slack ratio (XSLACK_R). These measures are important because they indicate the degree and the distribution of flexibility or buffer in the project schedule, as well as the potential opportunities for resource leveling or smoothing. The slack-free indicators measure the amount and the proportion of free slack in a project. Free slack is the amount of time an activity can be delayed without affecting the start of any other activity in the project. These measures include number of activities with positive (nonzero) free slack (NFREESLK), percentage of activities with positive (nonzero) free slack (PCTFREESLK), and average free slack (XFREESLK). They are important because they indicate the degree and the distribution of flexibility or buffers for specific activities in the project schedule, and potential opportunities for resource optimization or reallocation.

3.2.2. Resource-related indicators

These indicators measure the availability, demand, use, and efficiency of renewable resources (such as labor, equipment, and materials) that are needed to complete the project activities. They help to plan and allocate the resources accordingly, to avoid resource shortages or surpluses, optimizes resource utilization, and identifies resource bottlenecks or gaps. These indicators can be classified into two groups. (1) *Availability and demand indicators*: These indicators measure how many resources are available and how many resources are required for the project activities. These include the number of renewable resources (NRES), average resource demand (XDMND), percentage of activities requiring positive amount of resource type (PCTR), resource strength (RS), alpha distribution of resource strength (A_RS), and normalized average resource loading factor (NARLF). They help to assess the resource feasibility and to identify the resource gaps for the earliest start schedule, which minimizes the project duration and maximizes the resource utilization. (2) *Utilization and efficiency indicators*: These indicators measure how the resources are used and how efficiently they are used for project activities. These included resource use (RU), average resource utilization (XUTIL), Gini coefficient (GINI), resource constrainedness (for each resource type) (RC), average resource constrainedness over time (XCON), resource factor (RF), total obstruction factor (TOTOFACT) and total underutilization factor (TOTUFACT). These variables help to evaluate resource efficiency and to identify the resource bottlenecks or constraints that can cause delays or inefficiencies in the project schedule. They also help to optimize resource allocation by adjusting the resource demand or supply, by leveling or smoothing the resources, by crashing or fast-tracking the activities, or by changing the activity sequence or duration.

Another classification option is whether a schedule is needed to calculate the indicator. (I) The *nonscheduled resource-related indicators* measure resource characteristics without considering the project schedule or the activity start times. These include the NRES, XDMND, RU, PCTR, GINI, RC, RF, and TOTOFACT. They provide a general overview of the resource availability and demand, and the resource distribution and diversity among the activities. (II) The *early-scheduled resource-related indicators* measure the resource characteristics on the basis of the earliest start times of the activities, which minimizes the project duration and maximizes resource utilization. They include XUTIL, RS, A_RS, NARLF, XCON, and TOTUFACT. These measures provide a detailed analysis of resource feasibility and efficiency as well as the resource gaps and bottlenecks for the earliest start schedule.

3.2.3. Logic indicators

Structural (logical) project indicators (see Table 11) measure the complexity, flexibility, and the interconnectedness of the project network. They help to understand the logic and structure of the project, and how these factors affect the project duration, risk, and resource allocation. The logic indicators can be classified into two groups: (1) The *complexity and flexibility indicators* measure how complex and rigid or how simple and flexible the project network is on the basis of the number and length of the precedent relationships among the activities. These variables include the serial-parallel (SP) indicator (I2), the length of long arcs (LA) (I4), short arcs (SA) (I5), network complexity (C), coefficient of network complexity (CNC), and order strength (OS). Such variables help to assess the uncertainty and variability of the project schedule and the potential opportunities for changing activities sequence or duration. (2) *Interconnectedness and robustness indicators* measure how connected and robust the project network is, on the basis of the number and degree of the nodes and the arcs in the network. These variables include the activity distribution (AD) (I3), topological float (TF) (I6), total activity density (TDENSITY), average activity density (XDENSITY), number of articulation points (AP), components (COMPS), degrees (MEAN_DEGREE), and number of arcs (NARC). They help to evaluate the impact and propagation of delays or disruptions in the project schedule, and the potential opportunities for crashing or fast-tracking activities.

The project network indicators provide a complementary perspective to the logic indicator because the focus is on the network aspects of the project, rather than on the objectives or the outputs. Despite structural indicators already including several network parameters, several important network-level measures are already missing.

3.2.4. Network indicators

Table 12 lists the proposed network-level indicators when examining project networks. Like the logic indicators, the proposed network indicators help to understand the logic and the structure of the project and how they affect the project duration, risk, and resource allocation. These indicators can be separated into seven clusters. (1) The *density and diameter indicators* measure how many dependencies exist in the project network and how far apart the tasks are in the network. They include Dens and Diam. These measures indicate that the degree of complexity and dispersion of the project network and how they influence the project schedule and resource allocation. (2) *Path length and asymmetry indicators* measure how long and how uneven the paths are in the project network, on the basis of the shortest distances between the tasks. They include AVPL and VAs. These indices indicate the degree of concentration and diversity of the project network and how they affect the project schedule and resources allocation. (3) The *motif and assortativity indicators* measure how many recurring patterns and how similar the tasks are in the project network, on the basis of the number and degree of dependencies. They include the number of motifs (Mot) and assortativity (ASSORT). These measures indicate the degree of regularity and the homogeneity of the project network and how they influence the project schedule and resource allocation. (4) *Modularity and community indicators* measure how well the project network can be partitioned into communities, on the basis of different algorithms that optimize the modularity value. These include Leiden's graph modularity value (LeM), Infomap's graph modularity value (IM), and Spin-Glass graph modularity value SGM. These measures of modularity and the community indicate the degree of diversity and cohesion of the project network. Modularity can also help to identify the subprojects or clusters of activities that have strong dependencies within them but weak dependencies with other subprojects or clusters. This can help to improve project management and coordination by assigning different teams or resources to different subprojects or clusters. (5) *Centrality indicators* of how central or important some tasks are in the project network, on the basis of different criteria, such as degree, closeness, eigenvector, PageRank, and betweenness. They also measure how centralized the network is as a whole, on the basis of different indices such as alpha and power. They include in-degree centralization (DZI), out-degree centralization (DZO), degree centralization (DZ), closeness centralization (CZ), harmonic centralization (HZ), eigenvector centralization (EZ), PageRank centralization (PRZ), alpha centralization (AZ), power centralization (PZ), betweenness centralization (BZ), and edge-betweenness centralization (BEZ). These measures indicate the degree of dependency, influence, accessibility, efficiency, importance, impact, intermediation, and hierarchy of the tasks and the network. (6) The *resilience and efficiency indicators* measure how well the project network can maintain its connectivity and efficiency when some tasks or dependencies are removed, either randomly or systematically. These variables include resilience for random attack (RRes), resilience for systematic attack (SRes), average local efficiency (ALE), and global efficiency (GE). They indicate the degree of robustness, susceptibility, redundancy, and fragility of the project network. The role of indicators is emphasized more in the case of agile projects where the task completions must be prioritized [42]. Because of the limited time and resource availability, several tasks should be postponed to a later which can affect the change in the logic structure of the project [43]. (7) Last but not least, the *transitivity indicator* measures how many triangles or closed loops exist in the project network compared to the number of triplets or open paths. It includes Tra. This measure is important because it indicates the degree of cliquishness or openness of the project network.

3.3. Applied supervised and unsupervised machine learning algorithms

To answer the research questions, first, descriptive statistics were used to analyze synthetic and real-life project networks (see $\mathbf{RQ}_1$). ANOVA revealed significant differences between real and synthetic project plans are identified (see $\mathbf{RQ}_3$), and with nongeneric approaches, such as the random forest method, variable importance are also calculated (see Section 3.3.1). To answer $\mathbf{RQ}_2$, *traditional* methods, such as hierarchical clustering and multidimensional scaling, and *nonparametric* unsupervised learning methods are applied to identify groups of indicators, and the clusters of project networks (see Section 3.3.2). These methods identify the relationships between indicators and their classification into indicator groups (see $\mathbf{RQ}_2$). To maximize reproducibility by including the source code of the entire article as supplementary material, which was itself written in R, so that the entire calculation can be repeated.

3.3.1. Descriptive statistics on standardized values and the employed supervised learning methods

Prior to employing nongeneric techniques, such as the random forest (RF) method, we utilized descriptive statistics and analysis of variance (ANOVA) methodologies to investigate the project or network indicator that exhibited the greatest disparity between real and synthetic projects. To ensure a fair comparison, we standardized the variables in the descriptive analysis. The unstandardized results are utilized in ANOVA to determine the significance of differences.

In the case of ANOVA, in addition to R^2 or adjusted R^2, we also calculates η^2. The η^2 measures of effect size or variance explained by a factor or a variable in an ANOVA model. It is different from R^2 or adjusted R^2, which are measures of goodness of fit or how well the model predicts the outcome variable.

The employed supervised learning techniques can demonstrate how the indicators are correlated with each other; therefore, instead of being generic, such as logistic regression or linear discriminant analysis, more robust nongeneric methods, such as the random forest (RF) method, should be used to determine the importance of the variables. We split our database at an 80–20 ratio, where 80% was the training sample and 20% was the test sample. We calculated the accuracy metrics, such as accuracy, balanced accuracy, and F1-score on the test sample.

Since employed supervised learning techniques can show how the indicators are correlated with each other; therefore, instead of being generic, such as logistic regression or linear discriminant analysis, more robust nongeneric methods, such as the random forest (RF) method, should be used to identify the variable importance. Since we obtained almost perfect accuracy both for logistic regression and the random forest method, we did not use any other alternative solution.

3.3.2. Employed unsupervised learning methods

In addition to traditional clustering and dimensionality reduction methods, such as hierarchical clustering and multidimensional scaling techniques for detecting clusters of indicators and project networks, we applied the generalized network-based dimensional reduction method (GNDA) [24] instead of methods such as nonnegative matrix factorization (NMF) [44] due to GNDA's specific advantages in analyzing network structures. GNDA is particularly suited for this research as it can handle both high-dimensional low sample sizes (HDLSS) and low-dimensional high sample sizes (LDHSS), offering greater flexibility and scalability in analyzing diverse project network datasets.

The GNDA method has three mandatory (for dimensional and reduction) and two supplementary steps (for feature selection). Since we applied this method for feature selection, we use only the first three steps. Here, we examined how the supervised clusters are separated from each other.

The GNDA algorithm initially computes the similarity graphs for either the observations or the variables. The similarity function can be either symmetrical, such as with correlation and Euclidean distance, or asymmetric, such as with partial or semipartial correlations. The nodes in this graph can be either observations or variables. Hence, this approach can be employed for clustering either variables or observations. In the second stage, modules are determined on the basis of the similarity graph, where the density of connections between nodes within the modules is greater than that of the nodes in different modules. The community-based detection method is nonparametric, and there is no require that the number of clusters be predefined. During the third stage, the eigenvector centralities are computed for each node. The latent variables for a module are computed as the product of the standardized value of the data and the eigenvector centrality. This process is analogous to principal component analysis However, in this case, eigenvector centralities are defined as weights instead of eigenvalues.

This method has several advantages over traditional unsupervised machines learning techniques.

1. **Scalability**. Kosztyán et al. [45] reported that this method works not only on high-dimensional low sample sizes (HDLSS) but also on low-dimensional high sample sizes (LDHSS).
2. **Flexibility**. Kosztyán et al. [24] reported that this technique can be used with various types of distance functions.
3. **Nonparametric**. There is no need to predefine the number of clusters or dimensions.

The method calculates the NDA factor loadings, which are linear combination of the eigenvector centrality of the indicator and the standardized value of the given indicator. A larger factor absolute value of the factor loadings indicates greater embeddedness in the module and represents a greater influence on the latent variable.

Alternative techniques such as graph learning and spectral clustering could potentially be applied to analyze project networks, as they are well suited for capturing complex relationships in graph-structured data [46]. However, this study opts for supervised and unsupervised learning methods, including random forest and the GNDA, as they offer greater flexibility in handling both network-level and project-specific indicators, and can effectively address the research questions related to comparing real and synthetic projects across multiple dimensions.

Table 2
Mean values of standardized time-related project indicators.

	Synthetic projects				Real
	Mode 1	Mode 2	Mode 3	Mode 4	
XDUR	−0.131	−0.102	−0.119	−0.203	1.880
VADUR	−0.094	−0.091	−0.088	−0.089	1.226
MAXCPL	−0.173	−0.042	−0.057	−0.297	1.925
NSLACK	0.016	0.105	0.060	−0.093	−0.297
PCTSLACK	0.014	0.259	0.116	−0.286	−0.347
TOTSLACK_R	0.090	−0.215	−0.209	0.594	−0.877
XSLACK	−0.159	−0.111	−0.104	−0.146	1.758
XSLACK_R	−0.270	−0.205	−0.129	0.041	1.906
NFREESLK	0.290	0.305	0.157	−0.281	−1.591
PCTFREESLK	0.445	0.467	0.224	−0.512	−2.105
XFREESLK	−0.153	−0.029	−0.015	−0.148	1.164

3.4. Computational complexity

The computational complexity of the methods employed in this study can be grouped into three categories on the basis of their scalability for large-scale projects:

1. Linear complexity $O(n)$ and quasilinear complexity $O(n\log(n))$: Descriptive statistics, ANOVA, matrix representations and the calculation of most traditional and network-level indicators fall into this category, making them highly scalable for large datasets.
2. Polynomial complexity O(n^k): Hierarchical clustering and multidimensional scaling techniques typically involve polynomial-time complexity, which may limit their scalability for very large projects.
3. Higher complexity: Random forest and generalized network-based dimensional reduction method (GNDA) generally results in higher computational complexity. The random forest has a time complexity of $O(ntree \times nfeature \times nsamples \times \log(nsamples))$, where $ntree$ is the number of trees and $nfeature$ is the number of features, and nsamples is the number of samples. GNDA's complexity depends on the specific implementation but can be computationally intensive for large networks.

For large-scale projects, methods in the first two categories would be more readily applicable, whereas those in the third category might require optimization or sampling techniques to maintain computational feasibility.

4. Results

To address these resource questions, we compare the project and the proposed network-level indicators. In Section 4.1, we show the mean of standardized values of the project and network indicators. In Section 4.2, we show that a significant relationship between the network and project (not only structural) indicators. In Section 4.3, we show that these indicators can be used to classify project networks and can be used to detect real and synthetic projects.

4.1. Descriptive statistics

Since the employed indicators can be interpreted at different intervals, Table 2 shows the standardized mean values of the time-related indicators for synthetic and real projects.

The standardized values are calculated by subtracting the mean and dividing by the standard deviation of the original values so that they have a mean of zero and a standard deviation of one. This allows for a fair comparison between different indicators and different projects. Table 2 shows the main differences between the synthetic and real projects that are found in the following indicators: (1) XDUR: The

Table 3
Mean values of standardized resource-related project indicators.

	Synthetic projects				Real
	Mode 1	Mode 2	Mode 3	Mode 4	
NRES	−0.049	−0.049	−0.049	−0.049	0.660
XDMND	−0.032	−0.032	−0.032	−0.032	1.393
RU	0.801	0.069	−0.141	−0.782	0.435
XUTIL	0.032	0.032	0.032	0.031	−1.053
PCTR	1.116	0.067	−0.017	−1.039	−1.047
RS	−0.682	−0.419	0.102	1.375	−0.489
A_RS	0.315	−0.028	0.162	−0.219	−0.778
NARLF	−0.031	−0.031	−0.031	−0.031	0.421
GINI	−1.339	−0.180	0.285	1.238	−0.040
RC	0.032	0.032	0.032	0.032	−1.392
XCON	0.032	0.032	0.032	0.032	−1.392
RF	0.946	0.239	−0.047	−1.074	−0.531
TOTOFACT	−0.024	−0.011	−0.066	−0.070	1.869
TOTUFACT	−0.251	−0.190	0.012	0.152	3.006

average activity duration is much greater for the real projects than that for the synthetic projects, indicating that the real projects have longer and more complex activities. (2) MAXCPL: The maximum critical path length is also much greater for real projects than that for the synthetic projects, indicating that the real projects have longer and more constrained critical paths. (3) PCTSLACK and PCTFREESLK: The percentage of activities with positive (nonzero) total slack and free slack are much lower for real projects than that for synthetic projects, indicating that real projects have increasingly less flexible noncritical activities. (4) TOTSLACK_R and XSLACK_R: Total slack ratio and the average slack ratio are much greater for real projects than those for synthetic projects, indicating that real projects have more slack per activity and per critical path length than does the synthetic projects. (5) NFREESLK: The number of activities with positive (nonzero) free slack is much lower for real projects than that for the synthetic projects, indicating that real projects have fewer activities that can be delayed without affecting the start of their successors.

Table 3 shows the standardized mean values of the resource-related indicators for synthetic and real projects.

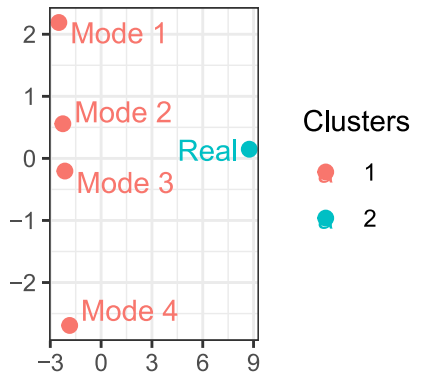
The largest differences between synthetic and real projects are found in the following indicators: (1) NRES: The number of renewable resources is much greater for real projects than for synthetic projects, indicating that real projects have more types of resources available for scheduling activities. (2) XDMND: The average resource demand is much greater for real projects than for synthetic projects, indicating that real projects have more resource requirements for each activity. (3) RU: the resource use varies more for synthetic projects than that for real projects, which indicates that the synthetic projects have different degrees of resource diversity among the activities. (4) XUTIL: The average resource utilization is much lower for real projects than that for synthetic projects, indicating that real projects have lower resource efficiency and higher resource bottlenecks on the longest critical path. (5) PCTR: The percentage of activities requiring a positive amount of resource type is much lower for real projects than that for synthetic projects, indicating that real projects have fewer resources distribution and more resource concentration among the activities. (6) RS: The resource strength is more variable for synthetic projects than that for real projects, indicating that synthetic projects have different degrees of resource feasibility and resource gaps for the earliest start schedule of each activity. (7) A_RS: The alpha distribution of resource strength is much lower for real projects than that for the synthetic projects, indicating that the real projects have less variation in resource strengths across different subprojects or disconnected components in the project network. (8) GINI: The Gini coefficient is more variable for synthetic projects than for real projects, indicating that the synthetic projects have different degrees of resource inequality or diversity among activities. (9) RC and XCON: The resource constraint (for each resource type and over time) is much lower for real projects than for synthetic projects, indicating that real projects have less resource tightness or scarcity for each resource type and over time. (10) RF: The resource factor is more variable for synthetic projects than for real projects, indicating that the synthetic projects have different degrees of resource density or complexity in the resource matrix. (11) TOTOFACT and TOTUFACT: Total obstruction factor and total underutilization factors are much greater for real projects than those for the synthetic projects, indicating that the real projects have more excessive and underutilized resource needs for the earliest start schedule.

Table 4
Mean values of standardized structural (logic) and network indicators.

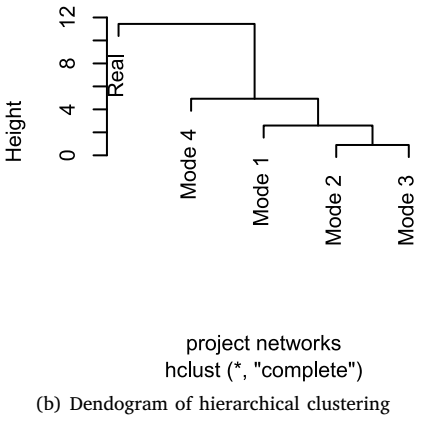
	Synthetic	Real
(a) logic indicators		
TASKS	0.055	−0.745
I2	0.002	−0.026
I3	−0.126	1.700
I4	−0.048	0.655
I5	0.016	−0.215
I6	−0.173	2.338
C	0.111	−1.497
CNC	0.179	−2.420
OS	0.170	−2.303
TDENSITY	0.057	−0.773
XDENSITY	0.025	−0.345
AP	−0.131	1.772
COMPS	−0.054	0.732
MEAN_DEGREE	0.179	−2.420
NARC	0.083	−1.126
(b) network indicators		
Dens	−0.123	1.667
Diam	0.015	−0.201
AVPL	0.028	−0.374
VAs	−0.065	0.878
Mot	0.062	−0.832
ASSORT	−0.037	0.495
LeM	0.046	−0.623
IM	0.069	−0.936
SGM	0.051	−0.685
DZI	−0.081	1.089
DZO	−0.156	2.107
DZ	−0.121	1.639
CZ	−0.064	0.872
HZ	−0.002	0.028
EZ	0.030	−0.412
PRZ	0.024	−0.322
AZ	0.031	−0.416
PZ	0.067	−0.906
BZ	0.040	−0.541
BEZ	−0.018	0.245
RRes	0.083	−1.123
SRes	0.099	−1.341
ALE	0.104	−1.408
GE	−0.042	0.565
Tra	−0.093	1.256

Table 4 shows the standardized mean values of the structural (logic) and network indicators for synthetic and real projects.

The largest differences between synthetic and real projects are found in the following logic and network indicators: (1) Dens: The density is much greater for real projects than that for synthetic projects, indicating that real projects have more arcs per node and more connections between the activities. (2) I3: The activity distribution is much greater for real projects than that for synthetic projects, indicating that the real projects have more balanced and evenly distributed activities. (3) DZO: The out-degree centralization is much higher for real projects than that for synthetic projects, indicating that the real projects have more variation in the number of successors per activity



(a) Multidimensional scaling



(b) Dendogram of hierarchical clustering

Fig. 2. Comparison of real and synthetic datasets on the basis of project and network indicators.

or more activities with many successors. (4) C and CNC: The network complexity and the coefficient of network complexity are much lower for real projects than those for synthetic projects, indicating that real projects have simpler and less dense project networks. (5) OS: The order strength is much lower for the real projects than that for the synthetic projects, indicating that the real projects have fewer constrained and more flexible project networks. (6) AP and COMPS: The number of articulation points and components are greater for real projects than those for synthetic projects, indicating that real projects have more disconnected and modular project networks. (7) SRes: The resilience to systematic attack is much lower for real projects than for synthetic projects, indicating that real projects are more vulnerable to targeted disruptions of activities. (8) Tra: The graph transitivity is much greater for real projects than for synthetic projects, indicating that real projects have more triangles and more clustering of the activities.

Tables 2–4 show that both project and network indicators greatly differ between synthetic and real project datasets. Fig. 2(a) shows that the distance between project and network indicators via multidimensional scaling and Fig. 2(b) shows the results of hierarchical clustering of the project networks.

Fig. 2(a) shows that for the vertical factor and the completion modes also differ. However, these differences are much less than those differences between real and synthetic projects, which is also confirmed by the results of hierarchical clustering (see Fig. 2(b)). The silhouette measure is: 60.31%, which indicates a fair classification. Nevertheless, Tables 2–4 show that distances between real and synthetic projects are significant both for project and network indicators. The largest distance of the project indicators occurs if we compare the real project with completion mode 1 synthetic project networks. Since real and synthetic project plans are significantly separated in the indicators, it is possible to classify project plans by using the data. The classification also reveals which indicator (or indicator groups) is the one that separates real and synthetic projects (see Section 4.3).

4.2. Relationship between project and network indicators

Fig. 3 shows the correlation graph of the project and the network indicators. In the correlation network, the vertices are the indicators, the edges are the correlation relationships between the variables. The thicker an edge is, the closer the connection between them is. The size of the nodes is proportional to the number of correlations. By using the Force Atlas II [47] algorithm, vertices with many connections are located in the center of the network, whereas vertices with few connections are located on the periphery of the network. The colors depend on which indicator belongs to which module. The connections between modules are given the same color. Indicators marked in black do not belong to any module.

The employed NDA method identifies four clusters. This method also calculates the factor loading values of these indicators, which is a linear combination of their eigenvector centralities and the values of latent variables. A higher absolute value of the factor loadings indicates greater embeddedness in the module. Table 5 shows the top five indicators which are ranked by the decreasing absolute values of the factor loadings, while the following indicators do not belong to any module: RU, PCTR, RS, GINI, RC, RF, TOTOFACT, TOTUFACT, XCON, XDMND, XUTIL, which we considered as outliers.

Table 5 shows that in three of the four clusters, the structural and network-level indicators are dominant, whereas the fourth cluster mainly collects time-dependent indicators. Table 6 shows the relative frequencies of the indicator groups in clusters, where cluster zero collects the outliers (mainly resource indicators).

4.3. Similarities and differences between synthetic and real project networks

In this section, we employed the analysis of variance approach to determine the variables that exhibit significant differences between the systematic and real datasets (see Section 4.3.1). Additionally, we utilized the random forest method to compute the importance of each variable (see Section 4.3.2).

4.3.1. Analysis of variances

Table 7 shows which indicators are significantly different in the case of synthetic project networks.

The independent variables are grouped into four types: time, resource, logic, and network. These types represent different aspects of the project network indicators, such as duration, resource allocation, network structure, and network properties. The dependent variable is whether a project network is synthetic or real, which means whether it was generated via a simulation model or derived from a real project.

Table 7 shows that most of the projects (time, resources, logic), and the network indicators are significant at the $p = 0.05$ level. In addition, the variance explained is ($\sum \eta^2 = 96.9\%$) acceptable, and the goodness of fit values are ($R^2 = 96.9\%$, Adj. $R^2 = 96.2\%$) also high.

4.3.2. Calculating relative importances with random forest methods

Table 7 shows that most project and network indicators are significant. However, the independent variables can have nonlinear connections. Therefore, we used the Boruta [27] method to (a) identify significant variables, and afterward, with a random forest method, (2) we rank the indicators by their variable importance.

After Boruta was calculated only the following indicators: number of renewable resources (NRES), number of articulation points (AP) are removed from the model. Fig. 4 shows the variable importance obtained via a tuned random forest method.

Table 8 shows the most important accuracy measures.

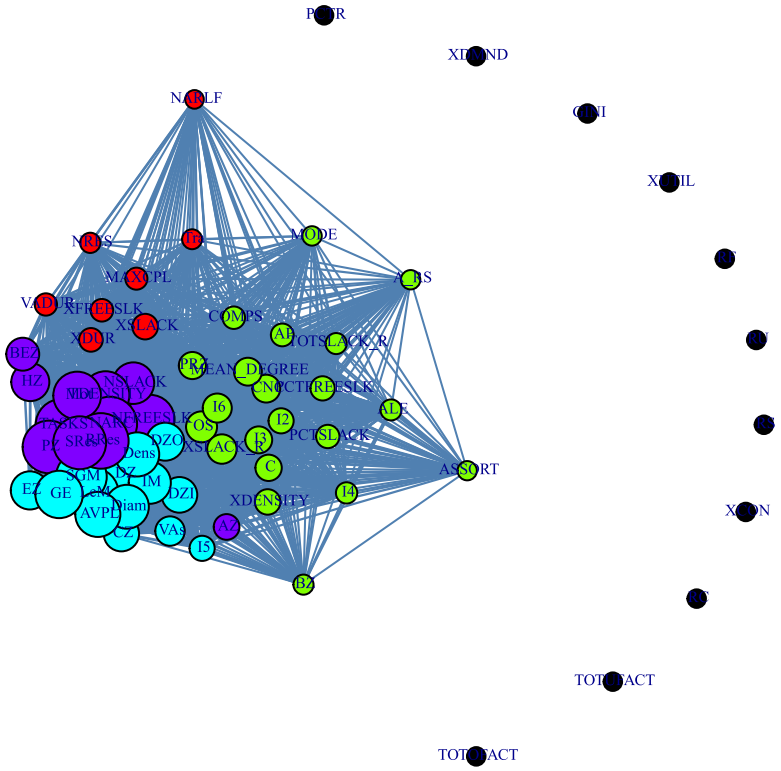


Fig. 3. Clustered correlation graph of the project and network indicators.

Table 5
Top 5 indicators and their NDA factor loadings values.

VAR1	NDA1	VAR2	*NDA2*	VAR3	NDA3	VAR4	NDA4
AP	0.964	Diam	0.998	AZ	1.000	VADUR	1.000
COMPS	0.797	AVPL	0.989	NFREESLK	0.472	XDUR	0.820
XSLACK_R	0.521	LeM	0.834	NARC	0.422	XFREESLK	0.791
OS	−0.500	SGM	0.833	RRes	0.379	XSLACK	0.760
I6	0.498	GE	−0.772	TASKS	0.365	MAXCPL	0.718

Table 6
Percentage distribution of different project and network indicator types.

	Cluster_0	Cluster_1	Cluster_2	Cluster_3	Cluster_4
Time	0.00%	36.36%	0.00%	18.18%	45.45%
Resource	78.57%	7.14%	0.00%	0.00%	14.29%
Logic	0.00%	73.33%	6.67%	20.00%	0.00%
Network	0.00%	16.00%	52.00%	28.00%	4.00%

The accuracy of both the logistic regression and the random forest method is high. The first three indicators: variance in activity durations (VADUR), and average activity duration (XDUR) are time-related indicators. However, the third most important indicator for identifying the separation between synthetic and real project networks is a network indicator: alpha centralization (AZ).

4.4. Unsupervised learning methods for identifying project networks

Fig. 5 shows how the indicators are separated into four-four clusters by using the GNDA method. Fig. 5(a) (Fig. 5(b)) shows the clustering results for only synthetic (only real) project networks are considered.

Fig. 6 shows the clustered project networks if only the proposed network indicators (see Fig. 6(a)) or the project indicators are employed (see Fig. 6(b)).

Fig. 6 shows that both clustering methods provide two-two clusters.

Considering only network indicators the first cluster contains 32.93% synthetic and 67.07% real project networks, whereas the second cluster contains 50% synthetic and 50% real project networks. Considering only project indicators, the first cluster contains 43.96% synthetic and 56.04% real project networks, whereas the second cluster contains 0% synthetic and 100% real project networks.

5. Discussion

According to the research questions, the network-level indicators are proposed, thus providing new insight into project planning. The existing time-, resource-related, and structural indicators and the proposed network-level indicators of real and synthetic project networks are compared.

The findings indicate that real projects exhibit variations compared with synthetic projects in terms of time-related, resource-related, structural, and network-level characteristics. These variations have consequences for the scheduling and allocation of project tasks and resources as well as for the overall performance, risk, and quality of the project.

Table 7
Analysis of variance for detecting the difference in indicators between synthetic and real project networks.

	Type	Df	Sum Sq	Mean Sq	F value	Pr (<F)	Sig.
XDUR	Time	1	12.385	12.385	1859.092	0.000	***
VADUR	Time	1	0.984	0.984	147.730	0.000	***
MAXCPL	Time	1	4.143	4.143	621.854	0.000	***
NSLACK	Time	1	0.237	0.237	35.522	0.000	***
PCTSLACK	Time	1	0.020	0.020	3.010	0.084	.
TOTSLACK R	Time	1	0.231	0.231	34.656	0.000	***
XSLACK	Time	1	10.460	10.460	1570.110	0.000	***
XSLACK R	Time	1	12.800	12.800	1921.302	0.000	***
NFREESLK	Time	1	1.254	1.254	188.300	0.000	***
PCTFREESLK	Time	1	2.188	2.188	328.494	0.000	***
XFREESLK	Time	1	0.151	0.151	22.727	0.000	***
NRES	Resource	1	0.061	0.061	9.195	0.003	**
A RS	Resource	1	0.367	0.367	55.057	0.000	***
NARLF	Resource	1	0.007	0.007	1.111	0.293	
TASKS	Logic	1	0.495	0.495	74.347	0.000	***
I2	Logic	1	0.688	0.688	103.278	0.000	***
I3	Logic	1	1.424	1.424	213.805	0.000	***
I4	Logic	1	0.557	0.557	83.639	0.000	***
I5	Logic	1	0.110	0.110	16.534	0.000	***
I6	Logic	1	0.015	0.015	2.229	0.137	
C	Logic	1	0.002	0.002	0.344	0.558	
CNC	Logic	1	0.295	0.295	44.293	0.000	***
OS	Logic	1	1.094	1.094	164.246	0.000	***
TDENSITY	Logic	1	0.128	0.128	19.257	0.000	***
XDENSITY	Logic	1	0.218	0.218	32.724	0.000	***
AP	Logic	1	0.011	0.011	1.604	0.206	
COMPS	Logic	1	0.009	0.009	1.334	0.249	
MEAN DEGREE	Logic	1	0.199	0.199	29.878	0.000	***
NARC	Logic	1	0.020	0.020	2.979	0.086	.
Dens	Network	1	0.015	0.015	2.256	0.134	
Diam	Network	1	0.137	0.137	20.527	0.000	***
AVPL	Network	1	0.000	0.000	0.065	0.798	
VAs	Network	1	0.018	0.018	2.643	0.105	
Mot	Network	1	0.268	0.268	40.254	0.000	***
ASSORT	Network	1	0.263	0.263	39.416	0.000	***
LeM	Network	1	0.237	0.237	35.649	0.000	***
IM	Network	1	0.200	0.200	29.952	0.000	***
SGM	Network	1	0.015	0.015	2.184	0.141	
DZI	Network	1	0.031	0.031	4.597	0.033	*
DZO	Network	1	0.003	0.003	0.484	0.487	
DZ	Network	1	0.160	0.160	24.010	0.000	***
CZ	Network	1	0.076	0.076	11.337	0.001	***
HZ	Network	1	0.133	0.133	19.942	0.000	***
EZ	Network	1	0.049	0.049	7.323	0.007	**
PRZ	Network	1	0.003	0.003	0.385	0.536	
AZ	Network	1	0.004	0.004	0.565	0.453	
PZ	Network	1	0.005	0.005	0.737	0.391	
BZ	Network	1	0.409	0.409	61.346	0.000	***
BEZ	Network	1	0.092	0.092	13.754	0.000	***
RRes	Network	1	0.087	0.087	13.129	0.000	***
SRes	Network	1	0.013	0.013	2.016	0.157	
ALE	Network	1	0.047	0.047	7.037	0.008	**
GE	Network	1	0.004	0.004	0.553	0.458	
Tra	Network	1	0.264	0.264	39.578	0.000	***

Note: $\sum \eta^2 = 0.969$, $R^2 = 0.969$, Adj. $R^2 = 0.962$.
Sum of squares (Sq) are calculated by type II mode.
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.

Table 8
Accuracy measures of logistic regression and random forest methods.

	Accuracy	AccuracyPValue	Balanced Accuracy	F1
Logistic Reg.	0.98039	6e−05	0.95455	0.95238
Random Forest	1.00000	0e+00	1.00000	1.00000

The time-related indicators in Table 2 indicate that real projects are less flexible and efficient than synthetic projects. This is evident from their longer and more variable durations, longer and more constrained critical paths, fewer and less flexible noncritical tasks, and greater slack per activity and per critical path length. The project manager must be cognizant of these problems and strategically plan in response.

The resource-related indicators (Table 3) confirm that real projects are also less flexible and efficient than are the synthetic projects in

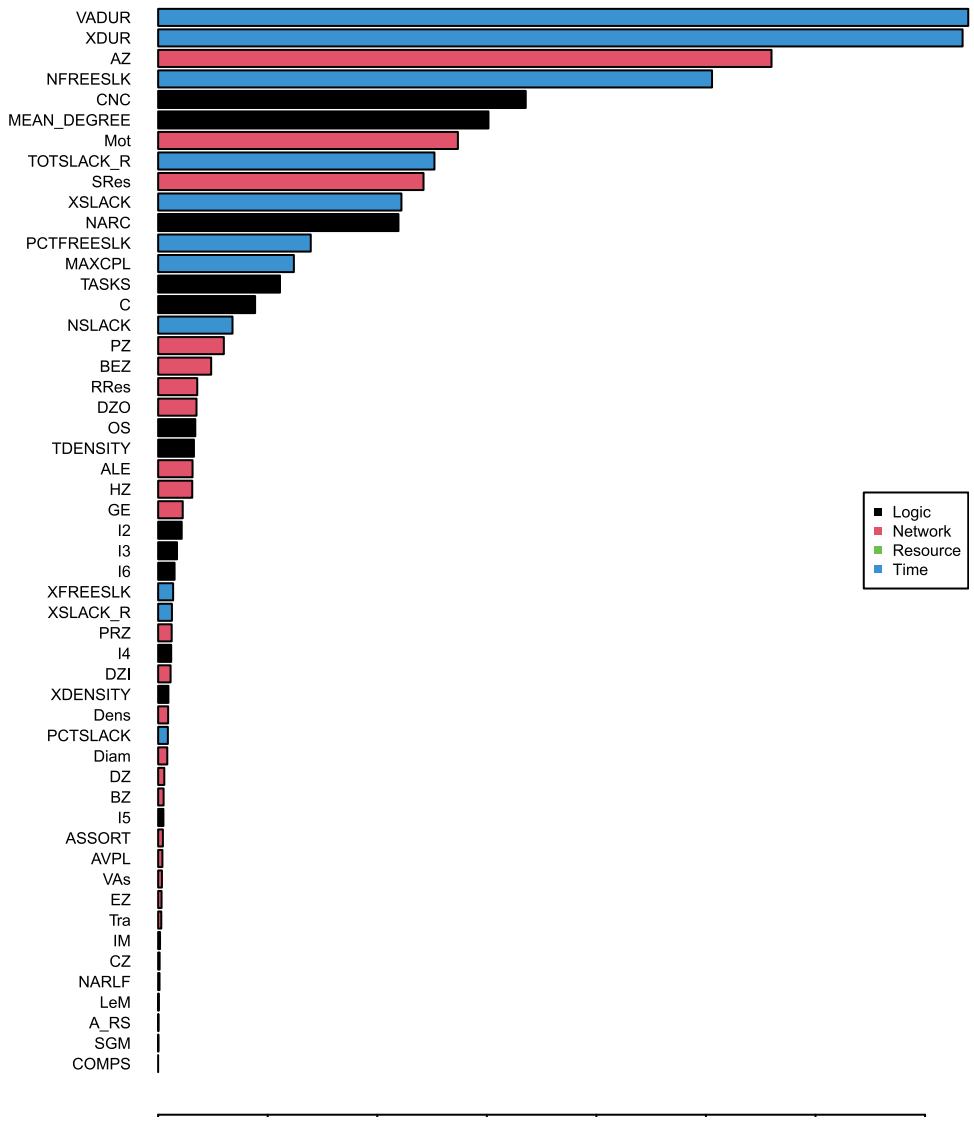


Fig. 4. Ranked indicators by variable importance via the tuned random forest method.

terms of resource planning and allocation, as they have more types and higher demands of resources, lower efficiency and utilization, less resource distribution and diversity, lower resource feasibility and strength, greater resource inequality and concentration, less resource tightness and constrainedness, and more resource excess and underutilization.

The structural and network-level indicators (Table 4) demonstrate that real projects exhibit higher connectedness and reduced complexity compared with those of synthetic projects in terms of the project network and project schedule. Real projects that exhibit a greater number of arcs per node indicates a greater number of connections between activities, a more evenly distributed allocation of activities, simple and less crowded project networks, increased flexibility and reduced constraints, more fragmented and modular networks, heightened vulnerability to targeted disruptions, a greater presence of triangles and clustering of activities, positive assortativity, lower modularity, higher centrality, lower efficiency, and higher transitivity.

The ANOVA results (Table 7) reveal the significant factors in each indicator group, corroborating the descriptive statistics and emphasizing the primary distinctions between the real and synthetic projects. These factors can be utilized to classify projects via supervised and unsupervised learning classification techniques; thus, enhancing the processes of project selection and prioritizing. According to Table 7, Fig. 4 also shows that network indicators have an important role in describing a project network. The results confirm that the network structure of real projects is significantly different than that of synthetic projects.

The results of the employed GNDA method in unsupervised clustering also confirm that, on the basis of the correlation between indicators, four-four clusters can be specified for both synthetic and real projects. Nevertheless, these clusters do not fully follow the grouping found in the literature. In addition, network indicators can be found in most clusters, which suggests that network indicators affect all types of indicators (see Fig. 5).

If we reverse the clustering and group project networks, we see that we can obtain two-two each on the basis of the project and the proposed network-level indicators. On the basis of the project indicators, one group is purely synthetic, whereas the other group contains a mixture of real and synthetic projects. However, if only network indicators are considered, We obtain more mixed clusters. Moreover, the supervised learning results as in Fig. 4) show that the indicators together separate the project networks well.

The differences in flexibility and efficiency between real and synthetic projects could be attributed to the idealized nature of synthetic datasets, which may not fully capture the complexities and constraints of real-world project management. Real projects often face unforeseen

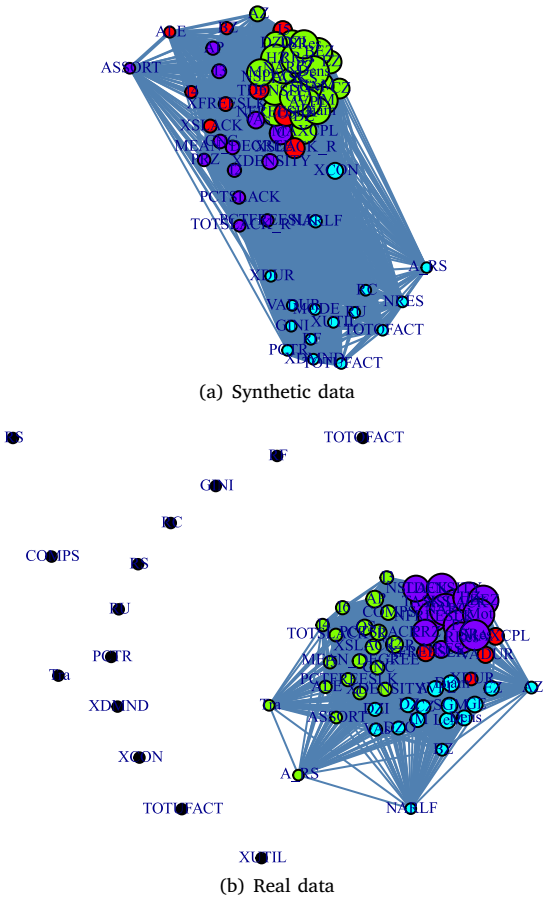


Fig. 5. Clustered correlation graph of the project and network indicators on synthetic and real project databases.

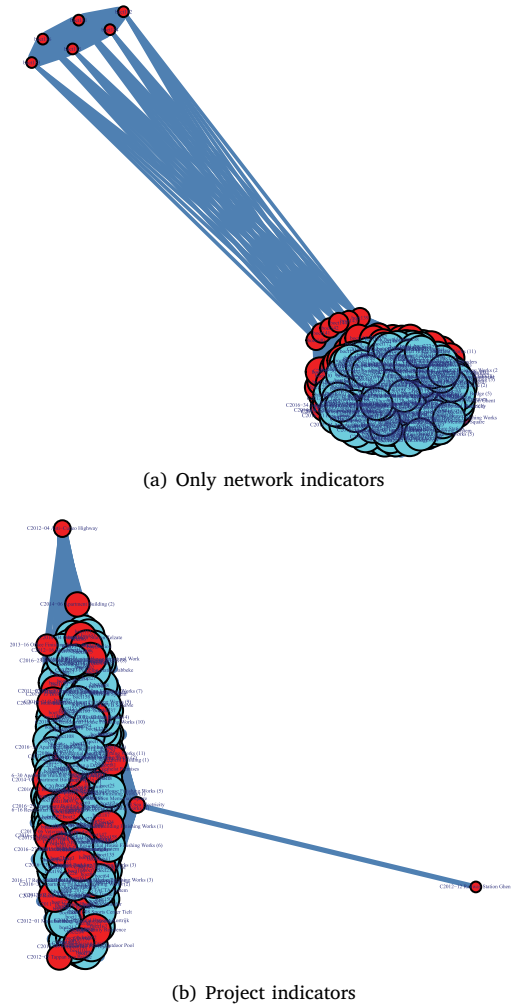


Fig. 6. Clustered Euclidean distance graph of projects for network and project indicators.

challenges, resource limitations, and external dependencies that are not typically modeled in synthetic datasets. Agile methodologies could potentially address these differences by promoting adaptability, continuous reassessment, and iterative planning. The proposed unified matrix planning (UMP) model can support this agile approach by allowing flexible project structures and enabling the prioritization of activities [10]. Kosztyán et al. [10] showed that the UMP model's ability to store both fixed and flexible project structures suggests that it could facilitate the dynamic reassessment of task relationships and priorities, which is crucial in agile project management. In addition, Kosztyán et al. [10] also noted that by introducing task priorities, the traditional project indicators of the resulting flexible project structures are also closer to the idealized indicators of synthetic projects. However, it is an open question whether this is also true for the network-level indicators now proposed. By providing a framework that can represent and analyze both ideal (synthetic) and real project networks, the UMP model could help practitioners identify areas where real projects deviate from ideal conditions and guide them in implementing more flexible and efficient project management practices.

Traditional project indicators and the proposed network-level indicators offer complementary perspectives on project analysis, with some alignments and contrasts. Traditional indicators focus on time, resources, and structural aspects, whereas network-level indicators provide insights into the project's network properties. Both types of indicators measure complexity, but in different ways. For instance, the traditional 'coefficient of network complexity' (CNC) measures connectedness, whereas network-level indicators such as 'density' and 'average path length' offer more nuanced views of network structure. Time-related indicators like 'slack' have some correlation with the network centrality measures, as tasks with high centrality often have less slack. Resource-related indicators do not have direct counterparts in network-level indicators, but network measures can indirectly reflect resource distribution. Structural indicators such as 'order strength' (OS) align somewhat with network modularity measures, both reflecting the project's organizational structure. However, network-level indicators provide additional insights into project robustness, efficiency, and hierarchical structure that is not captured by traditional indicators.

6. Summary and conclusion

In this study, we introduced and examined network-level indicators for activity-on-node project networks, in addition to the preexisting time-related, resource-related, and structural indicators. The network-level indicators are derived from the principles of network theory and graph analysis. These indicators offer fresh perspectives on the qualities and attributes of the project network, including density, diameter, modularity, centrality, resilience, efficiency, and transitivity. Moreover, such indicators can measure emergent and nonlinear characteristics of the project network, including the level of uniformity, integration, hierarchy, and coherence of the network structure. These aspects are not measured by the logic indicators, which are derived from project management theory and logical analysis. These indicators have the

ability to uncover concealed and unforeseen patterns and behaviors within the project network, which can have an impact on project performance, risk, and quality (see Contribution C_1).

A comparison between synthetic and real project networks, was conducted by focusing on project and network-level indicators, and we found the primary distinctions and similarities between them. In this study, we identified several distinct characteristics that differentiate real projects from synthetic projects. These include flexibility, efficiency, diversity, complexity, connectedness, modularity, centrality, resilience and efficiency. These variations affect scheduling and the allocating of resources in a project, as well as for project management and evaluation. In this paper, we propose utilizing supervised and unsupervised learning classification methods for classification projects according to their indicator values. This can enhance the processes of project selection and prioritizing (see Contribution C_3).

We have made significant contributions to the current body of literature. in multiple ways. We presented a novel approach to project networks analysis, which uses network theory and graph analysis. This approach can enhance and expand the current understanding, which is mostly on the basis of project management theory and logical analysis (C_1). Furthermore, we investigated the correlations at the network level. indicators and the current project indicators, uncovering the impact of network features on project aspects and the reciprocal influence they have on each other (C_2). This analysis highlights the potential for integrating and aligning these factors to enhance project management. Furthermore, we assessed the soundness and the relevance of the simulation model. We found disparities between real and synthetic project networks, thereby facilitating enhancements to the simulation model and project network analysis (C_3).

The study has also yielded practical ramifications for scholars and managers. We showed that network theory and graph analysis are valuable and applicable for project network analysis (C_1). A thorough and organized framework has also been presented for evaluating and for comparing project- and network-level indicators (C_2). We provided managers with fresh perspectives and resources to comprehend, to oversee, and to enhance the structure and performance of the project network, as well as to handle uncertainties, alterations and interruptions. We also proposed methods for choosing and ranking projects on the basis of their indicator values, which can enhance the efficiency of resource allocation and strategy implementation (C_3).

7. Limitations and future directions

This study has limitations that could be addressed in future work. The study focuses primarily on static project networks, whereas real projects often have dynamic structures that change over time. Developing methods to analyze and compare dynamic project networks could be valuable extension of this work. Additionally, the study does not consider the impact of external factors such as risk and uncertainty on project networks. Incorporating these elements into the analysis could provide more robust insights for project management.

Limitations of the proposed network properties in complex project networks, several areas warrant further investigation. The density measure, for instance, may not adequately captures the nuances of highly interconnected project networks, where the quality of connections might be more important than their quantity. Modularity measures could be refined to better reflect the practical groupings of tasks in real projects, which may not always align with purely structural divisions. Other network-level indicators, such as centrality measures, might need to be adapted to account for the unique characteristics of project networks, such as temporal dependencies and resource constraints. Future research could explore how these network properties can be modified or complemented to provide more meaningful insights in the context of complex project management. Additionally, developing new indicators that specifically address the challenges of large-scale, multimodal projects could significantly enhance our understanding complex project networks.

Funding

This work was performed for the TKP2021-NVA-10 project with support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, Hungary, financed under the 2021 Thematic Excellence Programme funding scheme. Zsolt T. Kosztyán's research contribution supported by Project no. K 142395 was implemented with support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, Hungary, financed under the K_22 "OTKA'' funding scheme.

Declaration of competing interest

The author retains full responsibility for the content of this publication and there are no competing interests.

Acknowledgments

We would like to thank the University of Pannonia for supporting Dr. Zsolt T. Kosztyán. We would also like to thank the Hungarian Ministry of Culture and Innovation for funding Dr. Zsolt T. Kosztyán.

Appendix A. Formal description of applied indicators

An activity-on-node project network is a set of nodes (i.e.,~tasks) and a set of directed edges (i.e.,~dependencies) that connect some pairs of tasks. In matrix-based project planning, both the logic plan and the time and the resource demands are in a submatrix.

We proposed project indicators, such as time-related, resource-related and structural (logic) and network-level indicators to describe the project plan.

The applied time-related indicators are described below.

Table 9 summarizes the applied time-related indicators. A detailed formal description is provided in the Supplementary material.

- **Average activity duration** (XDUR): This indicator reflects the average amount of time it takes to complete an activity in a project. It is important to measure the XDUR because it helps to estimate the total duration of the project, even if the given scheduling algorithms initiate activities later. The average task duration is calculated as

$$\mathrm{XDUR} := \frac{1}{n}\sum_{i=1}^{n} d_i \tag{1}$$

  where n is the number of tasks and d_i is the duration of the ith task.
- **Variance in activity duration** (VADUR): This indicator measures the variability or dispersion of the activity durations in a project. It is important to measure VADUR because it helps to assess the uncertainty and the risk of the project schedule, which is necessary to plan for contingencies and buffers. The variance in task duration is calculated as follows:

$$\mathrm{VADUR} := \frac{1}{n-1}\sum_{i=1}^{n}\left(d_i - \overline{X}\mathrm{DUR}\right)^2 \tag{2}$$

- **Maximum critical path length** (MAXCPL): This indicator is expressed as the total project time or the longest sequence of activities from the start node to the end node in an AoN project network. It is important for measuring the MAXCPL because it determines the minimum time required to complete the project, and any delay in an activity on the critical path delays the whole project. We can denote MAXCPL as a *total project time* (TPT)

Table 9
Description of time-related indicators.

ID	Variable name	Short description (what it measures)	Why is it important to measure in an activity-on-node project network?
XDUR	average activity duration	Sum of activity durations divided by the number of activities.	Shows the average units of the durations in projects.
VADUR	variance in activity durations	Variance in the duration of activities.	Likely to have an effect on scheduling on the basis of activity duration.
MAXCPL	maximum critical path length	Total project time or the longest critical path length.	The minimal duration of the project/portfolio considering the earliest (resource-unconstrained) start schedule.
NSLACK	number of activities with positive (nonzero) total float	Counts total number of nonzero activity float(s)	Reflects scheduling freedom in the sense that specific activities can be delayed without delaying the completion of a project.
PCTSLACK	percentage of activities with positive (nonzero) total float	The ratio of nonzero activity float(s) and all activities.	
TOTSLACK_R	total slack ratio	Total slack (float) of all activities divided by the critical path length.	
XSLACK	average slack	Average total slack per activity	
XSLACK_R	average slack ratio	Average slack ratio on the critical path	
NFREESLK	number of activities with positive (nonzero) free slack	Counts total number of nonzero activity slack(s)	Reflects scheduling freedom of specific activities can be delayed without delaying any other starts of the activities in the project.
PCTFREESLK	percentage of activities with positive (nonzero) free slack	The ratio of nonzero activity slack(s) and all activities	
XFREESLK	average free slack	Average free slack per activity	

- **Number of activities with positive (nonzero) total float** (NSLACK): This indicator counts the total number of activities that have some slack or float in a project. Slack or float is the amount of time an activity can be delayed without affecting the project completion date. It is important to measure NSLACK because it indicates the degree of flexibility or buffer in the project schedule, as well as potential opportunities for resource leveling or smoothing. Formally:

$$\text{NSLACK} = \sum_{i=1,TS_i>0}^{n} l_{ii} \tag{3}$$

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; $l_{ii} = [\mathbf{LD}]$ is the diagonal value of the logic domain.

- **Percent of activities with positive (nonzero) total float** PCTSLACK: This indicator represents the ratio of activities that have some slack or float to the total number of activities in a project. It is important for measuring PCTSLACK because it reflects the proportion of flexibility or buffer in the project schedule and the potential opportunities for resource leveling or smoothing. The percentage of tasks with positive total slack are calculated as follows:

$$\text{PCTSLACK} := \frac{1}{n}\sum_{i=1}^{n}\begin{cases}1 & \text{if } TS_i = LS_i - ES_i > 0\\ 0 & \text{if } TS_i = LS_i - ES_i = 0\end{cases} \tag{4}$$

- **Total Slack Ratio** (TOTSLACK_R): This indicator shows the ratio of the total slack or float of all activities to the critical path length in a project. It is important to measure TOTSLACK_R because it reflects the overall flexibility or buffer in the project schedule and the potential opportunities for resource leveling or smoothing. The total slack ratio is calculated as follows:

$$\text{TOTSLACK\_R} := \frac{\sum_{i=1}^{n} TS_i}{TPT} \tag{5}$$

- **Average slack** (XSLACK): This indicator shows the average amount of slack or float per activity in a project. It is important to measure XSLACK because it reflects the average flexibility or buffer per activity in the project schedule and the potential opportunities for resource leveling or smoothing. The average total slack per task is calculated as follows:

$$\text{XSLACK} := \frac{1}{n}\sum_{i=1}^{n} TS_i \tag{6}$$

- **Average slack ratio** (XSLACK_R): This indicator shows the average ratio of slack or float per activity to the critical path length in a project. It is important to measure XSLACK_R because it reflects the average flexibility or buffer per activity in relation to project duration and the potential opportunities for resources leveling or smoothing. The average slack ratio is calculated as

$$\text{XSLACK\_R} := \frac{XSLACK}{TPT} \tag{7}$$

- **Number of activities with positive (nonzero) free slack** (NFREESLK): This indicator counts the total number of activities that have some free slack in a project. Free slack is the amount of time an activity can be delayed without affecting the start of any other activity in the project. It is important to measure NFREESLK because indicates the degree of flexibility or buffer for specific activities in the project schedule and the potential opportunities for resource optimization or reallocation.

$$\text{NSLACK} = \sum_{i=1,FS_i>0}^{n} l_{ii} \tag{8}$$

where the earliest finishing time of the task j is $EF_j = ES_j + d_j$; and
$FS_i := \min_{l_{ij}=1} ES_j - EF_i$ is the free slack of task i (lowest early start of successors - early finish).

- **Percent of activities with positive (nonzero) free slack** PCTFREESLK: This indicator reflects the ratio of activities that have some free slack to the total number of activities in a project. It is important for measuring PCTFREESLK because it reflects the proportion of flexibility or buffer for specific activities in the project schedule and the potential opportunities for resource optimization or reallocation. The percentage of activities that have positive (nonzero) free slack can be calculated as follows:

$$\text{PCTFREESLK} := \frac{1}{n}\sum_{i=1}^{n}\begin{cases}1 & \text{if } FS_i > 0\\ 0 & \text{if } FS_i = 0\end{cases} \tag{9}$$

Table 10
Description of resource-related indicators.

ID	Variable name	Short description (what it measures)	Why is it important to measure in an activity-on-node project network?
NRES	number of renewable resources	How many resources are available for scheduling activities.	Number of renewable resources in the project.
XDMND	average resource demand	Average quantity of resources demanded if required by an activity.	Shows the average units of the resources in projects.
RU	resource use	Measures for each activity the number of resource types used.	How tasks are using resources types. Is a vector of the number of different resources each activity uses.
XUTIL	average resource utilization	The average utilization of resource types on the longest critical path length.	Assessing resource efficiency and identifying resource bottlenecks.
PCTR	percentage of activities requiring positive amount of resource type	Percentage of activities that require the given resource type	How resources are distributed among activities.
RS	resource strength	Measures the availability of resources (normalized) for the earliest start schedule.	Assessing the resource feasibility and identifying the resource gaps for the earliest start schedule.
A_RS	alpha distribution of resource strength	Measures the variation (α-distance) of multiple resource strength indicators.	Shows the variation of resource strengths in unconnected (sub)projects.
NARLF	normalized average resource loading factor	Shows whether the resource demands for all projects with the earliest start schedule occur in the first or second half of the critical path duration of the portfolio.	To know whether resource demand is front- or backloaded in the project (portfolio).
GINI	Gini coefficient	How the total work content (each activity's renewable resource demand multiplied by its duration) is distributed among all activities.	(In)equality of renewable resource profiles among tasks.
RC	resource constrainedness (for each resource type)	Measures per resource type the average quantity demanded by all activities divided by its availability.	Considers the amount of the resource needed.
XCON	average resource constrainedness over time	Sum of renewable resource demands (divided by the number of non-zero demands) divided by the availability of the given resource type.	Reflects the "tightness" of certain resource types.
RF	resource factor	Average portion of resource types requested per activity.	Density of the resource matrix. Does not account for the amount of the resource needs.
TOTOFACT	total obstruction factor	Excess resource requirements divided by resource work content (duration multiplied by resource demand), measured for the earliest start schedule.	Excessive resource needs.
TOTUFACT	total underutilization factor	Total resources need less than available divided by work content (duration multiplied by resource demand) for the earliest start schedule.	Reflects underutilization (demand below availability) of resources.

- **Average free slack** (XFREESLK): This indicator shows the average amount of free slack per activity in a project. It is important to measure the XFREESLK because it reflects the average flexibility or buffer for specific activities in the project schedule and the potential opportunities for resource optimization or reallocation.

$$\text{XFREESLK} := \frac{1}{n}\sum_{i=1}^{n} FS_i \tag{10}$$

Formal definitions of the employed resource indicators:

Table 10 summarizes the applied resource-related indicators. A detailed formal description is provided in the Supplementary material.

We write $\mathcal{S}$ as a realized project structure and $\mathbf{LD} \in \{0,1\}^{n\times n}, \mathbf{T} \in \mathbb{R}_+^n$ and $\mathbf{RD} \in \mathbb{R}_+^{n\times\rho}$ as a domain 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 write $t_i = [\mathbf{TD}]_{ii}$ as the duration of task i and TPT as the duration of the project, and $r_{ij} = [\mathbf{RD}]_{ij}$ as the resource demand of task i of resource j.

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

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

$$r_{ij}(\tau) := \begin{cases} r_{ij} \text{ if } a_i(T) \in \overline{\mathcal{S}}, \tau \in T_i \\ 0 \text{ otherwise} \end{cases} \tag{11}$$

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

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

$$\text{RF} := \frac{1}{n\rho}\sum_{i=1}^{n}\sum_{j=1}^{\rho}\begin{cases} 1 & \text{if } r_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} = \frac{1}{\rho}\sum_{j=1}^{\rho}\text{PCTR}_j \tag{12}$$

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

$$\text{PCTR}_j := \frac{1}{n}\sum_{i:=1}^{n}\begin{cases} 1 & \text{if } r_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

- **Resource Use** (RU), represents the resource use for each activity, i.e., the number of resource types used. The RU varies between 0 and r (the number of resource types). It is a rowwise view of RF

$(i = 1, \dots, n)$ as follows:

$$\mathrm{RU}_i := \sum_{j=1}^{\rho} \begin{cases} 1 & \text{if } r_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

- DMND_j is **the average quantity of resources** j demanded when needed by an activity $(j = 1, \dots, \rho)$ as follows:

$$\mathrm{DMND}_j := \frac{\sum_{i=1}^{n} r_{ij}}{\sum_{i=1}^{n} \begin{cases} 1 & \text{if } r_{ij} > 0 \\ 0 & \text{if } r_{ij} = 0 \end{cases}} \tag{15}$$

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

$$\mathrm{RC}_j := \frac{\mathrm{DMND}_j}{\alpha_j} \tag{16}$$

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

The following indicators of Patterson [48] require early scheduling $(\vec{S}_{\min})$ 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:

$$\mathrm{RS}_j := \frac{\alpha_j - r_j^{\min}}{r_j^{\max} - r_j^{\min}} \tag{17}$$

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. The $RS_i^{(\alpha)}$ can be calculated as follows:

$$RS_i^{(\alpha)} = \begin{cases} \frac{RS_i^{\alpha}-1}{\alpha}, & \alpha \neq 0 \\ \log(RS_i), & \alpha = 0 \end{cases} \tag{18}$$

the α-distance between the RS distributions of different subprojects or disconnected components of the project network, which is a measure of how different two distributions are on the basis of the alpha power transformation. The α-distance can be calculated as:

$$D_\alpha(RS_A, RS_B) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(RS_{A,i}^{(\alpha)} - RS_{B,i}^{(\alpha)})^2} \tag{19}$$

where RS_A and RS_B are the vectors of RS indicators for subprojects A and B, respectively, and n is the number of activities in each subproject. A_RS indicator, which is the average α-distance between the RS distributions of all possible pairs of subprojects or disconnected components in the project network. The A_RS indicator can be calculated as:

$$\mathrm{A\_RS} = \frac{2}{m(m-1)} \sum_{A<B} D_\alpha(RS_A, RS_B) \tag{20}$$

where m is the number of subprojects or disconnected components in the project network, and the summation is over all possible pairs of subprojects or components.

- UTIL_j is the **utilization** (rate) of resources, and it is measured on the basis of 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.

$$\mathrm{UTIL}_j := \frac{\sum_{i=1}^{n} r_{ij} t_i}{\alpha_j \cdot \mathrm{TPT}} \tag{21}$$

- The applied XUTIL is the **average resource utilization**.
- TCON_j is the **constraint on the (renewable) resource type** j over time. In practice, this is the average utilization (UTIL_j) considering only those tasks that use that particular resource type as follows:

$$\mathrm{TCON}_j := \frac{\sum_{i=1}^{n} r_{ij} t_i}{\alpha_j \cdot \mathrm{TPT} \cdot \sum_{i=1}^{n} \begin{cases} 1 & \text{if } r_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}} \tag{22}$$

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

$$\mathrm{OFACT}_j := \frac{\int_0^{\mathrm{TPT}} \max\{0; r_j(\tau) - \alpha_j\} d\tau}{\sum_{i=1}^{n} r_{ij} t_i} \tag{23}$$

- **Total obstruction factor** (TOTOFACT=$\sum_j OFACT_j$ is the sum of UFACT for resources.
- UFACT_j is the **underutilization factor** and is calculated as follows:

$$\mathrm{UFACT}_j := \frac{\int_0^{\mathrm{TPT}} \max\{0; \alpha_j - r_j(\tau)\} d\tau}{\sum_{i=1}^{n} r_{ij} t_i} \tag{24}$$

- The **total underutilization factor** (TOTUFACT=$\sum_j OFACT_j$)): sum of UFACT for resources.
- The **average resource loading factor** (ARLF), suggested by Kurtulus and Davis [49], represents the resource distribution of projects. If resource requirements are in the first half of the project, it has a negative value, whereas a positive value means that the resource demands are in the latter half of the project; these values are on the basis of the critical path duration of each project. For multiple (sub)projects, the possible issue of averaging individual ARLF values are managed by NARLF, where normalization is performed with the critical path duration of all (sub)projects [14]. The formula is further improved by [15] with NARLF, which, on top of this, determines if a resource demand falls at the beginning or the end on the basis of the critical path of the portfolio instead of the critical path for each project. It uses two auxiliary variables:

$$Y_{i(\tau)} = \begin{cases} 1 & \text{if activity } a_i \text{ is active at time } \tau, \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

$$Z_{i(\tau)} = \begin{cases} -1 & \text{if } \tau \leq \lceil TPT/2 \rceil, \\ 1 & \text{if } \tau > \lceil TPT/2 \rceil, \end{cases} \tag{26}$$

to obtain:

$$\mathrm{NARLF} = \frac{1}{TPT} \sum_{\tau}^{TPT} \sum_{i=1}^{n} \sum_{j=1}^{\rho} Z_{i(\tau)} Y_{i(\tau)} \left(\frac{r_{ij}}{|\{r_{ij} : r_{ij} > 0\}|} \right) \tag{27}$$

where τ is the time for the earliest start schedule considering release dates.

- The **Gini coefficient** is a measure of inequality that compares the actual distribution of a variable, such as income, wealth, or work content, with a hypothetical equal distribution. The Gini coefficient formula is:

$$\mathrm{GINI} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2n^2 \bar{x}} \tag{28}$$

where n is the number of observations, x_i is the value of the variable for the ith observation, and $\bar{x}$ is the mean value of the variable.

To apply this formula to the work content of activities, we define the variable x_i as the product of the renewable resource and the duration of the ith activity.

The Gini coefficient ranges from 0 to 1, where 0 means perfect equality (all activities have the same work content) and 1 means perfect inequality (one activity has all the work content). A higher Gini

Table 11
Description of logic indicators.

ID	Variable name	Short description (what it measures)	Why is it important to measure in an activity-on-node project network?
TASKS	number of activities	Measures the size of the network as the number of activities.	How many activities are present in the project (portfolio).
I2	serial-parallel (SP)	Measures the closeness to series or parallel graph.	Determines the number of serial activities in the network on the longest chain.
I3	activity distribution (AD)	Measures the distribution of the activities over the progressive levels on the basis of the width of each progressive level in the network.	Measures how the project activities that do not belong to the longest chain are distributed in the network.
I4	the length of long arcs (LA)	Measures the presence of long precedence relations. The length of an arc is the difference between the progressive level of the end node and the start node.	A lower value indicates many dependencies between two levels of activities; a higher value indicates the presence of many immediate successors at the next level of the network.
I5	short arcs (SA)	Measures the presence of short precedence relations. The length of an arc is the difference between the progressive level of the end node and the start node.	Alternative for I4 (length of long arcs).
I6	topological float (TF)	Ont the basis of the difference between the progressive and regressive levels of each activity, the topological float indicator is defined as the fractional total topological float in relation to its maximum.	The topological float of a precedence relation is the number of levels that each activity can shift without violating the maximal level of the network.
C	network complexity	The average number of non-redundant arcs per node.	The number of redundant arcs should not have an influence on project complexity.
CNC	coefficient of network complexity	A measure of the connectedness of a graph takes redundant nodes into account.	A higher complexity means greater connectedness of the network. However, many network problems become easier with increasing CNC.
OS	order strength	The number of precedence relations (including the transitive ones) divided by the theoretical maximum number of precedence relations.	Assessing the complexity and flexibility of the project network and the project schedule.
TDENSITY	total activity density	Measure the interconnectedness of a network. The sum of the maximal number of predecessors minus successors of each activity.	Influences when (in terms of network logic) an activity can be scheduled
XDENSITY	average activity density	Sum of the maximal number of predecessors minus successors of each activity, divided by the number of activities	Influences when (in terms of network logic) an activity can be scheduled
AP	number of articulation points	The number of tasks (and their dependencies) to remove to disconnect the graph (cut vertices).	Removing of articulation points increases the number of connected components in a graph.
COMPS	components	Size of the largest (strongly) connected component.	The robustness of the network against disruptions.
MEAN_DEGREE	degrees	The average number of dependencies connected to each task.	The average degree of each task.
NARC	number of arcs	Number of arcs (precedence relationships).	How many dependencies each task have.

coefficient indicates a more uneven distribution of work content among the activities.

Formal definitions of the employed logic indicators:

Table 11 summarizes the applied resource-related indicators. A detailed formal description is provided in the Supplementary material.

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

- TASKS=I_1, the **number of tasks** (nodes), is calculated as follows:

$$I_1 := n \tag{29}$$

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

$$PL_i := \begin{cases} 1 & \text{if } P_i = \emptyset \\ \max_{j\in P_i} PL_j + 1 & \text{if } P_i \neq \emptyset \end{cases} \tag{30}$$

and

$$RL_i := \begin{cases} m & \text{if } S_i = \emptyset \\ \min_{j\in S_i} RL_j - 1 & \text{if } S_i \neq \emptyset \end{cases} \tag{31}$$

where $m = \max_i PL_i$. Next, we have the following:

$$I_2 := \begin{cases} 1 & \text{if } n = 1 \\ \frac{m-1}{n-1} & \text{if } n > 1 \end{cases} \tag{32}$$

- 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 $j = 1,\dots,m$ as follows: $\mathbf{PL}_j := \{i \le n : PL_i = j\}$, i.e., the *set* of all tasks having progressive level number j. Then,

$$I_3 := \begin{cases} 0 & \text{if } m = 1 \text{ or } m = n \\ \frac{\alpha_w}{\alpha_{\max}} = \frac{\sum_{j=1}^{m} |w_j - \overline{w}|}{2(m-1)(\overline{w}-1)} & \text{if } 1 < m < n \end{cases} \tag{33}$$

where $w_j = |\mathbf{PL}_j|$ is the width (size) of progressive level $j = 1,\dots,m$, $w = (w_1, w_2, \dots, w_m)$ is the vector which contains 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,[1] $\alpha_{\max} = (m-1)(\overline{w}-1)+(n-m+1-\overline{w})$ this formula simplifies to $\alpha_{\max} = 2(m-1)(\overline{w}-1)$.

- 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 % $L(i\_1, i\_2) := |PL\_\{i\_1\} - PL\_\{i\_2\}|$ $, % i.e., 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 $1 \le L \le m-1$. Then, I_4 is calculated as follows:

$$I_4 := \begin{cases} 1 & \text{if } D = n - w_1 \\ \frac{n'_1 - n + w_1}{D - n + w_1} & \text{if } D > n - w_1 \end{cases} \tag{34}$$

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

$$I_5 := \begin{cases} 1 & \text{if } |\mathcal{A}| = n - w_1 \\ \frac{\left(\sum_{L=2}^{m-1} n'_L \frac{m-L-1}{m-2}\right) + n'_1 - n + w_1}{|\mathcal{A}| - n + w_1} & \text{if } |\mathcal{A}| > n - w_1 \end{cases} \tag{35}$$

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

$$I_6 := \begin{cases} 0 & \text{if } m \in \{1, n\} \\ \frac{\sum_{i=1}^{n} |RL_i - PL_i|}{(m-1)(n-m)} & \text{if } m \notin \{1, n\} \end{cases} \tag{36}$$

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

$$\text{CNC} = \frac{|\mathcal{A}|}{n} \tag{37}$$

- OS, the **order strength**, is calculated as follows:

$$\text{OS} = \frac{|\mathcal{A}|}{n(n-1)/2} \tag{38}$$

- C, the **network complexity**, is calculated as follows:

$$\text{C} = \begin{cases} \frac{\log \frac{|\mathcal{A}|}{n-1}}{\log \frac{n^2-1}{4(n-1)}} & \text{if } n \text{ is odd} \\ \frac{\log \frac{|\mathcal{A}|}{n-1}}{\log \frac{n^2}{4(n-1)}} & \text{if } n \text{ is even} \end{cases} \tag{39}$$

- TDENSITY, the **total activity density**, is calculated as follows:

$$\text{TDENSITY} := \sum_{i:=1}^{n} \max\left\{0, |P_i| - |S_i|\right\} \tag{40}$$

S_i and P_i were defined immediately before I_2.

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

$$\text{XDENSITY} := \frac{\text{TDENSITY}}{n} \tag{41}$$

Formal definitions of the employed network-level indicators:

Table 12 summarizes the applied resource-related indicators. A detailed formal description is provided in the Supplementary materials.

[1] 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 $1 < a \le \overline{w} \le b < n$ produces this extrema.

- **Density** (Dens): This measures how many dependencies exist in the project network, relative to the maximum possible number of dependencies. A high density means that the project network is more connected, whereas a low density means that the network is sparser. This indicator reflects the degree of complexity of the project network. The density of a network with n tasks and m dependencies are given by a directed network.

$$Dens = \frac{m}{n(n-1)} \tag{42}$$

- **Diameter** (Diam): This measures the longest shortest distance between any pair of tasks in the project network (without considering task durations). A high diameter means that the network is more dispersed, whereas a low diameter means that the network is more compact. *This indicator reflects the degree of dispersion or concentration of the project network.* The diameter of a network is given by

$$Diam = \max_{u,v \in V} d(u,v) \tag{43}$$

where V is the set of tasks and $d(u,v)$ is the shortest path distance between tasks u and v

- **Average path length** (AVPL): This measures the average shortest distance between any pair of tasks in the project network (without considering task durations). A high average path length means that the network is more dispersed, while a low average path length means that the network is more compact. *This indicator also reflects the degree of dispersion or concentration of the project network*. The average path length of a network is given by

$$AVPL = \frac{1}{n(n-1)} \sum_{u,v \in V} d(u,v) \tag{44}$$

- **Vertex asymmetries** (VAs): A task with a high vertex asymmetry may be a bottleneck, a source of risk in the network, or a task that needs more coordination or communication. The average value of the vertex asymmetries measure how asymmetric the network is in terms of the distribution of dependencies among the tasks. A high vertex asymmetry means that the network is more skewed, whereas a low vertex asymmetry means that the network is more balanced. *This indicator reflects the degree of inequality or diversity in the network structure.* The vertex asymmetry of a network is given by

$$VAs = \frac{1}{n(n-1)} \sum_{u,v \in V} |k_u - k_v| \tag{45}$$

where V is the set of tasks and k_u and k_v are the degrees of tasks u and v, respectively.

- **Number of motifs** (Mot): This metric measures the number of recurring patterns of dependencies in the project network. A high number of motifs means that the project network has a rich structure, while a low number of motifs means that the project network has a simple structure. *This indicator reflects the degree of regularity or randomness of the network.* The number of motifs in a network is given by

$$M = \sum_{s \in S} f_s \tag{46}$$

where S is the set of all possible subgraphs and f_s is the frequency of subgraph s in the network.

- **Assortativity** (Assort): This measures how similar the tasks are in terms of their degree, which is the number of dependencies they have. A positive assortativity means that tasks tend to depend on or influence other tasks that have similar degrees, while a negative assortativity means that tasks tend to depend on or influence other tasks that have different degrees. *This indicator represents the degree of homogeneity or heterogeneity in the project network structure.* The assortativity of a network is given by

$$r = \frac{\sum i,j A_{ij}(k_i - \bar{k})(k_j - \bar{k})}{\sum i,j A_{ij}(k_i - \bar{k})^2} \tag{47}$$

Table 12
Description of logic indicators.

ID	Variable name	Short description (what it measures)	Why is it important to measure in an activity-on-node project network?
Dens	density	How many dependencies exist in the project network	Degree of complexity of the project network
Diam	diameter	The longest shortest distance between any pair of tasks in the project network.	The degree of dispersion or concentration of the project network.
AVPL	average path length	Average shortest distance between any pair of tasks in the project network	The degree of dispersion or concentration of the project network.
VAs	vertex asymmetries	How asymmetric the network is in terms of the distribution of dependencies among the tasks.	The degree of inequality or diversity in the network structure.
Mot	number of motifs	The number of recurring patterns of dependencies in the project network.	The degree of regularity or randomness of the network.
ASSORT	assortativity	How similar the nodes are in terms of their degree; the degree is the number of connections they have.	Revealing the degree of homogeneity or heterogeneity in the project network structure
LeM	Leiden's graph modularity value	How well the network can be partitioned into communities.	Degree of diversity or cohesion in the project network.
IM	Infomap's graph modularity value	How well the network can be partitioned into communities.	Degree of diversity or cohesion in the project network.
SGM	Spin-Glass graph modularity value	How well the network can be partitioned into communities.	Degree of diversity or cohesion in the project network.
DZI	in-degree centralization	How concentrated the dependencies are on a few activities	The degree of dependency that some activities have over others.
DZO	out-degree centralization	How concentrated the dependencies are on a few activities	Degree of influence that some activities have over others.
DZ	degree centralization	How concentrated the dependencies are on a few activities	Degree of dependency and influence that some activities have over others.
CZ	closeness centralization	How close a task is to all other tasks in the network, on the basis of the sum of the shortest distances between them.	Degree of accessibility or reachability that some tasks have in the network.
HZ	harmonic centralization	How close a task is to all other tasks in the network, which is on the basis of the harmonic mean of the distances between them.	The degree of efficiency or effectiveness that some tasks have in the project network.
EZ	eigenvector centralization	How connected a task is to other well-connected tasks in the network, on the basis of the eigenvector of the adjacency matrix.	Degree of importance that some tasks have in the project network
PRZ	PageRank centralization	How influential or critical a task is for overall project performance.	The degree of impact that some tasks have on the project network.
AZ	alpha centralization	This measures how centralized the network is as a whole, on the basis of the alpha index.	Degree of hierarchy or equality in the project network structure.
PZ	power centralization	This measures how centralized the network is as a whole on the basis of the power index.	Degree of dominance or balance in the project network structure.
BZ	betweenness centralization	How often a task lies on the shortest path between two other tasks.	Degree of intermediation that some tasks have in the project network.
BEZ	edge-betweenness centralization	How often an edge lies on the shortest path between two tasks	The degree of vulnerability or robustness of the project network.
RRes	resilience for random attack	How well the network can maintain its connectivity when a fraction of tasks are randomly removed.	Degree of robustness or susceptibility of the network.
SRes	resilience for systematic attack	How well the network can maintain connectivity when a fraction of (high-degree) tasks are systematically removed.	Degree of robustness or susceptibility of the network.
ALE	average local efficiency	Average efficiency of the subgraphs induced by the dependencies of each task in the project network.	Degree of redundancy or fragility of the project network.
GE	global efficiency	Average efficiency of the network as a whole.	Degree of redundancy or fragility of the project network.
Tra	graph transitivity	Ratio of triangles to triplets in an project network.	Degree of cliquishness or openness of the project network.

where A_{ij} is the adjacency matrix, k_i and k_j are the degrees of tasks i and j, and $\bar{k}$ is the average degree of the network.

- **Leiden's** (LeM)/**Infomap's** (IM)/**Spin-Glass graph modularity value** (SGM): This measures how well the network can be partitioned into communities on the basis of the Leiden's/Infomap / Spin-Glass algorithm. A high graph modularity value means that the network has a clear community structure, whereas a low Infomap's graph modularity value means that the network is more homogeneous or random. *This indicator reflects the degree of diversity or cohesion in the network.* The similarity between these methods is that they all try to find the best partition of the network into communities that maximizes some objective function, which is related to the modularity of the network. The difference is that they use different algorithms, models, and parameters to achieve this goal, which can result in different community structures and modularity values. The modularity

value of a network is given by

$$Q = \frac{1}{2m} \sum i,j \left(lij - \gamma \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) \tag{48}$$

where l_{ij} (edge) is an element of the logic domain **LD** (adjacency matrix) and m is the number of dependencies, k_i and k_j are the degrees of task i and j, γ is the resolution parameter, c_i and c_j are the communities of tasks i and j, and δ is the Kronecker delta function. The Leiden algorithm is an improving the Louvain algorithm, which is a greedy optimization method that maximizes the modularity of the network. The modularity is a measure of the extent to which the network can be divided into densely connected groups of tasks with few dependencies between them. The Leiden algorithm improves upon the Louvain algorithm by refining the community [50], by using local moves to optimize the modularity, and by using a smart initialization that prevents empty communities. The Infomap algorithm of Rosvall and Bergstrom [51] is on the basis of the idea of using community partitions of the network as a Huffman code that compresses the information about a random walker exploring the network. The random walker moves along the dependencies of the network with probabilities given by a Markov transition matrix. The Infomap algorithm tries to find the optimal partition that minimizes the expected description length of the random walker's trajectory, which is equivalent to maximizing the map equation, a measure of the information flow in the network. The Spin-Glass algorithm of Reichardt and Bornholdt [52] is on the basis of the analogy of a Spin-Glass system in physics, where each task is a spin that can take a finite number of states and each dependency is a coupling that determines the energy of the system. The Spin-Glass algorithm tries to find the optimal partition that minimizes the Hamiltonian, a measure of the energy of the system, which is equivalent to maximizing the Spin-Glass modularity, a measure of the difference between the actual and the expected number of dependencies within and between communities.

- **Degree** (DZ)/**in-degree** (DZI)/**out-degree** (DZO) **centralizations**: This measures how concentrated the total/incoming / outgoing dependencies are on a few nodes. A high degree/in-degree/out-degree centralization means that there are some nodes that send/receive with many dependencies, while a low in-degree centralization means that the dependencies are more even distributed among the activities. In other words, high degree centralizations indicate that some activities are highly dependent or influential on others. *This indicator can reflect the degree of dependency or the influence that some activities have over others.* The degree/in-degree /out-degree centralization of a network is given by

$$DZ = \frac{\sum_{i \in V} (\max_{j \in V} k_j - k_i)}{(n-1)(n-2)} \tag{49}$$

$$DZI = \frac{\sum_{i \in V} (\max_{j \in V} k_j^{in} - k_i^{in})}{(n-1)(n-2)} \tag{50}$$

$$DZO = \frac{\sum_{i \in V} (\max_{j \in V} k_j^{out} - k_i^{out})}{(n-1)(n-2)} \tag{51}$$

where V is the set of tasks, n is the number of tasks, and k_i / k_i^{in} / k_i^{out} and k_j / k_j^{in} / k_j^{out} are the degrees/in-degrees / out-degrees of tasks i and j, respectively.

- **Closeness centralization** (CZ) / **Harmonic centralization** (HZ): This measures how close a task is to all other tasks in the project network; this measure is on the basis of the sum of the shortest distances/the harmonic mean of the distances between them (without considering task durations). A high closeness/harmonic centralization means that there are some tasks that are close to all other tasks, while a low closeness/harmonic centralization means that the tasks are more distant from each other. A task with a low harmonic centrality value may be a redundant task that does not contribute much to the project, or a task that consumes many resources or time. A task with high closeness centralization may be a critical task that affects many other tasks or a task that requires considerable resources or attention. *CZ reflects the degree of accessibility or reachability; the HZ reflects the efficiency or effectiveness of some tasks have in the project network.* The similarity between closeness centralization and harmonic centralization is that they both use the inverse of the distances between tasks to measure their closeness. The difference is that closeness centralization uses the arithmetic means of the inverse distances, whereas harmonic centralization uses the harmonic mean of the inverse distances. The harmonic mean is more sensitive to small values than is the arithmetic mean. This fact implies that harmonic centralization gives more weight to the tasks that are far from others, while closeness centralization gives more weight to the tasks that are close to others. Therefore, harmonic centralization can be more useful for detecting outliers or isolated tasks in the network, whereas closeness centralization can be more useful for detecting hubs or central tasks in the network.
- **Eigenvector centralization** (EZ): This measures how connected a task is to other well-connected tasks in the network, on the basis of the eigenvector of the adjacency matrix. High eigenvector centralization means that the network is more skewed, with a few tasks having higher eigenvector centrality scores than do the remaining scores. This can indicate that the network has a dominant or complex structure, with some tasks having more dependencies than do others. Low eigenvector centralization means that the network is more balanced, with the tasks having similar eigenvector centrality scores. This can indicate that the network has a democratic or simple structure, with the tasks having equal influence or responsibility. *This indicator reflects the degree of importance that some tasks have in the project network.*
- **PageRank centralization** (PRZ): The PageRank centrality value of a task identifies the tasks that have a high impact on the overall project network. For example, a task with a high PageRank centrality value may be a critical task that affects many other tasks, or a task that requires considerable resources or attention. PRZ measures how influential or critical a task is for overall project performance. A high PageRank centralization means that there are several tasks in which have more impact on other tasks, while a low PageRank centralization means that the tasks are more equally impacted. *This indicator reflects the degree of impact that some tasks have on the project network.*
- **Alpha centralization** (AZ)/**Power centralization** (PZ): This measures how centralized the network is as a whole, on the basis of the alpha /power index. A high alpha/power centralization means that the network is more centralized, while a low alpha centralization means that the network is more decentralized. *This indicator reflects the degree of hierarchy or equality/degree of dominance or balance in the project network structure.*
- **Betweenness centralization** (BZ): This measures how often a task lies on the shortest path between two other tasks. A high betweenness centralization means that there are some tasks that act as bridges or gatekeepers between many other tasks, while a low betweenness centralization means that the tasks are more direct connect to each other. *This indicator reflects the degree of intermediation that some tasks have in the project network, and it may be correlated with the free slack.*
- *Centralization* (Z): The centralization of a network is given by

$$Z = \frac{\sum_{i \in V} (\max_{j \in V} C_j - C_i)}{(n-1)(n-2)} \tag{52}$$

where V is the set of tasks, n is the number of tasks, and C_i and C_j are the given centralities of tasks i and j, respectively. The closeness/harmonic value PageRank/betweenness centralization can be calculated via the following formula. Therefore, the centrality values are formulated as follows:

- **Closeness centrality** (CC)/**Harmonic centrality** (HC): The closeness and harmonic centrality of a task i in a network is given by

$$CC_i = \frac{n-1}{\sum_{j\in V} d(i,j)} \tag{53}$$

$$HC_i = \sum_{j\in V} \frac{1}{d(i,j)} \tag{54}$$

where V is the set of tasks, n is the number of tasks, and $d(i,j)$ is the shortest path distance between tasks i and j.

- **Eigenvector centrality** (EC) / **PageRank centrality** (PRC): The eigenvector centrality/PageRank centrality of a task i in a network is given by

$$EC_i = \frac{1}{\lambda} \sum j \in V lij EC_j \tag{55}$$

$$PRC_i = \alpha \sum j \in V \frac{ljiPRC_j}{k_j^{out}} + \frac{1-\alpha}{n} \tag{56}$$

where V is the set of tasks, l_{ij} is the element (edge) from the logic domain (adjacency matrix), and λ is the largest eigenvalue of the adjacency matrix, k_j^{out} is the out-degree of task j, and α is the damping factor.

- **Alpha centrality** (AC), **Power centrality** (PC): The alpha/power centrality of a task i in a network is given by

$$AC_i(\alpha) = (\mathbf{I} - \alpha\mathbf{A})^{-1} x_i \tag{57}$$

$$PC_i(\alpha,\beta) = \alpha(\mathbf{I} - \beta\mathbf{A})^{-1}\mathbf{A} \tag{58}$$

where I is the identity matrix, α, β are the attenuation factors, $\mathbf{A} = \mathbf{LD}$ is the adjacency matrix, and x_i is the exogenous source of task i.

- **Betweenness centrality** (BC): The betweenness centrality of a task i in a network is given by

$$BC_i^{bet} = \sum_{j,k\in V} \frac{\sigma_{jk}(i)}{\sigma_{jk}} \tag{59}$$

where V is the set of tasks, σ_{jk} is the number of shortest paths between tasks j and k, and $\sigma_{jk}(i)$ is the number of shortest paths between tasks j and k that pass through task i.

- **Edge-betweenness centralization** (BEZ): This measures how often an edge lies on the shortest path between two tasks. A high edge-betweenness centralization means that there are several edges that act as bridges or bottlenecks between many tasks, while a low edge-betweenness centralization means that the edges are more redundant or dispensable. *This indicator reflects the degree of vulnerability or robustness of the project network.* The edge-betweenness centrality of a dependency (i,j) in a network is given by

$$BEC_{ij} = \sum_{k,l\in V} \frac{\sigma_{kl}(i,j)}{\sigma_{kl}} \tag{60}$$

where V is the set of tasks, σ_{kl} is the number of shortest paths between tasks k and l, and $\sigma_{kl}(i,j)$ is the number of shortest paths between tasks k and l that pass through dependency (i,j). The edge-betweenness centralization of a network is given by

$$BEZ = \frac{\sum_{i,j}(BEC_{\max} - BEC_{ij})}{(n-1)(n-2)(n-3)} \tag{61}$$

where V is the set of tasks, n is the number of tasks, and BEC_{ij} and BEC_{kl} are the edge betweenness centralities of dependencies (i,j) and (k,l), respectively.

- **Resilience for random** (RRes)/**systematic attack** (SRes): This measures how well the network can maintain its connectivity when a fraction of tasks are randomly removed/high degree tasks are removed. High resilience for random/systematic attack means that the network is more robust to random/systematic failures (task exclusions), while a low resilience for random/systematic attack means that the project network is more susceptible to random/systematic excludes, and in this way, it can be easily collapsed into components. *This indicator reflects the degree of robustness or susceptibility of the project network.* Resilience to random/systematic attack by a network is given by

$$RRes = \frac{S_r}{S_0} \tag{62}$$

$$SRes = \frac{S_s}{S_0} \tag{63}$$

where S_r, S_s are the sizes of the largest connected components after removing a fraction r and s of tasks, and S_0 is the size of the largest connected component before any removal. Fraction r represents the set of randomly selected tasks and s represents the set of high-degree tasks.

- **Average local efficiency** (ALE) / **Global efficiency** (GE): ALE measures the average efficiency of the subgraphs induced by the dependencies of each task in the project network, whereas the GE measures the average efficiency of the network as a whole. The efficiency of a pair of tasks is the multiplicative inverse of the shortest path distance between them. A high average local/global efficiency means that the network is more resilient to local failures (nonexecution of isolated tasks)/failure (nonexecution of highly dependent tasks), whereas a low average local/global efficiency means that the project network is more vulnerable to local/global failures. *This indicator reflects the degree of redundancy or fragility of the project network.* The similarity between these methods is that they both use the inverse of the distances between tasks to measure their efficiency. The difference is that the average local efficiency considers only the dependencies of each task, whereas the global efficiency considers the entire network. Therefore, the average local efficiency captures the local properties of the project network, while global efficiency captures the global properties of the project network. The average local/global efficiency of a network is given by

$$ALE = \frac{1}{n} \sum_{i\in V} LE_i \tag{64}$$

$$GE = \frac{1}{n(n-1)} \sum_{i,j\in V} \frac{1}{d(i,j)} \tag{65}$$

where V is the set of tasks, n is the number of tasks, and $LE_i = \frac{1}{k_i(k_i-1)} \sum_{j,h\in V_i} \frac{1}{d(j,h)}$ is the local efficiency of task i, and $d(i,j)$ is the shortest path distance between tasks i and j. V_i is the neighborhood of task i, k_i is the number of tasks in the neighborhood of task i, and $d(j,h)$ is the shortest path distance between tasks j and h in the subgraph induced by the neighborhood of task i.

- **Graph transitivity** (Tra): This measures the ratio of triangles to triplets in a project network. A triangle is a set of three tasks that are all connected to each other by dependencies, whereas a triplet is a set of three tasks that are connected by at least two dependencies. Graph transitivity is calculated by dividing the number of triangles by the number of triplets in the network. A high graph transitivity means that the network has a high clustering coefficient, while a low graph transitivity means that the network has a low clustering coefficient. A high graph transitivity means that the network has a clear community structure, with tasks that are densely connected to each other within groups and are sparsely connected to each other between groups. This can indicate that the network has a cohesive or modular structure, with tasks that have strong dependencies or influences within groups and weak dependencies or influences between groups. A low graph transitivity means that the project network is more homogeneous or random, with tasks that are equally connected to each other, regardless of the group. This can indicate that the network has a simple or uniform structure, with tasks that have similar dependencies or influences across the network. *This*

indicator reflects the degree of cliquishness or openness of the project network. The graph transitivity of a network is given by

$$Tra = \frac{3 \times \text{number of triangles in the network}}{\text{number of connected triples of tasks in the network}} \quad (66)$$

where a triangle is a closed triplet and a connected triple is a set of three tasks that are connected by at least two dependencies.

Appendix B. Employed goodness of fits and accuracy measures

Formal mathematical descriptions of the employed goodness-of-fit and accuracy measures:

Accuracy: Accuracy is defined as the proportion of correctly classified instances in a classification problem. Mathematically, it is given by:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (67)$$

where TP (true positives) are correctly predicted positive instances; TN (True Negatives) are correctly predicted negative instances; FP (false positives) are incorrectly predicted positive instances; FN (false negatives) are incorrectly predicted negative instances.

- **Significance of accuracy**: The significance of accuracy is evaluated via the statistical test **McNemar's test** to determine whether the observed accuracy is significantly different from a baseline or random chance. A common approach is to compute the **p-value** associated with the accuracy score.
- **F1 score**: The F1 score is the harmonic mean of precision and recall, balancing both measures. It is defined as:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (68)$$

where: $\text{Precision} = \frac{TP}{TP+FP}$, $\text{Recall} = \frac{TP}{TP+FN}$

- **R-Squared** (R^2): The coefficient of determination, R^2, measures the proportion of variance in the dependent variable that is predictable from the independent variables. It is given by:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \quad (69)$$

where: y_i are the actual values; $\hat{y}_i$ are the predicted values; $\bar{y}$ is the mean of the actual values.

- **Eta-Squared** (η^2): Eta-squared is a measure of effect size used in ANOVA to quantify the proportion of variance explained by a factor. It is defined as:

$$\eta^2 = \frac{SS_{\text{effect}}}{SS_{\text{total}}} \quad (70)$$

where: SS_{effect} is the sum of squares for the effect; SS_{total} is the total sum of squares.

- **Average silhouette score**: The silhouette score measures the quality of clustering by comparing intra-cluster cohesion and inter-cluster separation. The silhouette coefficient for a single data point i is:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (71)$$

where: $a(i)$ is the average distance between i and other points in the same cluster; $b(i)$ is the average distance between i and points in the nearest neighboring cluster. The **average silhouette score** is computed as the mean of $s(i)$ over all the data points.

Data availability

Data and functions are freely available and it can be downloaded from the CRAN websites. The applied packages and data are listed in Section 2. Both the entire article and all analysis is implemented by R in RStudio.

References

[1] C. Denizer, D. Kaufmann, A. Kraay, Good countries or good projects? Macro- and microcorrelates of world bank project performance, J. Dev. Econ. 105 (2013) 288–302, http://dx.doi.org/10.1016/j.jdeveco.2013.06.003.

[2] S. Hartmann, D. Briskorn, An updated survey of variants and extensions of the resource-constrained project scheduling problem, European J. Oper. Res. 297 (1) (2022) 1–14.

[3] D.M. Franco-Duran, J.M.d.l. Garza, Review of resource-constrained scheduling algorithms, J. Constr. Eng. Manag. 145 (11) (2019) 03119006, http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0001698.

[4] F.F. Boctor, Heuristics for scheduling projects with resource restrictions and several resource-duration modes, Int. J. Prod. Res. 31 (11) (1993) 2547–2558, http://dx.doi.org/10.1080/00207549308956882.

[5] J. Batselier, M. Vanhoucke, Construction and evaluation framework for a real-life project database, Int. J. Proj. Manage. 33 (3) (2015) 697–710, http://dx.doi.org/10.1016/j.ijproman.2014.09.004.

[6] A. Sprecher, R. Kolisch, PSPLIB-a project scheduling problem library, European J. Oper. Res. 96 (1996) 205–216, http://dx.doi.org/10.1016/S0377-2217(96)00170-1.

[7] T.R. Browning, A.A. Yassine, A random generator of resource-constrained multi-project network problems, J. Sched. 13 (2) (2010) 143–161, http://dx.doi.org/10.1007/s10951-009-0131-y.

[8] J.H. Patterson, A comparison of exact approaches for solving the multiple constrained resource, project scheduling problem, Manag. Sci. 30 (7) (1984) 854–867.

[9] V.V. Peteghem, M. Vanhoucke, An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances, European J. Oper. Res. 235 (1) (2014) 62–72, http://dx.doi.org/10.1016/j.ejor.2013.10.012.

[10] Z.T. Kosztyán, G. Novák, R. Jakab, I. Szalkai, C. Hegedűs, A matrix-based flexible project-planning library and indicators, Expert Syst. Appl. 216 (2023) 119472, http://dx.doi.org/10.1016/j.eswa.2022.119472.

[11] Z.T. Kosztyán, MFPP: Matrix-based flexible project planning, SoftwareX 17 (2022) 100973, http://dx.doi.org/10.1016/j.softx.2022.100973.

[12] Z.T. Kosztyán, A. Saghir, mfpp: an R package for matrix-based project planning, 2023, http://dx.doi.org/10.24433/CO.2915306.v1, Version 0.0.5, https://cran.r-project.org/web/packages/mfpp/index.html.

[13] M. Vanhoucke, J. Coelho, J. Batselier, An overview of project data for integrated project management and control, J. Mod. Proj. Manag. 3 (3) (2016) 6–21.

[14] T.R. Browning, A.A. Yassine, Resource-constrained multiproject scheduling: Priority rule performance revisited, Int. J. Prod. Econ. 126 (2) (2010) 212–228, http://dx.doi.org/10.1016/j.ijpe.2010.03.009.

[15] R. Van Eynde, M. Vanhoucke, Resource-constrained multi-project scheduling: benchmark datasets and decoupled scheduling, J. Sched. 23 (3) (2020) 301–325, http://dx.doi.org/10.1007/s10951-020-00651-w.

[16] R. Van Eynde, M. Vanhoucke, New summary measures and datasets for the multi-project scheduling problem, European J. Oper. Res. 299 (3) (2022) 853–868, http://dx.doi.org/10.1016/j.ejor.2021.10.006, URL: https://www.sciencedirect.com/science/article/pii/S0377221721008511.

[17] R. Van Eynde, M. Vanhoucke, J. Coelho, On summary measures for the resource-constrained project scheduling problem, Ann. Oper. Res. (2023) 1–33, http://dx.doi.org/10.1007/s10479-023-05470-8.

[18] J. Allaire, Y. Xie, C. Dervieux, R Foundation, H. Wickham, Journal of Statistical Software, R. Vaidyanathan, Association for Computing Machinery, C. Boettiger, Elsevier, K. Broman, K. Mueller, B. Quast, R. Pruim, B. Marwick, C. Wickham, O. Keyes, M. Yu, D. Emaasit, T. Onkelinx, A. Gasparini, M.-A. Desautels, D. Leutnant, MDPI, Taylor and Francis, O. Öğreden, D. Hance, D. Nüst, P. Uvesten, E. Campitelli, J. Muschelli, A. Hayes, Z.N. Kamvar, N. Ross, R. Cannoodt, D. Luguern, D.M. Kaplan, S. Kreutzer, S. Wang, J. Hesselberth, R. Hyndman, Articles: Article formats for r markdown, 2023, URL: https://CRAN.R-project.org/package=rticles, R package version 0.25.

[19] H. Wickham, J. Bryan, Readxl: Read excel files, 2023, URL: https://CRAN.R-project.org/package=readxl, R package version 1.4.3.

[20] Y. Xie, Xfun: Supporting functions for packages maintained by 'Yihui Xie', 2023, URL: https://CRAN.R-project.org/package=xfun, R package version 0.40.

[21] H. Wickham, Stringr: Simple, consistent wrappers for common string operations, 2022, URL: https://CRAN.R-project.org/package=stringr, R package version 1.5.0.

[22] A. Signorell, DescTools: Tools for descriptive statistics, 2023, URL: https://CRAN.R-project.org/package=DescTools, R package version 0.99.50.

[23] J. Fox, S. Weisberg, An r companion to applied regression, third ed., Sage, Thousand Oaks CA, 2019, URL: https://socialsciences.mcmaster.ca/jfox/Books/Companion/.

[24] Z.T. Kosztyán, A.I. Katona, M.T. Kurbucz, Z. Lantos, Generalized network-based dimensionality analysis, Expert Syst. Appl. 238 (Part A) (2024) 121779, http://dx.doi.org/10.1016/j.eswa.2023.121779.

[25] W.N. Venables, B.D. Ripley, Modern applied statistics with s, Fourth, Springer, New York, 2002, URL: https://www.stats.ox.ac.uk/pub/MASS4/, ISBN 0-387-95457-0.

[26] M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, K. Hornik, Cluster: Cluster analysis basics and extensions, 2022, URL: https://CRAN.R-project.org/package=cluster, R package version 2.1.3 — For new features, see the 'Changelog' file (in the package source).
[27] M.B. Kursa, W.R. Rudnicki, Feature selection with the Boruta package, J. Stat. Softw. 36 (11) (2010) 1–13, http://dx.doi.org/10.18637/jss.v036.i11.
[28] A. Liaw, M. Wiener, Classification and regression by randomforest, R News 2 (3) (2002) 18–22, URL: https://CRAN.R-project.org/doc/Rnews/.
[29] M. Kuhn, Caret: Classification and regression training, 2022, URL: https://CRAN.R-project.org/package=caret, R package version 6.0-93.
[30] Y. Xie, Knitr: A comprehensive tool for reproducible research in r, in: V. Stodden, F. Leisch, R.D. Peng (Eds.), Implementing Reproducible Computational Research, Chapman and Hall/CRC, 2014, ISBN 978-1466561595.
[31] H. Zhu, Kableextra: Construct complex table with 'kable' and pipe syntax, 2021, URL: https://CRAN.R-project.org/package=kableExtra, R package version 1.3.4.
[32] H. Wickham, D. Seidel, Scales: Scale functions for visualization, 2022, URL: https://CRAN.R-project.org/package=scales, R package version 1.2.1.
[33] H. Wickham, ggplot2: Elegant graphics for data analysis, Springer-Verlag New York, 2016, URL: https://ggplot2.tidyverse.org.
[34] K. Slowikowski, Ggrepel: Automatically position non-overlapping text labels with 'ggplot2', 2023, URL: https://CRAN.R-project.org/package=ggrepel, R package version 0.9.3.
[35] C. Sievert, Interactive web-based data visualization with r, plotly, and shiny, Chapman and Hall/CRC, 2020, URL: https://plotly-r.com.
[36] R. Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2022, URL: https://www.R-project.org/.
[37] R. Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2022, URL: https://www.R-project.org/.
[38] G. Csárdi, T. Nepusz, V. Traag, S. Horvát, F. Zanini, D. Noom, K. Müller, igraph: Network analysis and visualization in R, 2024, http://dx.doi.org/10.5281/zenodo.7682609, URL: https://CRAN.R-project.org/package=igraph, R package version 1.5.1.
[39] C.G. Watson, Braingraph: Graph theory analysis of brain MRI data, 2020, URL: https://CRAN.R-project.org/package=brainGraph, R package version 3.0.0.
[40] M. Danilovic, T.R. Browning, Managing complex product development projects with design structure matrices and domain mapping matrices, Int. J. Proj. Manage. 25 (3) (2007) 300–314, http://dx.doi.org/10.1016/j.ijproman.2006.11.003, URL: http://www.sciencedirect.com/science/article/pii/S0263786306001645.
[41] Z.T. Kosztyán, G.L. Novák, Compound matrix-based project database (CMPD), Sci. Data 11 (1) (2024) 319, http://dx.doi.org/10.1038/s41597-024-03154-x.
[42] D. Ciric, B. Lalic, D. Gracanin, N. Tasic, M. Delic, N. Medic, Agile vs. Traditional approach in project management: Strategies, challenges and reasons to introduce agile, Procedia Manuf. 39 (2019) 1407–1414, http://dx.doi.org/10.1016/j.promfg.2020.01.314.
[43] Z.T. Kosztyán, Exact algorithm for matrix-based project planning problems, Expert Syst. Appl. 42 (9) (2015) 4460–4473, http://dx.doi.org/10.1016/j.eswa.2015.01.066.
[44] F. Saberi-Movahed, K. Berahman, R. Sheikhpour, Y. Li, S. Pan, Nonnegative matrix factorization in dimensionality reduction: A survey, 2024, http://dx.doi.org/10.48550/arXiv.2405.03615, arXiv preprint arXiv:2405.03615.
[45] Z.T. Kosztyán, M.T. Kurbucz, A.I. Katona, Network-based dimensionality reduction of high-dimensional, low-sample-size datasets, Knowl.-Based Syst. (2022) 109180, http://dx.doi.org/10.1016/j.knosys.2022.109180.
[46] K. Berahmand, F. Saberi-Movahed, R. Sheikhpour, Y. Li, M. Jalili, A comprehensive survey on spectral clustering with graph structure learning, 2025, http://dx.doi.org/10.48550/arXiv.2501.13597, arXiv preprint arXiv:2501.13597.
[47] M. Jacomy, T. Venturini, S. Heymann, M. Bastian, ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software, Plos One 9 (6) (2014) 1–12, http://dx.doi.org/10.1371/journal.pone.0098679.
[48] J.H. Patterson, Project scheduling: The effects of problem structure on heuristic performance, Nav. Res. Logist. Q. 23 (1) (1976) 95–123, http://dx.doi.org/10.1002/nav.3800230110.
[49] I. Kurtulus, E. Davis, Multiproject scheduling: Categorization of heuristic rules performance, Manag. Sci. 28 (2) (1982) 161–172, http://dx.doi.org/10.1287/mnsc.28.2.161.
[50] V.A. Traag, L. Waltman, N.J. van Eck, From louvain to leiden: guaranteeing well-connected communities, Sci. Rep. 9 (1) (2019) 5233, http://dx.doi.org/10.1038/s41598-019-41695-z.
[51] M. Rosvall, C.T. Bergstrom, Maps of random walks on complex networks reveal community structure, Proc. Natl. Acad. Sci. 105 (4) (2008) 1118–1123, http://dx.doi.org/10.1073/pnas.0706851105.
[52] J. Reichardt, S. Bornholdt, When are networks truly modular? Phys. D: Nonlinear Phenom. 224 (1–2) (2006) 20–26, http://dx.doi.org/10.1016/j.physd.2006.09.009.