**CCC 2025** **Proceedings of the Creative Construction Conference** (2025)
*Edited by: Miroslaw J. Skibniewski & Miklós Hajdu & Žiga Turk*
*https://doi.org/10.22260/CCC2025/0015*

# SIMPLIFICATION AND OPTIMIZATION OF PROJECT STRUCTURES BASED ON PRECEDENCE-CONSTRAINED KNAPSACK PROBLEM

Gergely Lülök, Artúr Vlaszov, **Zoltán Sebestyén**
*Budapest University of Technology and Economics, Budapest, Hungary*

**Abstract**

The COVID-19 pandemic has highlighted the structural vulnerability of construction projects, especially in crises when the effectiveness of traditional scheduling approaches has significantly decreased. This paper presents a novel algorithmic approach that combines graph-based project structure simplification with cost-constrained profit maximization based on the mathematical model of the precedence-constrained knapsack problem. The method first reduces the size of the decision space by removing non-revenue-generating projects based on rules while preserving project logic dependencies and cost. After that, two optimization methods – integer linear programming and graph theory heuristics – are used to select the project. Based on practical examples, simplification significantly reduces the need for computation, making it possible to find optimal solutions that were not manageable on the original structures. The algorithm prioritizes project paths that provide the highest specific gains while also meeting precedent requirements. The analysis confirmed the model's flexibility and applicability in different project environments. The presented approach thus contributes to the development of crisis-resistant project management and, at the same time, offers a basis for the development of a generalizable decision support system.

**Keywords:** algorithm, construction, crisis management, project scheduling



**Peer-review under responsibility of the scientific committee of the Creative Construction Conference 2025.**

## 1. Introduction

As a result of the pandemic and other unexpected crises, construction projects worldwide have suffered structural disruptions that have fundamentally challenged the effectiveness of traditional project management and scheduling practices. Material shortages, labor losses, financial uncertainty, and the disruption of interdependent professional activities have shown that existing models are limited in handling decision-making situations arising from a rapidly changing environment. Although there have been many scheduling and rescheduling approaches in the literature in recent years, most of them are either overly static or unable to cope with the optimization of complex project structures – especially when it comes to balancing multiple goals (e.g., cost, time, strategic priorities) at the same time.

Previous approaches often focus solely on the scheduling side and ignore opportunities such as structural simplification of the project graph, which could significantly reduce the computational load. In addition, methods that support the choice between alternatives for project continuation in a uniform, multidimensional decision-making framework are rare.

In response to these challenges, the present study presents a new approach based on graph-based analysis and simplification of project structures, combining this with the principle of profit maximization with cost limitations. Our model is based on the mathematical foundations of the precedence-constrained knapsack problem. The surplus of the model lies in the fact that it offers a flexible, transparent decision support system that can integrate and harmonize various goals in an adaptable environment.

*Corresponding author email address: sebestyen.zoltan@gtk.bme.hu*

Thus, our study outlines a theoretical concept and develops an algorithmic solution that measurably reduces the computational need for optimization by structurally simplifying project graphs. In this way, it can generate new, crisis-resistant project structures that contribute to the development of modern crisis management practices and increase the flexibility of project management.

## 2. Literature review

The COVID-19 pandemic has severely affected construction projects worldwide, disrupting business as usual. Even in the early stages of the pandemic, transportation problems, material shortages, labor shortages, and productivity disruptions were experienced in the United States and Jordan, causing project cancellations and delays [1], [2]. In Slovakia, profitability declined, and inventory turnaround times increased [3], [4]. In India and Saudi Arabia, labor shortages and supply chain disruptions were the most severe [5], [6], while in the UAE, project management failures exacerbated the situation [7], [8]. In China, digital technology and state governance have facilitated a relatively rapid recovery [9], while in South Africa, unpreparedness and economic downturn caused lasting disruption [10]. Overall, the epidemic has posed new challenges to construction projects not only at the operational level but also at the strategic and organizational levels.

The above cases highlight the lack of reliable and proactive crisis management models in the construction industry. Hazaa et al. (2021) identified eight key factors, including strategic foresight and IT integration, on which our present approach builds [11]. Most previous models provide only a theoretical framework [12], while our algorithm can be applied to restructure project networks concretely. Loosemore and Teo (2012) found that Australian firms tend to respond reactively to crises, while our model allows for active, adaptive restructuring. In COVID-19 [13], Iqbal et al. (2021) proposed a three-stage operational recovery framework [14], and Murthi et al. (2023) emphasized the importance of restarting, but these are mainly static approaches [15]. In contrast, our study proposes a dynamic, optimization-based framework. The three-phase model of Sfakianaki et al. (2015) was designed for an economic recession [16], while our approach effectively integrates in- and post-crisis reengineering in one step.

Rescheduling and optimization approaches have evolved significantly in recent years, especially in addressing crises and uncertain project environments. Cui et al. (2010) demonstrated two rescheduling methods after buffering critical chain scheduling [17], where global rescheduling proved more effective. García-Mata et al. (2015) pointed toward predictive-reactive strategies and robust, knowledge-based decision support, highlighting the need to address uncertainty in complex industrial environments [18]. Compared to these, the novelty of our study lies in the fact that it connects graph-based project representation with a multi-dimensional decision space. The proactive scheduling models of Brčić et al. (2019) and Brčić & Mlinarić (2018) strengthen adaptivity by integrating flexibility and forecasting, but they are mainly based on resource constraints and static metrics [19], [20]. An et al. (2017) presented an effective heuristic focusing on cost constraints, but they do not integrate the non-financial dimensions of project success as our approach does [21]. Our study will also focus on the crisis situation, the multi-project environment, and the parallel running disciplines. The classic CPM-based rescheduling [22] and the multi-project, agent-based model [23] are also important advancements. Still, the novelty of our model lies in the fact that it embeds adaptive rescheduling into a transparent decision support system.

## 3. Research goals, objectives and limitations

In a complex, multi-sector investment project, the tight sequencing and parallel running of activities were subject to strict cost constraints and limited implementation capacity. The structural complexity and resource constraints required rapid and informed re-planning. The aim was to select a mix of project activities that maximized the financial results that could be achieved within the cost constraints while respecting the structure defined by the logical dependencies.

This research aims to develop an algorithmic approach that reduces the size of the decision space by simplifying the project graph structure, thus enabling efficient, cost-constrained project selection. The resulting schedule provides a financially advantageous implementation trajectory.

The research seeks to answer the following questions:

Q1: How can we simplify the project structure while avoiding information loss?

Q2: To what extent does simplification increase the computational efficiency of optimization?

Q3: What project combinations does the algorithm produce in a cost-constrained environment?

Q4: How does the performance of integer linear programming compare with that of the Samphaiboon-Yamada algorithm for the same input?

## 4. Methodology

In the project structure examined, each project has a positive cost, a non-negative payment, and an implementation sequence. We aim to choose a project combination that yields maximum profit for a given cost constraint. This decision problem is equivalent to the precedence-constrained knapsack problem [24], a sequential extension of the classical knapsack problem [25]. A directed acyclic graph (DAG) describes the relationship between the projects. The acyclic property of the graph ensures that the topological order of the projects is unique and contradiction free. Although the problem formulation is straightforward, its solution is NP-hard [26]; that is, the computational demand increases dramatically for large numbers of projects. Therefore, simplification of the input is necessary.

Non-paying projects are iteratively removed from the graph if their costs can be transferred unambiguously to their direct descendant. In this approach, paying projects are always retained, while non-paying ones with only one descendant are deleted and their cost is added to their descendant. Non-paying projects with multiple descendants are preserved to avoid contradictions. However, since the simplification rules are applied iteratively, some new simplifications may become possible after earlier modifications. Due to this, the required number of iterations cannot be determined in advance. Hence, further refinement of the algorithm is necessary. The complete solution process consists of four steps:

1. creating a directed graph from the data table
2. simplifying the graph according to the above rules
3. finding the optimal solution on the simplified graph using two algorithms
4. derivation of the solution back to the original graph by including the necessary non-paying projects

Steps 1 and 4 are simple, while steps 2 and 3 provide the computational kernel of the model.

## 5. Results and Discussion

### *5.1. Current simplification algorithm*

The goal is to develop a generalized algorithm that would return the simplest possible form of a general directed acyclic graph (DAG) input. Our current version works efficiently on graphs built from non-branching project branches starting from a single initial vertex – a standard structure in practice. The algorithm traverses the branches in topological order. At each step, it records the last vertex to be retained, the vertices candidate for deletion and their total cost, and a dictionary of incoming edges from other branches. In operation, the algorithm handles different types of vertices. If a non-paying vertex has only one descendant on the same branch, it is deleted, and the descendant bears its cost. If multiple edges from other branches come into the vertex, redundant edges are discarded, always keeping the last relevant edge information. This logic is illustrated in Figure 1, where an edge leads to the vertex B of branch one from the vertices D and E of branch 2. Since E already implies the completion of D, the algorithm keeps the E→B relation and directs the edge binding to the vertex C.

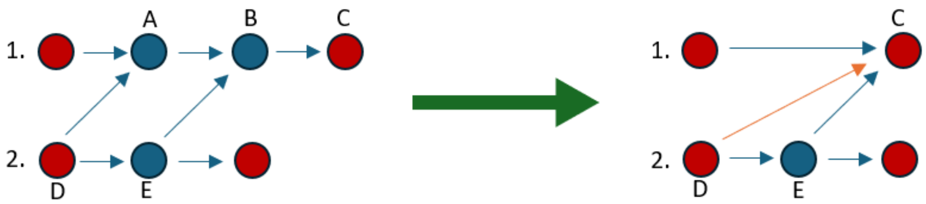


*Fig. 1. Removing redundant edges and edge traversal when simplifying a project graph*

If a non-paying vertex has a single descendant, but on a different branch, the cost is transferred to that descendant, and the edges are re-bound accordingly. If there are multiple descendants or a positive payment, the vertex cannot be simplified, so we keep it. A non-paying vertex can be deleted if it has no paying descendant.

It is important to note that simplifications can only be decided with knowledge of the current graph structure. A vertex may become deletable only after a subsequent edge is deleted. Therefore, the algorithm is run twice: first in the original order of the main branches and then in the reverse order. In our experience, this double run is sufficient, but in the long term, the goal is to develop an algorithm that can be generalized for all such cases.

*5.2. Optimisation problem-solving algorithms*

Once the simplifying algorithm has been used to make the project structure as simple as possible, algorithms can be applied to find the optimal project choice. We have chosen two types of solvers to compare their efficiency. In the first case, we formulate the problem as an integer linear programming problem and give it as input to a solving algorithm. In the second case, we apply Samphaiboon and Yamada's algorithm algorithm with a simplified graph as input [24].

We can write down the above integer LP with concrete values based on the simplified graph. There are several methods for solving this type of optimization problem, but unfortunately, in terms of algorithm theory, this problem is also NP-complete. This means that, in practice, the input size can only be increased up to a certain value as the computation time becomes too high beyond that. For our research, we chose the SCIP solver [27], [28], more specifically, the pySCIPOpt implementation [29].

The graph-based algorithm's conditions include the fact that a vertex's profit cannot be negative. When there remains a non-paying vertex in the graph, this condition is not satisfied clearly. We do not know if there is a Python implementation of the algorithm, so our goal is to program it based on pseudocode and apply it to our graphs. If the algorithm does not run in the presence of negative profit vertices, we only compare it to SCIP in cases where no such vertices are present.

*5.3. Observations/results*

SCIP proved effective on the simplified graph but failed to run on the original project graph—the original graphs contained about twice as many vertices and edges, significantly increasing the number of conditions. In addition, a further problem may be that the original graphs have a very large number of vertices with zero payoff, and the algorithm must consider too many redundant cases when searching for a solution.

## 6. Conclusions

This study aimed to develop an algorithmic model that supports cost-constrained decision-making and optimized project selection in crises by simplifying project structures. The practical functioning of the methodological elements was evaluated along the research questions Q1-Q4. The answer to the first question demonstrates that the structural simplification achieved by selectively removing non-paying vertices and perpetuating their costs does not compromise information integrity.

The simplification significantly reduced the computational demand in answer to the second question: the linear programming approach (SCIP) did not run efficiently on the original graph. At the same time, it provided fast and robust solutions on the simplified structure.

The third question shows that the algorithm identified a subgraph with high profit where precedence constraints are fulfilled rather than selecting the most profitable paths. Our tests showed that the integer LP model provided feasible and efficient solutions on the simplified project graphs. The implementation of the Samphaiboon–Yamada algorithm is still in progress, and its performance will be subject to comparison in future research. Overall, the paper provides a solution that is not only an optimization but also a pre-processing, which enhances the crisis resilience of project management from a practical point of view. Future goals are to generalize the method to more complex graphs and fully implement the graph-theoretic algorithm.

## References


[1]. Alsharef, S. Banerjee, S. M. J. Uddin, A. Albert, and E. Jaselskis, “Early Impacts of the COVID-19 Pandemic on the United States Construction Industry,” International Journal of Environmental Research and Public Health, vol. 18, no. 4, p. 1559, Feb. 2021, doi: 10.3390/ijerph18041559.

[2]. K. A.-D. Bsisu, “The Impact of COVID-19 Pandemic on Jordanian Civil Engineers and Construction Industry,” International Journal of Engineering Research and Technology, vol. 13, no. 5, p. 828, May 2020, doi: 10.37624/ijert/13.5.2020.828-830.

[3]. D. Gajdosikova, K. Valaskova, T. Kliestik, and V. Machova, “COVID-19 Pandemic and Its Impact on Challenges in the Construction Sector: A Case Study of Slovak Enterprises,” Mathematics, vol. 10, no. 17, p. 3130, Sep. 2022, doi: 10.3390/math10173130.

[4]. Stehlíková, M. Taušová, and K. Čulková, “Impact of the COVID-19 Pandemic on the Economic Development of the Mining and Construction Industry: Case Study in Slovakia,” Economies, vol. 12, no. 5, p. 119, May 2024, doi: 10.3390/economies12050119.

[5]. H. A. Rani, A. M. Farouk, K. S. Anandh, S. Almutairi, and R. A. Rahman, “Impact of COVID-19 on Construction Projects: The Case of India,” Buildings, vol. 12, no. 6, p. 762, Jun. 2022, doi: 10.3390/buildings12060762.

[6]. S. Almutairi, M. Bakri, A. A. AlMunifi, M. Algahtany, and S. Aldalbahy, “The Status of the Saudi Construction Industry during the COVID-19 Pandemic,” Sustainability, vol. 15, no. 21, p. 15432, Oct. 2023, doi: 10.3390/su152115432.

[7]. H. Alajmani, S. Ahmed, and S. M. El-Sayegh, “Factors causing delays in the UAE construction industry amid the Covid-19 pandemic,” Journal of Financial Management of Property and Construction, vol. 29, no. 1, pp. 135–151, Jul. 2023, doi: 10.1108/jfmpc-02-2023-0006.

[8]. M. Sami Ur Rehman, M. T. Shafiq, and M. Afzal, “Impact of COVID-19 on project performance in the UAE construction industry,” Journal of Engineering, Design and Technology, vol. 20, no. 1, pp. 245–266, Jun. 2021, doi: 10.1108/jedt-12-2020-0481.

[9]. Z. Wang, X. Cai, and Z. Liu, “Survival and Revival: Transition Path of the Chinese Construction Industry During the COVID-19 Pandemic,” Engineering Management Journal, vol. 35, no. 4, pp. 333–345, Aug. 2022, doi: 10.1080/10429247.2022.2108670.

[10].O. Aigbavboa, D. O. Aghimien, W. D. Thwala, and M. N. Ngozwana, “Unprepared industry meet pandemic: COVID-19 and the South Africa construction industry,” Journal of Engineering, Design and Technology, vol. 20, no. 1, pp. 183–200, Jul. 2021, doi: 10.1108/jedt-02-2021-0079.

[11].Y. M. H. Hazaa, F. A. Almaqtari, and A. Al-Swidi, “Factors Influencing Crisis Management: A systematic review and synthesis for future research,” Cogent Business & Management, vol. 8, no. 1, Jan. 2021, doi: 10.1080/23311975.2021.1878979.

[12].R. J, T. V, and M. N, “Role of Crisis Management in Construction Projects,” International Journal of Engineering & Technology, vol. 7, no. 2.12, p. 451, Apr. 2018, doi: 10.14419/ijet.v7i2.12.11515.

[13].M. Loosemore and M. M. M. Teo, “The Crisis Management Practices of Australian Construction Companies,” Construction Economics and Building, vol. 2, no. 2, pp. 15–26, Nov. 2012, doi: 10.5130/ajceb.v2i2.2897.

[14].M. Iqbal, N. Ahmad, M. Waqas, and M. Abrar, “COVID-19 pandemic and construction industry: Impacts, emerging construction safety practices, and proposed crisis management,” Brazilian Journal of Operations & Production Management, vol. 18, no. 2, pp. 1–17, Jun. 2021, doi: 10.14488/bjopm.2021.034.

[15].P. Murthi, K. Poongodi, and V. Mahesh, “Crisis Management Due to Covid-19 in Indian Construction Industry - An Overview,” Advances in Construction Materials and Management, pp. 107–115, 2023, doi: 10.1007/978-981-99-2552-0_9.

[16].Sfakianaki, T. Iliadis, and E. Zafeiris, “Crisis management under an economic recession in construction: the Greek case,” International Journal of Management and Decision Making, vol. 14, no. 4, p. 373, 2015, doi: 10.1504/ijmdm.2015.074015.

[17].N. Cui, W. Tian, and L. Bie, “Rescheduling after inserting the buffer in the critical chain scheduling,” 2010 International Conference on Logistics Systems and Intelligent Management (ICLSIM), pp. 1105–1110, Jan. 2010, doi: 10.1109/iclsim.2010.5461129.

[18].L. García-Mata, P. R. Márquez-Gutiérrez, and Larysa Burtseva, “Rescheduling in Industrial Environments: Emerging Technologies and Forthcoming Trends,” International Journal of Combinatorial Optimization Problems and Informatics, vol. 6, no. 3, pp. 34–48, Nov. 2015.

[19].M. Brčić, M. Katić, and N. Hlupić, “Planning horizons based proactive rescheduling for stochastic resource-constrained project scheduling problems,” European Journal of Operational Research, vol. 273, no. 1, pp. 58–66, Feb. 2019, doi: 10.1016/j.ejor.2018.07.037.

[20].M. Brčić and D. Mlinarić, “Tracking Predictive Gantt Chart for Proactive Rescheduling in Stochastic Resource Constrained Project Scheduling,” Journal of information and organizational sciences, vol. 42, no. 2, pp. 179–192, Dec. 2018, doi: 10.31341/jios.42.2.2.

[21].S.-M. An, S. Woo, C.-S. Cho, and S. Lee, “Development of budget-constrained rescheduling method in mega construction project,” KSCE Journal of Civil Engineering, vol. 21, no. 1, pp. 85–93, Jan. 2017, doi: 10.1007/s12205-016-0966-7.

[22].Zakia and D. Febrianti, “The Critical Path Method in Construction Project Rescheduling,” IOP Conference Series: Earth and Environmental Science, vol. 832, no. 1, p. 012009, Jul. 2021, doi: 10.1088/1755-1315/832/1/012009.

[23].L. Tosselli, V. Bogado, and E. Martínez, “A repeated-negotiation game approach to distributed (re)scheduling of multiple projects using decoupled learning,” Simulation Modelling Practice and Theory, vol. 98, p. 101980, Jan. 2020, doi: 10.1016/j.simpat.2019.101980.
[24].N. Samphaiboon and Y. Yamada, “Heuristic and Exact Algorithms for the Precedence-Constrained Knapsack Problem,” Journal of Optimization Theory and Applications, vol. 105, no. 3, pp. 659–676, Jun. 2000, doi: 10.1023/a:1004649425222.
[25].Pisinger and P. Toth, “Knapsack Problems,” Handbook of Combinatorial Optimization, pp. 299–428, 1998, doi: 10.1007/978-1-4613-0303-9_5.
[26].R. M. Karp, “Reducibility among Combinatorial Problems,” Complexity of Computer Computations, pp. 85–103, 1972, doi: 10.1007/978-1-4684-2001-2_9.
[27].T. Achterberg, “SCIP: solving constraint integer programs,” Mathematical Programming Computation, vol. 1, no. 1, pp. 1–41, Jan. 2009, doi: 10.1007/s12532-008-0001-1.
[28].T. Achterberg, T. Berthold, T. Koch, and K. Wolter, “Constraint Integer Programming: A New Approach to Integrate CP and MIP,” Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, pp. 6–20, doi: 10.1007/978-3-540-68155-7_4.
[29].S. Maher, M. Miltenberger, J. P. Pedroso, D. Rehfeldt, R. Schwarz, and F. Serrano, “PySCIPOpt: Mathematical Programming in Python with the SCIP Optimization Suite,” Mathematical Software – ICMS 2016, pp. 301–307, 2016, doi: 10.1007/978-3-319-42432-3_37.