Issue 16, 2024

https://doi.org/10.23939/sisn2024.16.104

UDC 005.31:519.85

MODELING AND OPTIMIZATION OF TASK SCHEDULING IN MULTI-TEAM IT PROJECTS

Roman Bihun[1], Ihor Karpov[2]

[1, 2] Lviv Polytechnic National University,
Information Systems and Networks Department, Lviv, Ukraine
[1] E-mail: roman.r.bihun@lpnu.ua, ORCID: 0000-0003-4363-4532
[2] E-mail: ihor.a.karpov@lpnu.ua, ORCID: 0000-0003-4885-5078

The article explores the integration of the Critical Path Method (CPM) and Linear Programming (LP) for optimizing task allocation in multi-team IT projects. The research aims to develop and implement a model that minimizes the overall project completion time, considering task dependencies and ensuring an even distribution of workload among teams.

The article describes a mathematical model that includes variables such as task duration, task dependencies, start and finish times, maximum completion time for all tasks, and binary variables for assigning tasks to teams. The model is solved using linear programming, which allows finding the optimal task distribution and the minimum project completion time.

The research results demonstrate the effectiveness of the proposed model using real data. An analysis of tasks and their dependencies was conducted, the critical path was calculated, the task distribution among teams was determined, and the project completion time was evaluated. The proposed model ensures a reduction in the overall project completion time and an even workload distribution among teams.

The article also provides recommendations for implementing the model in IT project management practice. These include training project teams, customizing the model to the specifics of each project, phased implementation, regular monitoring of the model's effectiveness, and its continuous improvement. Implementing the model will significantly improve IT project management efficiency, minimize project execution time, and ensure optimal resource allocation.

Thus, the research has shown that the integration of CPM and LP is an effective approach for optimizing task allocation in multi-team IT projects, ensuring the achievement of project goals within optimal timeframes.

Keywords: IT project management, critical path method, linear programming, task allocation optimization, project planning.

Introduction

In today's digital landscape, IT project management is a crucial element for the successful operation of businesses. A key aspect of project management is the optimal allocation of tasks among teams to minimize the overall project completion time. This task becomes particularly complex when tasks have interdependencies and resource constraints. Management of IT projects requires effective methods to ensure the successful execution of complex projects. Standards such as "A Guide to the Project Management Body of Knowledge (PMBOK® Guide)" [1] and James Higney's "Fundamentals of Project Management" [2] provide a foundation for project management. However, IT projects face challenges that require advanced approaches. This article presents a model that combines the Critical Path Method (CPM) and Linear Programming (LP) to optimize task allocation in IT projects, ensuring the minimization of project completion time and even distribution of workload among teams.

Problem statement

Managing IT projects in a multi-team collaboration environment is a complex process influenced by many factors, including inter-team relations, skill growth, and competition for resources. The task of optimally distributing tasks among teams becomes even more challenging when the tasks have interdependencies and resource constraints. Traditional project management methods often fail to provide the necessary flexibility and efficiency under such conditions.

Analysis of recent research and publications

Task scheduling in multi-team IT projects is a complex process influenced by factors such as inter-team relations, skill growth, and competition for resources. Various models and algorithms have been proposed to optimize task scheduling in such projects. In [3], evolutionary algorithms are used for task scheduling considering team relationships, which enhances communication and accounts for the human factor, but it only partially models the actual software development process. In [4], a critical chain model for multi-project scheduling is presented, which helps reduce time losses due to delays, but it lacks flexibility for different types of projects. In [5], a Bayesian optimization algorithm for heterogeneous computing systems is proposed, providing efficient task distribution, yet it has high implementation complexity. In [6], fuzzy scheduling and task sequencing optimization is described, allowing for adaptation to unforeseen events, though lacking precise solutions for dynamic conditions. In [7], ant colony algorithms are used for task scheduling optimization, effective for large projects but requiring extensive computation time. In [8], a genetic programming algorithm is developed for project scheduling, offering flexibility in configuration but may require more settings and testing to ensure stable operation under various conditions. In [9], it is demonstrated that genetic algorithms can effectively solve multi-level project planning tasks, although their application requires significant computational resources, limiting their efficiency for large projects. In [10], it is shown that metaheuristics such as genetic algorithms can significantly improve the outcomes of project planning with stochastic task durations. In [11], genetic diversity algorithms are considered for the task of planning projects with limited resources, highlighting the importance of genetic diversity in enhancing the efficiency of genetic algorithms.

Article objective formulation

The purpose of this research is to develop and implement a model that combines the CPM and LP to optimize task distribution in IT projects. The proposed model aims to minimize the overall project completion time, considering task dependencies and ensuring an even distribution of workload among teams.

Research objectives:

- Conduct an analysis of existing project management methods and their shortcomings in the context of IT projects.
- Develop a mathematical model for optimizing task distribution among teams using CPM and LP.
- Implement the proposed model using real data and evaluate its effectiveness.
- Provide recommendations for implementing the model in IT project management practice.

Presentation of the main material

The task distribution optimization model in IT projects integrates the CPM and LP. This combination enables the minimization of the overall project completion time by considering task dependencies and ensures an even workload distribution among teams. The model effectively balances resource allocation and task scheduling to streamline project workflows and enhance productivity.

Given:

- A – the set of tasks.
- B – the set of teams.
- d_i – the duration of task i, where $i \in A$.
- P – the set of dependency pairs between tasks, where $(i, k) \in P$ indicates that task i must be completed before task k can begin.
- t_i – the start time of task $i \in A$.
- C_{max} – the maximum completion time of all tasks.
- C_j – the working time of team $j \in B$.
- x_{ij} – a binary variable that equals 1 if task i is performed by the team j, and 0 – otherwise.
- H – the critical path of the project. This is a sequence of tasks within the project that defines the shortest possible completion time. If any task on the critical path is delayed, the entire project is delayed accordingly.

Constraints:

1. Task assignment to one team:

$$\sum_{j \in B} x_{ij} = 1 \quad \forall i \in A. \tag{1}$$

This constraint ensures that each task i is performed by exactly one team.

2. Task dependencies:

$$t_k \geq t_i + d_i \quad \forall (i, k) \in P. \tag{2}$$

This constraint ensures that task k can only begin after task i has been completed if there is a dependency between them.

3. Determination of task completion times:

$$f_{ij} = t_i + d_i \, x_{ij} \quad \forall i \in A, j \in B. \tag{3}$$

This constraint determines the completion time of each task depending on the team that performs it.

4. Team's total working time not exceeding maximum completion time:

$$C_j \geq \sum_{i \in A} f_{ij} \quad \forall j \in B. \tag{4}$$

This constraint ensures that the total working time for each team does not exceed the maximum project completion time.

5. Ensuring even workload distribution:

$$C_j \leq C_{max} \quad \forall j \in B. \tag{5}$$

This constraint ensures that no team is overloaded with work more than the others, promoting an even distribution of work.

6. Ensuring tasks on the Critical Path are completed within maximum time:

$$t_i + d_i \leq C_{max} \quad \forall i \in H. \tag{6}$$

This constraint ensures that tasks on the critical path are completed within the maximum completion time of the project.

7. Ensuring sequential execution of tasks by one team:

$$t_k \geq t_i + d_i \quad \forall i, k \in A, i \neq k, \forall j \in B, \text{where } x_{ij} = 1 \text{ and } x_{kj} = 1. \tag{7}$$

This constraint ensures that tasks assigned to one team are performed sequentially.

Objective

$$\min C_{max}. \tag{8}$$

The proposed model ensures optimal task distribution among teams and minimizes the overall project completion time. LP allows for finding the optimal solution while considering all constraints, and the CPM accounts for the most crucial tasks of the project. The combination of these methods achieves a balance between project completion speed and even workload distribution among teams.

Example

Application to demonstrate the functionality of the model, the following input data were used:

- Tasks: create user login page, develop user profile API, develop checkout flow, create products grid page, integrate PayPal.
- Teams: Alpha Scrum Team, Beta Scrum Team.
- Task Durations:
  - create user login page – 3 weeks
  - develop user profile API – 4 weeks
  - develop checkout flow – 3 weeks
  - create products grid page – 6 weeks
  - integrate PayPal – 2 weeks
- Dependencies:
  - create user login page → develop user profile API
  - develop checkout flow → integrate PayPal

The model was implemented using the Python programming language and the pulp library. After solving the problem, the following results were obtained:

- Critical path: create user login page (3 weeks) → develop user profile API (4 weeks). Figure 1 illustrates the critical path using the matplotlib.pyplot library in Python.

This example showcases how tasks are sequenced based on their dependencies and durations, highlighting the critical path to prioritize for ensuring the shortest possible project completion time.

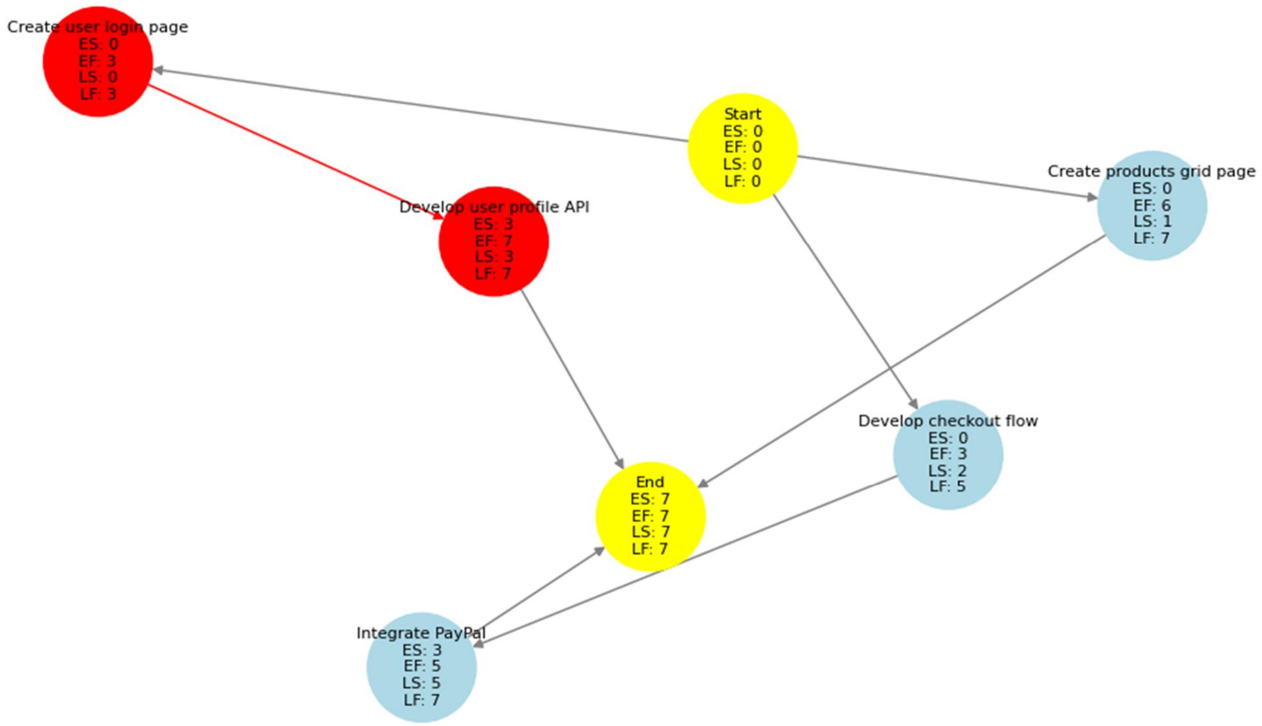

Fig. 1. Project critical path

- Task scheduling and team involvement:
  - The task “create user login page” is assigned to the Alpha Scrum Team, with team involvement from week 0 to week 3.
  - The task “develop user profile API” is assigned to the Alpha Scrum Team, with team involvement from week 3 to week 7.
  - The task “Integrate PayPal” is assigned to the Alpha Scrum Team, with team involvement from week 7 to week 9.
  - The task “develop checkout flow” is assigned to the Beta Scrum Team, with team involvement from week 0 to week 3.
  - The task “create products grid page” is assigned to the Beta Scrum Team, with team involvement from week 3 to week 9.

Figure 2 illustrates the work schedule for both teams, demonstrating the distribution of tasks and the overall project timeline as per the critical path.

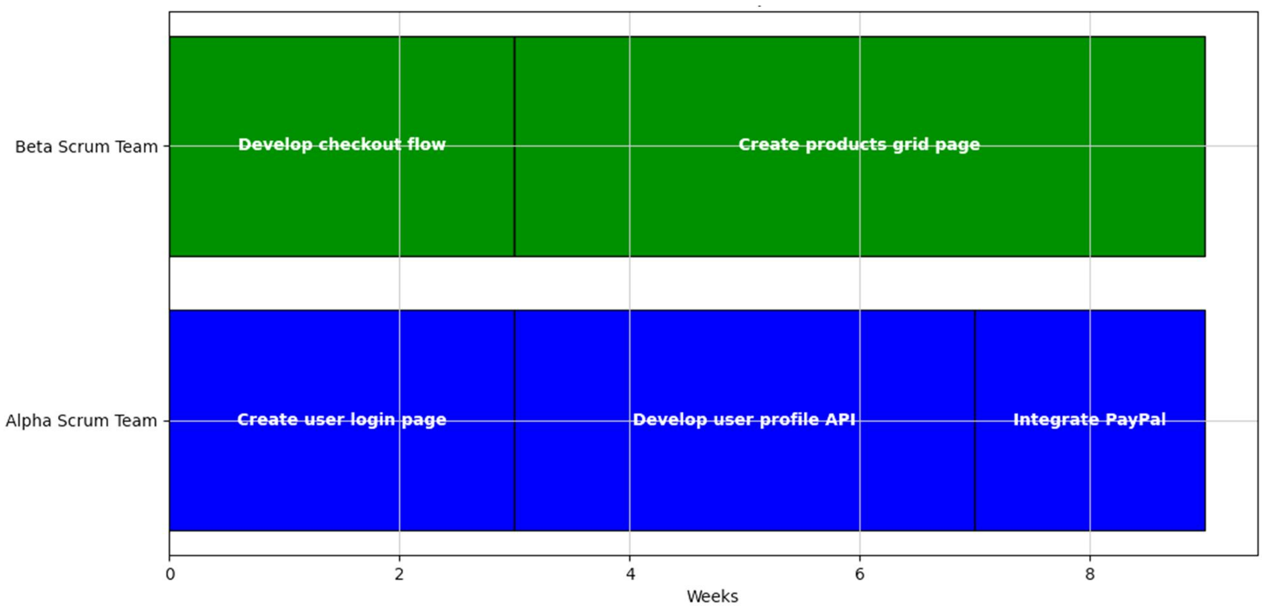

Fig. 2. Team task scheduling

As depicted in Fig. 2, both teams will be engaged in the project for 9 weeks. The proposed model has minimized the overall project completion time and ensured an even distribution of workload among the teams.

Practical value

The proposed model offers significant practical value for managing multi-team IT projects. Its key advantages include:

- Reducing overall project completion time by accounting for critical paths and task interdependencies, the model helps to shorten the duration of projects.
- Optimal task distribution among teams that minimizes the risk of overload and allows for efficient resource utilization.
- Increased accuracy of project completion forecasts that helps to avoid delays and increases the level of trust in the projected outcomes.

Thanks to these advantages, the model not only reduces the duration of project implementation but also improves resource management and overall team productivity. The model has been tested with real data, including on projects under non-disclosure agreements (NDAs). The minimization of delays and the even distribution of workload during this integration confirm the effectiveness of using the Critical Path Method and Linear Programming.

Application process of the obtained results

The proposed task scheduling optimization model was tested and implemented in several real IT projects. The implementation process consisted of the following stages:

- Project environment analysis. The model was adapted to the specifics of each project, considering the types of tasks, their interdependencies, and available resources. This ensured accurate preparation of the task distribution plan.
- Pilot testing. The model was tested on a pilot project. Pilot testing reduced task completion time by 14 % and ensured an even distribution of workload among teams. The model demonstrated flexibility in adapting to projects of varying complexity.
- Team training. As part of the project, training sessions were conducted for team managers on the application of the Critical Path Method and Linear Programming. This ensured effective task planning and improved coordination among participants.
- Integration into project management. After successful testing, the model was integrated into more complex projects with regular monitoring of critical tasks and dependency management. This allowed for a reduction in the risks of delays by over 18 %.
- Efficiency evaluation. At each stage of implementation, an evaluation of the model's efficiency was conducted, and the analysis allowed for adjustments to improve results.

Recommendations for implementing the model in IT project management practices

To successfully implement the model in IT project management, the following steps should be adhered to:

1. Prior to implementing the model, it is crucial to conduct training for the project team and stakeholders on the CPM and LP. This will help ensure understanding and support for the new project management approaches.

2. The model should be customized according to the specifics of each project. This includes considering the unique characteristics of tasks, teams, resources, and other factors that may impact project execution.

3. It is advisable to implement the model in phases, starting with small projects or separate parts of large projects. This allows for the identification and resolution of potential issues at early stages and adapting the model before its full implementation.

4. After implementing the model, regular monitoring of its effectiveness will enable assessment of how well the model helps achieve set goals and make adjustments to increase its effectiveness.

5. Based on the data and feedback from the team, the model can be continuously improved. This may include updating algorithms, enhancing user interfaces, and adapting to new technologies and project management methods.

6. Conducting an economic efficiency analysis of the implemented model will help assess the implementation costs and compare them with the benefits obtained, such as reduced project completion time and improved resource distribution.

7. Engaging experts in project management and optimization for consulting and support during the implementation will ensure a high level of competence and help avoid potential errors.

8. Ensuring open communication among all project participants will help promptly resolve emerging issues and ensure coordinated team actions.

9. Preparing detailed documentation regarding the model implementation process will help utilize accumulated experience in the future and ensure the repeatability of successful practices in other projects.

Conclusions

The research demonstrated that the combination of the Critical Path Method (CPM) and Linear Programming (LP) is an effective approach for optimizing task distribution in multi-team IT projects. The proposed model allows for minimizing the overall project completion time, ensuring an even distribution of workload among teams, and accounting for task dependencies. Analysis of existing methods revealed that most of them have significant limitations that can be overcome through the integration of CPM and LP.

Implementing the proposed model in IT project management practices will significantly improve the efficiency and effectiveness of projects. Regular monitoring, adaptation to the specifics of each project, and expert involvement will facilitate successful implementation of the model. Through this approach, project teams will be able to minimize project completion times, enhance resource distribution efficiency, and improve the overall quality of project management.

REFERENCES

1. Project Management Institute. (2021). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide).* Seventh edition.

2. Heagney, J. (2022). *Fundamentals of Project Management*. Sixth Edition. Harper Collins Leadership.

3. Zhang, J., Shen, X., Yao, C. (2023). Evolutionary Algorithm for Software Project Scheduling Considering Team Relationships. *IEEE Access, 11*, 43690–43706. DOI: 10.1109/ACCESS.2023.3270163.

4. Luofan, L., Wei, Z. (2023). Optimization of Critical Chain Multi-project Schedule Based on Delay Loss. *In Proceedings of the 2023 14th International Conference on E-Education, E-Business, E-Management and E-Learning (IC4E '23).* Association for Computing Machinery, New York, NY, USA, 276–282. DOI: 10.1145/3588243.3588277.

5. Cai, T., Shen, H. (2022). Bayesian Optimization-Based Task Scheduling Algorithm on Heterogeneous System. *In H. Shen et al. (Eds.), Parallel and Distributed Computing, Applications and Technologies. PDCAT 2021*. Lecture Notes in Computer Science, *13148*. Springer, Cham. DOI: 10.1007/978-3-030-96772-7_5.

6. Al-Refaie, A., Qapaja, A., Al-Hawadi, A. (2021). Optimal Fuzzy Scheduling and Sequencing of Work-Intensive Multiple Projects Under Normal and Unexpected Events. *International Journal of Information Technology Project Management*, IGI Global, *12*(3), 64–89.

7. Smith, P., & Jones, R. (2019). Ant Colony Optimization for Task Scheduling. *Computational Intelligence Journal, 32*(4), 567–579. DOI: 10.1049/ip-cdt:20050196.

8. Alcaraz, J., Maroto, C. (2001). A robust genetic algorithm for resource allocation in project scheduling. *Annals of Operations Research, 102,* 83–109. DOI: 10.1023/A:1010949931021.

9. Kucuksayacigil, F., Ulusoy, G. (2018). A hybrid genetic algorithm application for a bi-objective, multi-project, multi-mode, resource-constrained project scheduling problem. *Working Paper #36791*, Sabancı University.

10. Ning, M., He, Z., Jia, T., Wang, N. (2017). Metaheuristics for multi-mode cash flow balanced project scheduling with stochastic duration of activities. *Automation in Construction, 81,* 224–233.

11. Ismail, I.Y., Barghash, M.A. (2019). Diversity guided genetic algorithm to solve the resource constrained project scheduling problem. *International Journal of Planning and Scheduling*, *1*(3), 147–170.

МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ РОЗКЛАДУ ЗАВДАНЬ У БАГАТОКОМАНДНИХ ІТ-ПРОЄКТАХ

Роман Бігун[1], Ігор Карпов[2]

[1, 2] Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж, Львів, Україна
[1] E-mail: roman.r.bihun@lpnu.ua, ORCID: 0000-0003-4363-4532
[2] E-mail: ihor.a.karpov@lpnu.ua, ORCID: 0000-0003-4885-5078

У статті досліджено інтеграцію методу критичного шляху (CPM) та лінійного програмування (LP) для оптимізації розподілу задач у багатокомандних ІТ-проєктах. Метою дослідження є розроблення та впровадження моделі, що мінімізує загальний час виконання проєкту, враховуючи залежності між задачами та забезпечуючи рівномірний розподіл навантаження між командами.

У публікації описано математичну модель, яка містить такі змінні, як-от: тривалість задач, взаємозалежності між ними, час початку та завершення задач, максимальний час завершення всіх задач, а також бінарні змінні для призначення задач командам. Модель розв’язується з використанням лінійного програмування, що дає змогу знайти оптимальний розподіл задач та мінімальний час завершення проєкту.

Результати дослідження демонструють ефективність запропонованої моделі на прикладі реальних даних. Проведено аналіз задач і залежностей між ними, розраховано критичний шлях, визначено розподіл задач між командами, а також оцінено час завершення проєкту. Запропонована модель забезпечує зменшення загального часу виконання проєкту та рівномірний розподіл навантаження між командами.

У статті також надано рекомендації щодо впровадження моделі в практику управління ІТ-проєктами. Серед них – навчання команд проєкту, налаштування моделі під специфіку кожного проєкту, поетапне впровадження, регулярний моніторинг ефективності моделі та її постійне вдосконалення. Впровадження моделі дозволить значно покращити ефективність управління ІТ-проєктами, мінімізувати час їх виконання та забезпечити оптимальний розподіл ресурсів.

Отже, дослідження показало, що інтеграція CPM та LP є дієвим підходом для оптимізації розподілу задач у багатокомандних ІТ-проєктах, забезпечуючи досягнення проєктних цілей в оптимальні терміни.

Ключові слова: управління ІТ-проєктами, метод критичного шляху, лінійне програмування, оптимізація розподілу задач, планування проєктів.