Decision Making in Manufacturing and Services
Vol. 20 • 2026 • No. 1 • pp. 5–21

Task-Location Assignment Problem in Planning Horizon

Klaudia Krawiec[*], Roger Książek[**], Aleksander Woźniak[***], Weronika Łyszczan[****]

Abstract. Efficient task allocation in production environments is essential for optimizing resource utilization and minimizing operating costs. This paper proposes a Mixed-Integer Linear Programming (MILP) model for the Task-Location Assignment Problem in Planning Horizon, with the objective of minimizing total operating costs. Additionally, two heuristic algorithms – a greedy scheduling algorithm and a priority-based scheduling algorithm – were developed to generate feasible solutions quickly. Computational experiments were conducted using real-world production process data across various scenarios. The results indicate that the proposed MILP model achieves an average cost reduction of approximately 11% compared to the greedy scheduling algorithm and about 6% compared to the priority-based scheduling algorithm. The results demonstrate the practical value of mathematical optimization in reducing internal production supply costs while supporting the efficient allocation of production tasks to available locations.

Keywords: task-location allocation, mathematical programming, Mixed-Integer Linear Programming, heuristics, greedy scheduling algorithm, priority-based scheduling algorithm

Mathematics Subject Classification: 90B80, 90C11, 90B35

JEL Classification: C61, C44, M11

Submitted: April 7, 2026

Revised: June 21, 2026

[*] AGH University of Krakow, Faculty of Management, Krakow, Poland, e-mail: klakrawiec@agh.edu.pl, ORCID: 0009-0000-0343-4574.

[**] AGH University of Krakow, Faculty of Management, Krakow, Poland, e-mail: roger@agh.edu.pl, ORCID: 0009-0000-0343-4574.

[***] AGH University of Krakow, Faculty of Management, Krakow, Poland, e-mail: alekwozniak@student.agh.edu.pl, ORCID: 0009-0002-5130-0081.

[****] AGH University of Krakow, Faculty of Management, Krakow, Poland, e-mail: wlyszczan@agh.edu.pl, ORCID: 0009-0003-7581-9053.

DOI: https://doi.org/10.7494/dmms.2026.20.1.8153

1. INTRODUCTION

The efficiency of a production process depends on multiple factors, among which the effective allocation of tasks to available resources plays a crucial role. Although task allocation problems have been extensively studied in the fields of scheduling and optimization (e.g., Drexl & Haase, 1995; Harris, 1913; Kaczmarczyk, 2009; Książek et al., 2017; Pochet & Wolsey, 2006), the diversity of production environments and task characteristics means that many practical aspects remain insufficiently explored. Consequently, further research is needed to develop allocation methods that better reflect the specific operational requirements of modern manufacturing enterprises and support more effective production management.

The motivation for this study originates from a company specializing in the production of customized industrial equipment and machinery using robotic manufacturing technologies. The company faced significant operational challenges due to the intensive involvement of its components warehouse in projects to construct new production workstations for external customers. Such projects require employees to frequently move between production locations and the warehouse in order to collect materials and components necessary for task execution.

In this environment, the absence of an optimized task allocation strategy increases operating costs and results in inefficient use of available resources, including workforce time and production space. Ineffective allocation decisions may also disrupt shop-floor operations by creating resource conflicts, production area congestion, and excessive warehouse workload. These inefficiencies contribute to production delays, reduce the company's ability to meet delivery deadlines, and negatively affect overall operational performance.

To address these challenges, this paper introduces the Task-Location Assignment Problem in Planning Horizon (TLAPPH), which focuses on assigning production projects to available locations within a finite planning horizon while minimizing internal production supply costs. The proposed approach aims to improve the utilization of production space and support more efficient management of production processes.

The main research question (MRQ) addressed in this study is:

MRQ: What is the impact of optimization based on the TLAPPH on reducing operating costs and improving production process efficiency?

To answer this question, two detailed research questions (DRQs) were formulated:

DRQ 1: To what extent can operating costs be reduced and production efficiency improved using a Mixed-Integer Linear Programming (MILP) model for the TLAPPH?

DRQ 2: To what extent can operating costs be reduced and production efficiency improved using a priority-based scheduling algorithm for the TLAPPH?

The main contribution of this paper is the formulation of the Task-Location Assignment Problem in Planning Horizon and the development of a dedicated Mixed-Integer Linear Programming model that explicitly incorporates spatial constraints and internal production supply costs. The proposed model is applicable to manufacturing companies with similar operational characteristics and can be adapted to specific industrial requirements by including additional constraints. Furthermore, two heuristic approaches – a greedy scheduling algorithm and a priority-based scheduling algorithm – have been developed and compared with the MILP model. The computational study demonstrates that mathematical optimization can effectively reduce operating costs and provide practical decision-support tools for real-world production environments.

The remainder of the paper is organized as follows. Section 2 presents a review of the relevant literature on task allocation and scheduling problems. Section 3 introduces the proposed mathematical model and heuristic algorithms. Section 4 describes the computational experiments and discusses the obtained results. Finally, Section 5 summarizes the main findings and outlines directions for future research.

2. PAPER POSITIONING

The problem of allocating tasks to resources, particularly in production scheduling, has been widely studied using various optimization approaches, including mathematical programming, heuristic algorithms, and metaheuristics. Mathematical models based on combinatorial optimization play an important role in supporting efficient task assignment and generating high-quality schedules (e.g., Kapłan et al., 2023; Książek et al., 2022). Moreover, the increasing volume of data generated in manufacturing systems is driving the growing demand for advanced decision-support tools (e.g., Książek et al., 2019; Łyszczan et al., 2024). The following section reviews selected studies that are closely related to the problem addressed in this paper.

Task allocation problems in multi-agent systems and robotics have received considerable attention in the literature. Miloradovic et al. (2023) investigated task allocation for multi-agent systems in the context of robot mission planning. Their approach employed integer linear programming and constraint programming models to optimize task assignments. However, the proposed method does not consider sequencing constraints or project completion deadlines, which are important factors in industrial production environments. Similarly, Wilde and Alonso-Mora (2023) addressed the problem of selecting robots for tasks with different requirements while accounting for budget limitations. Their solution combined integer linear programming with a Large Neighborhood Search (LNS) heuristic. A similar perspective was adopted by Tihanyi et al. (2023), who examined task allocation in dynamic and uncertain environments, with a particular emphasis on safety considerations. Task allocation also plays an important role in robotic process automation and business process management research (Gabryelczyk et al., 2022; Di Ciccio et al., 2024; Köpke et al., 2023). Although these studies contribute significantly to task allocation optimization, they primarily focus on robotic systems and do not explicitly consider production-space constraints, project scheduling requirements, or internal logistics costs, which are critical in manufacturing environments.

Another stream of research focuses on task allocation in manufacturing and computing systems. Nguyen et al. (2023) developed a model to optimize worker-to-task assignments in apparel manufacturing, accounting for employee skills and labor costs.

However, their study did not address issues related to workstation allocation, internal logistics costs, or precedence relations between tasks. Similarly, Valentin et al. (2016) proposed an integer programming model for task allocation in heterogeneous real-time computing systems to minimize energy consumption. Although this approach relies on mathematical optimization, it does not account for production-space management or sequencing constraints. In a different application domain, Delaram and Valilai (2018) introduced a scheduling model for cloud manufacturing systems that balances workloads and minimizes task execution times. Unlike manufacturing environments, where physical space limitations and internal logistics play a crucial role, their model does not incorporate location-related constraints or project precedence requirements.

Heuristic and metaheuristic approaches constitute another important research direction. Fadhil (2024) analyzed five scheduling and resource allocation algorithms in dynamic computing environments and found that genetic algorithms and particle swarm optimization achieved the best performance in terms of task completion efficiency. However, the study did not consider production-specific factors such as internal logistics or spatial constraints. Similarly, Gao and Zhao (2022) proposed greedy and genetic algorithms for task allocation in crowdsourcing systems while incorporating user preferences. Although these approaches provide efficient solutions in dynamic environments, their direct application to manufacturing systems would require adaptations to account for production-specific constraints, particularly operation synchronization and internal transportation costs. In another study, Yu and Lee (2023) investigated route-planning methods for collaborative multi-UAV area search operations. Their proposed MUCS-GD (multi-UAV collaborative search algorithm based on the greedy algorithm) and MUCS-BSAE (multi-UAV collaborative search algorithm based on the binary search algorithm) improve the coordination of autonomous units. While such methods could be adapted to optimize internal transportation routes in production environments, additional modifications would be necessary to account for production floor layouts, transportation costs, and synchronization requirements.

The literature provides numerous examples of mathematical optimization, heuristic methods, and metaheuristic approaches applied to task allocation problems. Nevertheless, most existing studies do not simultaneously consider production-space management, internal logistics costs, planning-horizon constraints, and project precedence relations. This paper addresses this gap by introducing the Task-Location Assignment Problem in Planning Horizon and proposing a dedicated optimization framework that integrates task allocation decisions with production-space management to minimize total operating costs.

3. TASK-LOCATION ASSIGNMENT PROBLEM IN PLANNING HORIZON (TLAPPH)

The Task-Location Assignment Problem in Planning Horizon considered in this paper aims to assign production projects to available locations on the production floor while minimizing total internal production supply costs. The main challenge lies in simultaneously satisfying the project-specific requirements and spatial limitations of the production environment.

Several constraints must be taken into account when determining a feasible solution. First, some projects may be subject to precedence relations that require them to be executed in a specific order due to technological or organizational requirements. Second, each project must be assigned to exactly one location that provides sufficient production space. Third, internal production supply costs depend on the number of warehouse transitions required by each project and on the cost associated with the assigned location. Finally, all projects must be completed within the planning horizon considered by the company. Consequently, the scheduler must determine both the location and the start time for each project to minimize the total operating costs while satisfying all imposed constraints. The output of the MILP model developed for the TLAPPH, presented in Subsection 3.1, is a cost-minimizing allocation of projects to locations and start times that satisfies all model constraints.

A MILP model was developed to address the TLAPPH described above. The model incorporates constraints related to production-space availability, project completion within the planning horizon, precedence relations, and internal production supply costs. The model was verified and validated using real production data, and computational experiments were performed using parameters reflecting the operational conditions of the company under study. As the TLAPPH extends the classical assignment problem by incorporating project durations, location capacities, planning-horizon constraints, and precedence relations, the resulting optimization model may become computationally challenging for larger problem instances. Therefore, two dedicated heuristic algorithms were developed to complement the MILP approach and provide feasible solutions within short computation times. The first heuristic is a greedy scheduling algorithm that reflects the company's current task assignment practice. Projects are assigned iteratively by selecting the first feasible location associated with the lowest immediate transition cost, without considering broader cost implications. Although this approach provides solutions very quickly, it may lead to suboptimal assignments. To improve solution quality while maintaining short computation times, a priority-based scheduling algorithm was also developed. This approach introduces additional selection criteria, such as project size and the number of required warehouse transitions, enabling more informed assignment decisions and a more efficient utilization of available locations. The effectiveness of the three approaches, namely the MILP model, the greedy scheduling algorithm, and the priority-based scheduling algorithm, was evaluated through computational experiments. Particular attention was paid to the objective function values and computation times obtained for real-world production instances, thereby allowing the trade-off between solution quality and computational efficiency to be assessed.

3.1. Mixed-Integer Linear Programming model for the TLAPPH

The MILP model for the TLAPPH is defined as follows. Let T denote the planning horizon, P the set of production projects, and L the set of available locations on the production floor. Furthermore, let R denote the set of precedence relations between projects.

For each project $i \in P$, the following parameters are defined: v_i representing the project size; k_i, representing the project duration; and d_i, representing the number of required warehouse transitions during project execution. For each location $j \in L$, the parameters w_j and c_j denote the available space and the cost of moving between the location and the warehouse, respectively.

The decision problem is to determine the location and start time of each project while satisfying all imposed constraints.

The objective is to minimize the total internal production supply cost associated with warehouse transitions and project allocation decisions. The notation used in the MILP formulation is summarized in Table 1.

Table 1. *Notation used in the MILP model*

Symbol	Description
T	set of planning periods, $T = \{1, ..., n\}$
P	set of projects
L	set of available locations
R	set of project pairs (r, l) such that project r must be started no later than project l
c_j	cost of transitions from location j to the warehouse
w_j	size of location j
v_i	size of project i
k_i	duration of project i
d_i	number of required transitions for project i
α	auxiliary parameter controlling the impact of the time criterion on the cost function
M	large number
K	auxiliary variable
y_{ijt}	binary variable equal to 1 if project i starts at location j in period t, 0 otherwise

The MILP model for the TLAPPH is defined by objective function (1a) and constraints (1b)–(1j):

$$min : \sum_{t\in T}\sum_{i\in P}\sum_{j\in L} c_j \cdot d_i \cdot y_{ijt} + \alpha \sum_{t\in T}\sum_{i\in P}\sum_{j\in L} y_{ijt} \cdot t \tag{1a}$$

$$K = \sum_{t\in T}\sum_{i\in P}\sum_{j\in L} c_j \cdot d_i \cdot y_{ijt} \tag{1b}$$

$$\sum_{i\in P} y_{ijt} \le 1, t \in T, j \in L \tag{1c}$$

$$\sum_{j\in L}\sum_{t\in T} y_{ijt} = 1, i \in P \tag{1d}$$

$$M(1 - y_{ijt}) \ge \sum_{i_1 \in P} \sum_{t_1 \in T: t_1 > t \wedge t_1 < t + k_i} y_{ijt}, t \in T, j \in L, i \in P \tag{1e}$$

$$y_{ijt} \cdot t + (k_i - 1) \leq n, t \in T, j \in L, i \in P \tag{1f}$$

$$y_{ijt} \cdot v_i \leq w_j, t \in T, j \in L, i \in P \tag{1g}$$

$$\sum_{j \in L} \sum_{t \in T} y_{r,j,1} \cdot t \leq \sum_{j \in L} \sum_{t \in T} y_{l,j,1} \cdot t, \ (r,l) \in R \tag{1h}$$

$$y_{ijt} \in \{0,1\}, \ t \in T, \ i \in P, \ j \in L \tag{1i}$$

$$K \geq 0 \tag{1j}$$

Objective function (1a) minimizes internal production supply costs and comprises two components. The first component represents the operational costs incurred when moving workers to the warehouse at the appropriate times, while the second is a regularization term designed to control the shifting of projects over time. Minimizing this function yields an optimal solution that balances operational costs and ensures an even distribution of projects throughout the planning horizon.

The auxiliary decision variable K, as defined in constraint (1b), represents the total actual cost of moving between locations over the specified time horizon, incorporating both distance and associated costs. It serves as an auxiliary variable, enabling the analysis of the overall efficiency of the production plan without regularizing project shifts over time. The mathematical model will still function correctly even if constraint (1b) is omitted. Constraint (1c) limits the number of projects that can be started at a given location and period to one. This constraint is essential for preventing projects from overlapping at the same location during the same period. Constraint (1d) ensures that every project is initiated. This requirement is essential to guarantee that all projects are started within the planning horizon. Constraint (1e) ensures that if project i is initiated at location j in period t, it cannot be started in any overlapping periods during its duration. Constraint (1f) ensures that starting project i at location j in period t does not exceed the available space at that location. This constraint accounts for the physical limitations of space access at each location. Constraint (1g) ensures that project i starts early enough that, after accounting for its duration, it completes no later than the last period of the planning horizon. An additional constraint (1h) defines the time dependencies between projects r and l, requiring that project r be started no later than project l. This constraint establishes a logical order for project starts, enabling a planned start sequence, and serves as an extension of the problem. Without this constraint, optimal solutions would not account for potential dependencies in the required execution sequence of individual projects. While a strictly defined sequence of project execution is typically unnecessary in the company under study, the model has been expanded to accommodate such cases, allowing flexible incorporation when needed. Constraint (1i) ensures that the variable y is binary, while constraint (1j) ensures that the auxiliary variable K is non-negative.

3.2. Heuristic algorithms for the TLAPPH

To compare the practical applicability of the developed mathematical model with heuristic approaches, this paper introduces two algorithms: a greedy scheduling algorithm and a priority-based scheduling algorithm.

The greedy scheduling algorithm reflects the company's current project assignment procedure. It orders projects by the non-increasing number of required warehouse transitions, assuming that projects with higher operating costs should be scheduled first, unless specific temporal dependencies require otherwise. The algorithm also allows for additional time constraints, such as precedence relationships between projects. Each project is then assigned to the first feasible location on the production floor. The location is selected according to accessibility, with the objective of minimizing worker movement between production areas. The procedure is repeated iteratively until all projects have been assigned. This approach ensures very short computational time; however, it may lead to suboptimal solutions in more complex cases involving additional temporal dependencies.

The main steps of the greedy scheduling algorithm are as follows:

```
 INPUT:  Set of projects P, set of locations L.
     1:  SORT projects in non-increasing order of required transitions.
     2:  FOR each project i∈P:
     a:  SELECT the first feasible location j∈L.
     b:  ASSIGN project i to location j.
     3:  REPEAT until every project is assigned.
OUTPUT:  Project schedule.
```

The priority-based scheduling algorithm provides an alternative heuristic approach by introducing explicit priority rules for both projects and locations. First, projects are sorted in descending order of size, and in the case of ties, by decreasing number of required warehouse transitions. Similarly, locations are sorted in descending order of size and, for equal sizes, by increasing warehouse-access cost. Based on these characteristics, groups of projects and locations are formed. Each project group is matched with a corresponding set of feasible locations. The most cost-intensive projects (i.e., those with the highest number of warehouse transitions) are assigned to the least costly locations within each group. If capacity constraints are violated, assignments are adjusted by moving projects to earlier or later groups. Finally, projects within each group are allocated to the first available locations in order to minimize completion times and operational costs.

The main steps of the priority-based scheduling algorithm are as follows:

```
 INPUT:  Set of projects P, set of locations L.
     1:  SORT projects by size (descending).
     2:  IF sizes are equal, SORT by required transitions (descending).
     3:  SORT locations by size (descending).
     4:  IF sizes are equal, SORT by warehouse-access cost (ascending).
     5:  CREATE project and location groups.
     6:  ASSIGN projects to feasible locations within groups.
     7:  IF capacity limits are exceeded, ADJUST assignments by moving
         projects between groups.
     8:  GENERATE final schedule.
OUTPUT:  Project schedule.
```

The priority-based scheduling algorithm, together with the TLAPPH model, is intended to experimentally assess whether substantially better solutions can be obtained for the problem under consideration. A comparison of all three approaches enables evaluation of the potential benefits of their application and provides insight into the trade-off between solution quality and computational efficiency.

4. EXPERIMENTS

Ten scenarios were prepared for the computational experiments based on actual data provided by the company. All computational experiments were performed on a workstation equipped with an Intel® Core™ i7-4710HQ processor and 16 GB RAM. The developed MILP model was solved using the GLPK (GNU Linear Programming Kit) solver. During scenario development, a 14-day planning horizon was assumed, along with a fixed number of 15 projects and 10 available locations. The objective was to allocate all projects to production locations to minimize operating costs while ensuring they were completed within the specified planning horizon. Solutions were obtained using three methods: the greedy scheduling algorithm, the priority-based scheduling algorithm, and the TLAPPH model. The results from these methods were compared to evaluate their effectiveness in optimizing task allocation and reducing operating costs.

The parameter values for the projects in each dataset are presented in Table 2. The parameter values for the locations in each scenario are presented in Table 3. Each dataset is identified by a Roman numeral (I–X). In some scenarios, additional constraints related to time dependencies between selected pairs of projects were introduced. The notation (P_1, P_2) indicates that project P_1 should start no later than project P_2. The additional constraints were defined as follows: Scenario II: (P_3, P_2); Scenario III: (P_5, P_4); Scenario VIII: (P_{12}, P_2); and Scenario X: (P_{14}, P_4).

Table 2. *Projects' parameter values in each scenario*

Scenario	Parameter	Project														
		P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
I	d_i	66	73	52	84	25	77	28	39	33	48	93	62	99	80	44
	v_i	23	15	10	17	16	21	14	21	18	20	19	14	15	10	19
	k_i	5	8	10	4	6	3	8	6	6	6	7	8	10	6	8
II	d_i	73	19	81	25	67	38	60	48	95	92	59	79	64	52	33
	v_i	15	20	16	19	19	16	14	18	20	17	22	18	21	21	17
	k_i	6	7	7	8	8	6	7	6	7	6	7	6	6	8	8
III	d_i	97	71	104	58	69	110	44	90	49	112	82	108	86	101	78
	v_i	17	12	20	17	10	14	16	22	21	17	19	21	16	15	20
	k_i	4	5	5	6	3	2	5	6	4	3	5	6	7	4	6

Table 2 (cont.)

Scenario	Parameter	Project														
		P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
IV	d_i	146	136	134	123	93	140	110	131	129	115	106	86	98	145	113
	v_i	20	17	18	16	21	19	16	22	14	19	21	15	17	18	17
	k_i	5	6	5	7	6	5	4	7	4	6	5	6	7	5	6
V	d_i	84	46	27	68	58	96	74	60	45	39	92	77	35	59	28
	v_i	6	5	7	9	4	6	10	3	7	8	9	5	10	8	4
	k_i	4	4	5	6	4	3	5	6	4	7	4	5	6	7	5
VI	d_i	81	34	20	46	94	68	78	17	106	61	50	39	99	85	73
	v_i	7	5	7	6	10	8	9	4	8	7	10	6	9	8	9
	k_i	5	6	5	7	6	7	5	6	5	5	6	7	7	6	5
VII	d_i	100	134	169	127	109	156	186	178	203	149	193	170	115	122	141
	v_i	11	15	16	13	14	17	18	20	22	16	17	16	19	18	17
	k_i	5	4	6	5	7	6	5	4	6	7	5	7	6	5	5
VIII	d_i	10	31	63	84	55	24	78	46	92	107	44	79	22	16	70
	v_i	6	7	6	10	7	6	9	10	8	7	9	8	10	8	8
	k_i	6	7	8	7	6	5	8	5	7	6	5	8	5	6	7
IX	d_i	50	100	130	70	40	90	150	120	20	80	160	170	110	30	180
	v_i	20	25	24	21	22	27	30	23	20	26	29	27	28	25	23
	k_i	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
X	d_i	66	37	53	29	71	44	89	14	92	58	23	19	77	81	33
	v_i	24	22	26	27	25	21	29	22	28	26	28	27	28	26	29
	k_i	6	9	4	8	5	6	9	9	6	7	5	7	8	6	4

Table 3. *Available locations' parameter values in each scenario*

Scenario	Parameter	Location									
		L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}
I	w_j	16	14	20	10	18	14	24	16	21	17
	c_j	8	6	9	4	8	9	7	6	5	8
II	w_j	18	16	20	17	15	21	19	23	20	17
	c_j	6	7	10	5	5	7	6	8	9	9
III	w_j	13	12	16	19	15	21	17	22	18	20
	c_j	5	4	8	10	6	7	11	9	12	8

Table 3 (cont.)

IV	w_j	16	18	19	17	20	18	22	17	15	21
	c_j	6	6	5	6	7	6	9	7	5	8
V	w_j	10	6	7	5	9	8	9	10	4	7
	c_j	20	12	14	10	18	16	18	20	8	14
VI	w_j	9	7	6	8	10	9	5	8	10	6
	c_j	20	15	14	16	22	18	10	17	19	12
VII	w_j	20	16	14	17	18	16	19	22	17	16
	c_j	10	8	7	9	9	8	12	11	9	8
VIII	w_j	9	8	7	6	10	6	8	9	8	10
	c_j	8	5	4	7	6	9	4	5	10	7
IX	w_j	25	28	23	21	28	30	27	25	29	23
	c_j	7	9	6	8	7	8	7	5	6	5
X	w_j	27	24	21	29	26	28	27	23	25	29
	c_j	5	3	4	5	2	5	4	6	5	3

The results presented in Table 4 highlight differences in the objective function values and execution times among the three methods: the greedy scheduling algorithm, the priority-based scheduling algorithm, and the mathematical optimization model. The mathematical model, despite being limited to a maximum computation time of ten minutes, consistently produced better solutions than the heuristic algorithms. The GAP column further illustrates the difficulty of obtaining optimal solutions within the imposed time limit. In some scenarios, for example, Scenario X, the reported GAP reached 20%, indicating that further improvements might be achieved with additional computation time. Nevertheless, even in these cases, the mathematical model outperformed the heuristic approaches in minimizing operating costs.

In contrast, the greedy scheduling and priority-based scheduling algorithms enerated solutions within a few seconds. The priority-based scheduling algorithm's additional selection criteria ensured that it generally produced better allocations than the greedy scheduling algorithm. Although the heuristic approaches offered very short computation times and satisfactory results, the experiments confirmed that mathematical optimization can substantially reduce operating costs, demonstrating its practical value for industrial production planning.

An additional advantage of the proposed approach is its flexibility. Since the model is formulated as a Mixed-Integer Linear Programming problem, it can be extended to incorporate additional practical constraints, making it a promising framework for supporting more comprehensive production planning.

Table 4. *Objective function values and computation times for each method across all scenarios*

Scenario	Greedy scheduling algorithm		Priority-based scheduling algorithm		Mathematical model		
	objective	time [s]	objective	time [s]	objective	GAP [%]	time [s]
I	6,432	0.8	6,393	1.2	5,759	9.6	600
II	6,424	1.3	6,147	1.3	5,617	0.7	600
III	10,393	1.4	9,939	1.0	9,118	2.3	600
IV	11,889	0.8	11,640	1.4	11,272	0.0	600
V	14,034	0.9	12,554	0.8	12,250	0.0	600
VI	16,628	1.2	15,995	1.2	15,669	1.7	600
VII	20,842	0.8	20,003	1.2	19,773	0.0	600
VIII	5,222	1.2	4,954	1.0	4,140	9.9	600
IX	10,970	1.3	991	1.0	9,160	2.6	600
X	3,289	0.8	3,223	1.2	3,048	20.0	600

The differential indicators calculated for each dataset provide a quantitative measure of the deviation between the results obtained using the heuristic algorithms and the best solution identified by the mathematical model. These indicators, expressed as percentage differences, enable an assessment of the effectiveness of the heuristic approaches in minimizing the objective function. Table 5 summarizes the minimum, maximum, and average percentage differences obtained for both heuristic algorithms. The results indicate that the priority-based scheduling algorithm generally performs better than the greedy scheduling algorithm. However, both methods exhibit varying degrees of deviation from the solutions obtained using the mathematical model. The maximum percentage differences identify scenarios in which the heuristics struggled to generate competitive solutions, whereas the minimum values indicate cases in which the heuristic solutions were close to those produced by the mathematical model. The average percentage differences provide an overall assessment of the effectiveness of the heuristic approaches, confirming that while they are computationally efficient, they do not always achieve the lowest operating costs.

Table 5. *Average percentage deviation indicators for the analyzed scenarios*

Method	Percentage deviation indicators [%]		
	minimum	maximum	average
Greedy scheduling algorithm	5.41	26.14	12.54
Priority-based scheduling algorithm	1.16	19.66	7.20

The results indicate that the mathematical model effectively reduced operating costs at the company analyzed. On average, the model achieved a 10.86% cost reduction compared with the greedy scheduling algorithm and a 6.5% reduction compared with the priority-based scheduling algorithm. These improvements demonstrate the advantage of mathematical optimization over heuristic methods for minimizing production-related costs. Furthermore, the analyzed scenarios closely reflected real-world conditions, and the computation times for obtaining solutions were within an acceptable range. Notably, these times were comparable to those required by a planner to assign projects manually, reinforcing the practical applicability of the proposed model. These findings confirm that the mathematical model is not only effective from a theoretical perspective but also suitable for implementation in real manufacturing environments.

The distribution of projects on the production floor for Scenario V, obtained using the greedy scheduling algorithm, the priority-based scheduling algorithm, and the mathematical model, is presented in Figures 1–3, respectively. Figure 1 presents the project distribution generated by the greedy scheduling algorithm. This method has a very short execution time; however, its straightforward assignment procedure results in less efficient use of the available space than other approaches. Figure 2 presents the project distribution obtained using the priority-based scheduling algorithm. This approach incorporates additional criteria, such as location size preferences and the number of required warehouse transitions, leading to a more balanced allocation of projects and improved utilization of lower-cost locations. Figure 3 presents the project distribution generated by the mathematical model. The model seeks to minimize total operating costs while satisfying planning-horizon and space-availability constraints. Unlike the heuristic approaches, the model focuses exclusively on cost minimization, resulting in the most efficient use of available resources.

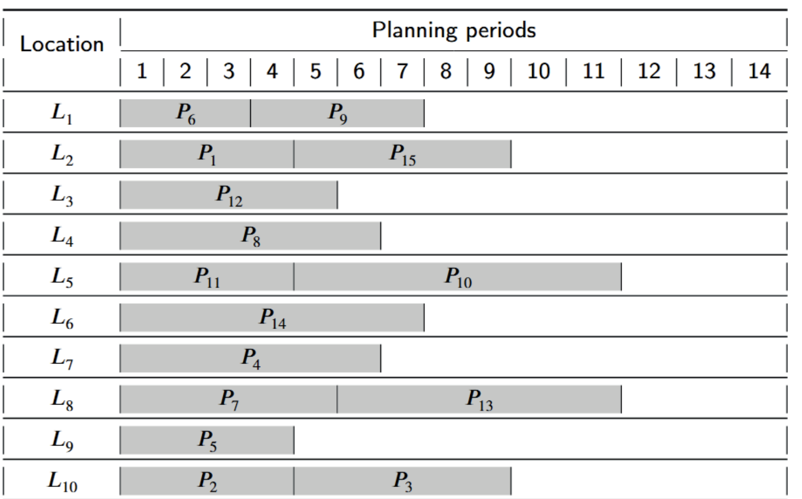

Fig. 1. *Allocation of projects in scenario V to available locations across the planning horizon using the greedy scheduling algorithm*

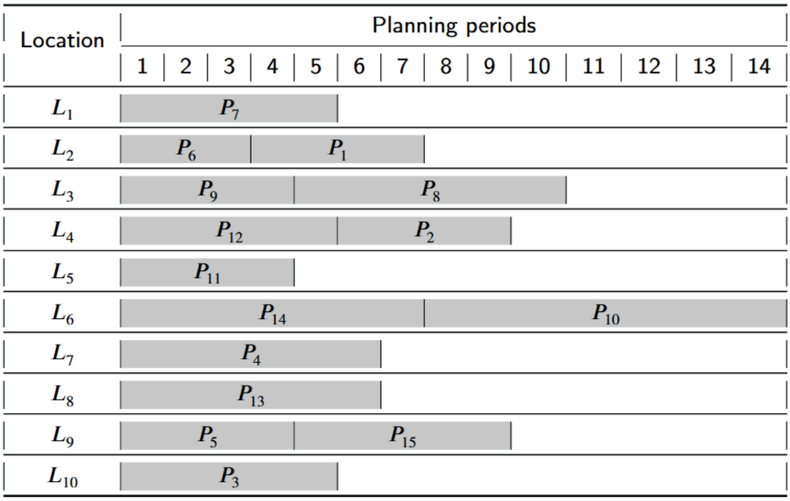

Fig. 2. *Allocation of projects in scenario V to available locations across the planning horizon using the priority-based scheduling algorithm*

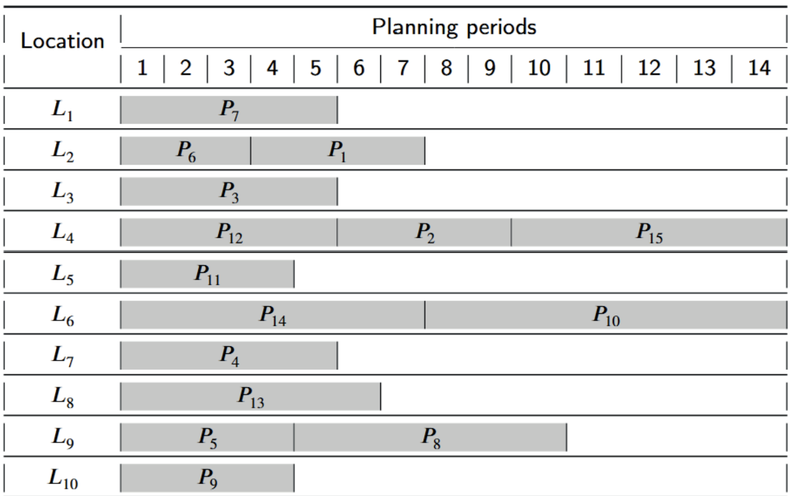

Fig. 3. *Allocation of projects in scenario V to available locations across the planning horizon using the mathematical model*

5. CONCLUSION

The analysis of the results from the conducted calculations confirmed the hypothesis outlined in the paper. A comparison between the outcomes obtained using the developed mathematical model and the heuristics (greedy and priority-based scheduling algorithms) revealed that advanced methods, such as integer programming, can substantially reduce operating costs for the analyzed company. The average cost reductions of 10.86% relative to the greedy scheduling algorithm and 6.5% relative to the priority-based scheduling algorithm underscore the significance of optimization in industrial contexts. The study demonstrated that while the mathematical model provides an optimal solution, the priority-based scheduling algorithm can serve as an effective alternative in situations requiring rapid decision-making. Its design incorporates key parameters while offering faster execution times than the full optimization approach.

However, it is important to note that the solutions developed were specifically tailored to the characteristics of the case studied, in which the primary requirement was to quickly obtain a practical solution that could be implemented within the company under analysis. While the methods employed in this study proved effective within this context, they are not the only possible strategies for addressing the problem. Alternative optimization approaches, such as metaheuristic algorithms (e.g., genetic algorithms or simulated annealing), are also explored in the literature and may prove to be effective for handling larger data sets or dynamically changing production conditions.

In summary, the solutions developed in this study offer companies valuable decision-support tools for managing production processes. The results demonstrate that implementing the optimization model not only reduces operating costs but also improves work organization and the overall efficiency of the production system. Future research could examine the model's scalability for longer planning horizons and explore the impact of dynamic changes in parameters, such as fluctuating costs or shifting project priorities, over the planning horizon.

ACKNOWLEDGMENTS

The research was partially financed by a subsidy and the program "Excellence Initiative – Research University" for the AGH University of Krakow.

REFERENCES

Delaram J. & Valilai O.F. (2018). A mathematical model for task scheduling in cloud manufacturing systems focusing on global logistics. *Procedia Manufacturing*, **17**, pp. 387–394. **DOI:** https://doi.org/10.1016/j.promfg.2018.10.061.

Di Ciccio C., Fdhila W., Agostinelli S., Amyot D., Leopold H., Krčál M., Malinova Mandelburger M., Polančič G., Tomičić-Pupek K., Gdowska K., Grisold T., Sliz P., Beerepoot I., Gabryelczyk R. & Plattfaut R. (Eds.) (2024). *Business Process Management: Blockchain, Robotic Process Automation, Central and Eastern European, Educators and Industry Forum: BPM 2024 Blockchain, RPA, CEE, Educators and Industry Forum, Krakow, Poland, September 1–6, 2024, Proceedings*. Lecture Notes in Business Information Processing, **527**, Cham: Springer. **DOI:** https://doi.org/10.1007/978-3-031-70445-1.

Drexl A. & Haase K. (1995). Proportional lotsizing and scheduling. *International Journal of Production Economics*, **40**(1), pp. 73–87. **DOI:** https://doi.org/10.1016/0925-5273(95)00040-U.

Fadhil H. (2024). Optimizing task scheduling and resource allocation in computing environments using metaheuristic methods. *Fusion: Practice and Applications*, **15**(1), pp. 157–179. **DOI:** https://doi.org/10.54216/FPA.150113.

Gabryelczyk R., Brzychczy E., Gdowska K. & Kluza K. (2022). Business process management in CEE countries: A literature-based research landscape. In: Marrella A., Matulevičius R., Gabryelczyk R., Axmann B., Vukšić V.B., Gaaloul W., Štemberger M.I., Kő A. & Lu Q. (Eds.). *Business Process Management: Blockchain, Robotic Process Automation, and Central and Eastern Europe Forum. BPM 2022*. Lecture Notes in Business Information Processing, **459**, pp. 279–294. Cham: Springer. **DOI:** https://doi.org/10.1007/978-3-031-16168-1_18.

Gao H. & Zhao H. (2022). A personalized task allocation strategy in mobile crowdsensing for minimizing total cost. *Sensors*, **22**(7), 2751. **DOI:** https://doi.org/10.3390/s22072751.

Harris F.W. (1913). How many parts to make at once. *Factory, The Magazine of Management*, **10**(2), pp. 135–136.

Kaczmarczyk W. (2009). Modelling multi-period set-up times in the proportional lotsizing problem. *Decision Making in Manufacturing and Services*, **3**(2), pp. 15–35. **DOI:** https://doi.org/10.7494/dmms.2009.3.2.15.

Kapłan R., Książek R., Gdowska K. & Łebkowski P. (2023). Scheduling lockdowns under conditions of pandemic uncertainty. *IEEE Access*, **11**, pp. 118689–118697. **DOI:** https://doi.org/10.1109/access.2023.3327101.

Köpke J., López-Pintado O., Plattfaut R., Rehse J.R., Gdowska K., Gonzalez-Lopez F., Munoz-Gama J., Smit K. & van der Werf J.M.E. (Eds.) (2023). *Business Process Management: Blockchain, Robotic Process Automation and Educators Forum: BPM 2023 Blockchain, RPA and Educators Forum, Utrecht, The Netherlands, September 11–15, 2023, Proceedings*. Lecture Notes in Business Information Processing, **491**. Cham: Springer. **DOI:** https://doi.org/10.1007/978-3-031-43433-4.

Książek R., Gdowska K. & Książek D. (2017). What have been developed for lotsizing and scheduling problems since the EOQ model was introduced. In: *CLC'2016 – Carpathian Logistics Congress, Zakopane, Poland, November 28–30, 2016: Conference Proceedings*, pp. 418–423. Ostrava: TANGER Ltd. **URL:** https://www.confer.cz/clc/2016?page=1 [25.05.2026].

Książek R., Gdowska K. & Kapłan R. (2019). Selected tools for supporting decision making in the era of big data. In: *Proceedings CLC 2018 – Carpathian Logistics Congress, Prague, Czech Republic, December 3–5, 2018*, pp. 527–532. Ostrava: TANGER Ltd. **URL:** https://www.confer.cz/clc/2016/2718-what-have-been-developed-for-lot-sizing-and-scheduling-problem-since-eoq-model-was-introduced [25.05.2026]

Książek R., Kapłan R., Gdowska K. & Łebkowski P. (2022). Vaccination schedule under conditions of limited vaccine production rate. *Vaccines*, **10**(1), 116. **DOI:** https://doi.org/10.3390/vaccines10010116.

Łyszczan W., Książek R. & Gdowska K. (2024). Enhancing production efficiency through computer simulation: A case study. In: *Tehnologìï, ìnstrumenti ta strategìï realìzacìï naukovih doslìdžen': Proceedings of the VIII International Scientific Conference, Rivne, Ukraine, August 30, 2024*, pp. 88–95. Vinnytsia: Ukrlogos Grup. **URL:** https://archives.mcnd.org.ua/index.php/conference-proceeding/article/view/148 [25.05.2026].

Miloradovic B., Çürüklü B., Ekström M. & Papadopoulos A.V. (2023). Optimizing parallel task execution for multi-agent mission planning. *IEEE Access*, **11**, 24367–24381. **DOI:** https://doi.org/10.1109/ACCESS.2023.3254900.

Nguyen V., Tran Q., Altarazi F. & Tran T. (2023). Binary integer formulations for task allocation and optimal labor cost in small and medium apparel manufacturing. *Designs*, **7**(4), 84. **DOI:** https://doi.org/10.3390/designs7040084.

Pochet Y. & Wolsey L.A. (2006). *Production Planning by Mixed Integer Programming.* New York: Springer. **DOI:** https://doi.org/10.1007/0-387-33477-7.

Tihanyi D., Lu Y., Karaca O. & Kamgarpour M. (2023). Multi-robot task allocation for safe planning against stochastic hazard dynamics. In: *Proceedings of the 2023 European Control Conference (ECC), Bucharest, Romania, June 13–16, 2023*, pp. 1–6. IEEE. **DOI:** https://doi.org/10.23919/ECC57647.2023.10178126.

Valentin E., de Freitas R. & Barreto R. (2016). Applying MGAP modeling to the hard real-time task allocation on multiple heterogeneous processors problem. *Procedia Computer Science*, **80**, pp. 1135–1146. **DOI:** https://doi.org/10.1016/j.procs.2016.05.425.

Wilde N. & Alonso-Mora J. (2023). Designing heterogeneous robot fleets for task allocation and sequencing In: *Proceedings of the International Symposium on Multi-Robot and Multi-Agent Systems (MRS), Boston, United States, December 4–5, 2023*, pp. 156–162. IEEE. **DOI:** https://doi.org/10.1109/MRS60187.2023.10416795.

Yu Y. & Lee S. (2023). Efficient multi-UAV path planning for collaborative area search operations. *Applied Sciences*, **13**(15), 8728. **DOI:** https://doi.org/10.3390/app13158728.