Performance Evaluation of MOSM Method on Resource-Constrained Multi-Objective Multi-Project Scheduling Problems

Krisztián Mihály, Gyula Kulcsár, and Mónika Kulcsárné-Forrai

***Abstract*—This paper presents an extended model to address complicated project scheduling challenges, which deals with diverse resources, projects, and tasks characterized by unique attributes, interdependencies, and constraints. Moreover, the model can accommodate a variable system of various objective functions, whose elements can include customized optimization direction and priority value. To tackle these extended problems, we define an optimization model and apply a novel decision- making framework, which integrates metaheuristic search strategies, constructive algorithms, and multi-objective relative comparison models. Measurements were executed on created multi-objective resource-constrained multi-project scheduling to evaluate the performance of the proposed method. A new test problem containing 20 projects has been developed. The J30 series from the PSLIB benchmark set were utilized and we defined deadlines for selected projects. During the tests, 6 objective functions were investigated. The priorities of these objective functions were adjusted prior to each examination. In this paper the results of these performance tests are summarized. The obtained results demonstrate that the proposed method is effectively capable of solving multi-objective, multi-project scheduling problems.**

***Index Terms*—Resource Constrained Project Scheduling (RCPSP), Many Objective Search Method (MOSM)**

I. Introduction

PROJECT SCHEDULING is a crucial aspect of managing various systems and environments, as evidenced by numerous review papers available on the subject. Based on these reviews, we identified the resource-constrained project scheduling problem (RCPSP) as a suitable starting model for our investigation. The first optimization model for RCPSP was introduced by Pritsker in [1]. RCPSP problem proved as an NP-hard problem was done by Blazewicz at all. [2].

In the literature, several survey papers on RCPSP have been published, for example [3],[4], and [5]. Even though the original RCPSP model has been known for a relatively long time and it is sufficiently powerful for many cases, the practical applications require further extensions. An updated overview of various extensions of RCPSP model [6]categorized the model variants using the following main aspects: generalization of the activity, alternative precedence constraints and network characteristics, and consideration of multiple projects [6].

Our model simultaneously addresses two primary extension categories: multiple objectives and multiple projects. When optimization problems involve more than three objectives, they are often termed many-objective optimization problems by researchers [7]. The literature includes methods for addressing multiple objective functions concurrently [8].

To solve multi-objective or many-objective project scheduling problems three main patterns have been widely used by researchers: application of a weighted combination of objective functions, Pareto-efficiency and ordered objectives. Combination of minimization of makespan and costs is applied by Dai et al. [9] and by Schnabel et al. [10]. Considering optimization on individual project level and combined project portfolio level has been presented [11]. Pareto-efficiency considers a few objective functions as the main driver for the key performance indicators. Tabrizi defined two objectives; first is the combination of project completion time and due date, and the second one is called the ecological impact of orders [12]. Dridi et al. consider makespan and cost minimization related to renewable resources [13].

Many researchers propose new models and algorithms to solve different industrial optimization problems. This category includes, for example, vehicle routing [14], workload control [15], process control [16], worker assignment [17], project progress evaluation [18], layout optimization [19], supply chain optimization [20], indoor access point networks [21], encryption [22], process mining [23], production planning [24], production scheduling [25], and many other problems.

Several heuristics have been proposed to solve the RCPSP, classified into three main groups: single-pass heuristics, multi-pass heuristics, and metaheuristics. Single-pass and multi-pass heuristics quickly generate feasible schedules using serial and parallel schedule generation schemes (GS) with various priority rules. Metaheuristics, such as genetic algorithms, simulated annealing, tabu search, artificial immune algorithms, particle swarm optimization, bee colony optimization, and ant colony optimization, employ more complex algorithms to thoroughly explore the solution space. While metaheuristics can find high-quality solutions, they typically require more computing time.

Submitted on 01/06/2024
Krisztián Mihály, Gyula Kulcsár, and Mónika Kulcsárné-Forrai, Department of Information Engineering, University of Miskolc, Miskolc, Hungary (E-mail: krisztian.mihaly@uni-miskolc.hu – correspondence author, gyula.kulcsar@uni-miskolc.hu, monika.kulcsarne@uni-miskolc.hu)

DOI: 10.36244/ICJ.2025.5.7

The primary objective of our research was to develop an extended optimization model and a decision-making framework for effectively solving diverse multi-objective, multi-project scheduling problems. In this paper, we focus on examining and evaluating the performance of the proposed solution method. To demonstrate the flexibility and effectiveness of the method, we present numerical results.

The remaining part of the paper is structured as follows: Chapter II formulates the mathematical model, providing the theoretical framework for our study. Chapter III details the solution approach, outlining the methodologies and algorithms employed. Chapters IV and V present some numerical results, illustrating the application and validation of the models and methods. Finally, Chapter VI forms the conclusion, summarizing the key findings and addressing potential directions for further work.

II. Mathematical Model

The multi-project, multi-objective resource constrained scheduling problem is an extension of the classical Resource Constrained Project Scheduling problem and it is formulated as following: (A) Input data, (B) Indirect values calculated from input data, (C) Constraints for input data, (D) Primary decision variables, (E) Auxiliary values calculated from the primary decision variables, (F) Constraints for the solutions, (G) Objective functions.

A. Input data

The multi project scheduling problem is described with the input data presented in Table I.

TABLE I
Input data

Symbol	Description	Definition
$\boldsymbol{P}$	set of projects to be executed	$\boldsymbol{P} = \{p_1, p_2, \ldots, p_i, \ldots, p_{NP}\}$
$\boldsymbol{T}$	set of tasks to be executed	$\boldsymbol{T} = \{t_1, t_2, \ldots, t_j, \ldots, t_{NT}\}$
$\boldsymbol{R}$	set of available resource types	$\boldsymbol{R} = \{r_1, r_2, \ldots, r_k, \ldots, r_{NR}\}$
RC_k	the predefined maximal capacity of resource type r_k	$RC_k \in \mathbb{Z}_0^+$
$rr_{j,k}$	the amount of resource units required by task t_j from resource type r_k	$rr_{j,k} \in \mathbb{Z}_0^+$
$pt_{j,k}$	the processing time of task t_j on the resource type r_k	$pt_{j,k} \in \mathbb{Z}_0^+$
$\boldsymbol{TA}$	set of assigned pairs of projects and tasks	$\boldsymbol{TA} = \{(p_i, t_j) \mid \forall t_j \in \boldsymbol{T}\}$
$\boldsymbol{PRE}_j$	set of predecessor tasks of task t_j	$\boldsymbol{PRE}_j \subset \boldsymbol{T} \mid \forall t_j \in \boldsymbol{T}$
$\boldsymbol{TPROP}$	set of available task arguments	$\boldsymbol{TPROP} = \{\boldsymbol{T}_{rTime}, \boldsymbol{T}_{dTime}, \boldsymbol{T}_{prio}, \ldots, \boldsymbol{T}_{prop}\}$
$\boldsymbol{TP}_j$	set of maintained properties of task t_j	$\boldsymbol{TP}_j = \{T_{rTime,j}, T_{dTime,j}, T_{prio,j}, \ldots, T_{prop,j}\}$
$\boldsymbol{PPROP}$	set of available project arguments	$\boldsymbol{PPROP} = \{P_{rTime}, P_{dTime}, P_{tCost}, P_{prio}, \ldots, P_{prop}\}$
$\boldsymbol{PP}_i$	set of maintained properties of task p_i	$\boldsymbol{PP}_i = \{P_{rTime,i}, P_{dTime,i}, P_{tCost,i}, P_{prio,i}, \ldots, P_{prop,i}\}$

B. Indirect values calculated from input data

From input data additional data can be calculated, which helps to formulate constraints and objective functions in a condensed way. Symbols and definition of the indirect values are presented in Table II.

TABLE II
Input data

Symbol	Description	Definition
$\boldsymbol{PT}_i$	set of tasks assigned to project p_i	$\boldsymbol{PT}_i = \{t_j \mid (p_i, t_j) \in \boldsymbol{TA}\}$
$\boldsymbol{A}$	set of ordered pairs of tasks by prerequisite definition	$\boldsymbol{A} = \{(t_j, t_{pre}) \mid t_j \in \boldsymbol{T}, t_{pre} \in \boldsymbol{PRE}_j\}$
G	directed graph of tasks	$G = (\boldsymbol{T}, \boldsymbol{A})$
$TP(t_j, T_{prop})$	function to get T_{prop} property of task t_j	
$PP(p_i, P_{prop})$	function to get P_{prop} property of project p_i	

C. Constraints for input data

Table III lists the constraints to be fulfilled by the input data.

TABLE III
Constraints for input data

Description	Definition
Negative capacity of a resource type is not allowed.	$RC_k \geq 0 \mid \forall r_k \in \mathrm{R}$
Negative processing time is not allowed.	$pt_{j,k} \geq 0 \mid \forall t_j \in \boldsymbol{T}$
Resource requirement cannot exceed the available maximum resource type capacity.	$0 \leq rr_{j,k} \leq RC_k \mid \forall t_j \in \boldsymbol{T}, \forall r_k \in \boldsymbol{R}$
Task cannot exist without project assignment.	$(p_i, t_j) \in \boldsymbol{TA} \mid \forall t_j \in \boldsymbol{T}$
Task cannot be a predecessor for itself.	$\boldsymbol{PRE}_j \subseteq (\boldsymbol{T} \setminus \{t_j\})$
No circle is allowed in the task dependencies.	Graph G shall fulfil the requirement of directed acyclic graph.

D. Primary decision variables

The main decision variables are the start time of the tasks. Symbols of decision variables are defined in Table IV. These values are non-negative, integer values that describes a solution for the problem. The application area can define the mapping between the integer values to applicable time units such as seconds, days, weeks, and tasks.

TABLE IV
Primary decision variables

Symbol	Description	Definition
S_j	start time of task t_j	$S_j \in \mathbb{Z}_0^+$
SCH	start vector representing start time of each task	$SCH = (S_1, S_2, \ldots, S_j, \ldots, S_{NT})$

E. *Auxiliary values calculated from the primary decision variables*

Additional auciliary values are defined in Table V., which can be calculated from the primary decision variables and the input data, allowing the constraints and objective functions in the mathematical model of the RCMOMPSP problem to be more simply formulated.

TABLE V
AUXILIARY VALUES CALCULATED FROM THE PRIMARY DECISION VARIABLES

Symbol	Description	Definition
C_j	completion time of task t_j	$C_j \in \mathbb{Z}_0^+$
C_{max}	maximum completion time of the tasks	$C_{max} = max_{j\in\boldsymbol{T}}\,(C_j)$
$\boldsymbol{RT}(\tau)$	set of running tasks at time τ	$\boldsymbol{RT}(\tau) = \{t_j \in \boldsymbol{T} \mid S_j \le \tau \le C_j\}$
$\lambda_{j,k}(\tau)$	indicates that task t_j is processed on resource r_k at time τ	$\lambda_{j,k}(\tau) = \begin{cases} 1, if\ S_j \le \tau \le S_j + pt_{j,k} \\ 0, otherwise \end{cases}$
$RL_k(\tau)$	resource load at r_k at time τ	$RL_k(\tau) = \sum_{j\in\boldsymbol{RT}(\tau)} \lambda_{j,k}(\tau) * rr_{j,k}$
$\lambda_{tdelayed}(i)$	indicates that task t_j is delayed	$\lambda_{tdelayed}(j) = \begin{cases} 1,\ if\ TP(t_j, T_{dTime}) > 0\ \wedge \\ \quad C_j > TP(t_j, T_{dTime}) \\ 0,\ otherwise \end{cases}$
$TL(j)$	task latency	$TL(j) = \lambda_{tdelayed}(j) * (\,C_j - TP(t_j, T_{dTime})\,)$
$\lambda_{pdelayed}(i)$	indicates that project p_i is delayed	$\lambda_{pdelayed}(i) = \begin{cases} 1, if\ PP(p_i, P_{dTime}) > 0\ \wedge \\ \max_{j\in PT_i}(\{C_j\}) > PP(p_i, P_{dTime}) \\ 0, otherwise \end{cases}$

F. *Constraints for the solutions*

A schedule is a feasible solution if the constraints defined for the solution are not violated. The constraints are represented in Table VI.

TABLE VI
CONSTRAINTS FOR THE SOLUTION

Description	Definition
A task cannot be started until all predecessor tasks have been completed.	$S_j \ge \max_{pre\in\boldsymbol{PRE}_j}(\{C_{pre}\})$
If task has predefined release time it cannot be started earlier.	$S_j \ge TP(t_j, T_{rTime})$
If project p_i has predefined release time, then any of the assigned tasks cannot be started earlier.	$S_j \ge max_{(p_i,t_j)\in\boldsymbol{TA}}\,(PP(p_i, P_{rTime}))$
Load of any resource type cannot exceed the maximum capacity of the resource type at any time.	$RL_k(\tau) \le \mathrm{RC}_k \mid \forall r_k \in \boldsymbol{R}, 0 \le \tau \le C_{max}$

G. *Objective functions*

We have designed the optimization model to find the best solution by simultaneously considering multiple objective functions. The main goal of our approach is to ensure that the solving method does not rely on specific information about the objective functions; we aim to use a general model that is completely independent of the actual set of objective functions. However, in this paper, we present six objective functions in Table VII. that were used in the performance measurement.

TABLE VII
OBJECTIVE FUNCTIONS

Symbol	Description	Definition
$L_{max} = f_{PC_{max}}$	maximum lateness	$f_{PC_{max}} = max_{j\in\boldsymbol{T}}\,(TL(j))$
$T_{max} = f_{TT_{max}}$	maximum tardiness	$f_{TT_{max}} = \max\,(0, max_{j\in\boldsymbol{T}}(TL(j)))$
$T_{sum} = f_{TT_{sum}}$	sum of tardiness of all tasks	$f_{TT_{sum}} = \sum_{j\in\boldsymbol{T}}(TL(j))$
$U_{sum} = f_{TDC}$	sum of tardy tasks	$f_{TDC} = \sum_{j\in\boldsymbol{T}} \lambda_{tdelayed}(j)$
C_{max}	maximum completion time	$C_{max} = max_{j\in\boldsymbol{T}}\,(C_j)$
C_{sum}	sum of completion time of all tasks	$C_{sum} = \sum_{j\in\boldsymbol{T}}(C_j)$

III. SOLUTION APPROACH

In a previous work, we proposed a hybrid method to solve extended scheduling problems [23]. The approach to solve scheduling problems consists of two main components: a search module and a scheduling construction module. The process contains three primary decision-making phases: (1) Establishing a priority sequence for tasks to be scheduled in the search module, (2) Incorporating the highest priority task into the schedule while accounting for resource availability and task requirements in the schedule generation, (3) Selecting the optimal schedule from a set of feasible solutions by evaluating each objective function concurrently. The proposed method was named as multi objective search method (MOSM).

A. *Search algorithm to construct priority sequence of tasks*

Inspired by the Jaya algorithm [27] and considering local search principles, we developed a hybrid approach to use an evolutionary algorithm together with local search techniques. This algorithm features three novelties:

(1) A search strategy that facilitates exploration of the search space. (2) An enhanced procedure for generating new candidate members for the next generation. (3) Utilize multiple objective functions simultaneously with individual priorities and optimization directions. The pseudo-code of the developed MOSM algorithm is presented in Algorithm 1. Notations used in the pseudo-code is listed in the Table VIII.

TABLE VIII
MOSM ALGORITHM NOTATION

Notation	Description
$M_{i,g}$	Control vector of the i^{th} candidate member of the g^{th} generation containing the priority sequence of tasks.
$SC(M_{i,g})$	The feasible schedule generated by the scheduling construction algorithm (SC) considering $M_{i,g}$ control vector.
NM	Number of members in the population
NG	Number of generations

Algorithm 1: Many-Objective Scheduling Method (MOSM)

```
Input: RCMPMOSP Problem definition, search parameters
Output: Scheduling vector SCH
Begin
1.1   i = 1; g = 1;
1.2   while (i <= NM)
1.3      Create the Mi,g vector by the parallel schedule generation scheme
         with the earliest starting task selection rule
1.4      Construct the SC(Mi,g)
1.5      Evaluate the SC(Mi,g)
1.6      Add the Mi,g vector as the ith member to the 1th generation;
1.7      i = i + 1
1.8   end while
1.9   Select the best member of 1th generation
1.10  g = g + 1;
1.11  while (g <= NG)
1.12     i = 1
1.13     limit = 1 – g / NG;
1.14     while (i <= NM)
1.15        number = Generate a pseudo random number from the interval
            [0, 1] with uniform probability;
1.16        if (number < limit)
1.17           Create a new candidate Mi,g vector by mutating the ith
               member of the previous generation;
1.18        else
1.19           Create a new candidate Mi,g vector by mutating the best
               member of the previous generation;
1.20        Construct the SC(Mi,g) schedule based on the Mi,g vector;
1.21        Evaluate the SC(Mi,g) schedule;
1.22        Add the better version of Mi,g and Mi,g-1 as the ith new member
            to the gth generation;
1.23        i = i + 1
1.24     end while
1.25     Select the best member of the gth generation;
1.26     g = g + 1;
1.27  end while
1.28  Return best member of the latest generation;
end
```

B. *Construction algorithm to generate a feasible schedule considering task selection control parameters*

MOSM consists of a constructive algorithm to generate a feasible schedule. The schedule construction algorithm (SC) begins with an empty schedule. In each iteration, a new, feasible partial schedule is generated by adding one selected task to the existing partial schedule. This method continues until all tasks are scheduled. Algorithm 2 provides the pseudo code for SC.

Algorithm 2: Schedule Construction Algorithm (SC)

```
Input: RCMPMOSP Problem definition, task selection rules and priority
values
Output: Feasible schedule SC(Mi,g)
Begin
2.1   Create an empty schedule ;
2.2   i = 1;
2.3   while (i <= number of tasks)
2.4         Choose the t_c task from the ith position of the vector Mi,g;
2.5         Insert the chosen t_c task into the schedule with the earliest
            applicable start time;
2.6         i = i + 1;
2.7   end while
2.8   Return schedule SC(Mi,g)
End
```

The MOSM algorithm was compared to the published PSLIB benchmark results using the C_{max} objective function. The experimental result showed a good performance of the applied solution method. Based on this experience, we examined the efficiency of the procedure when applied to multiple projects and multiple objective functions simultaneously.

C. *Qualification Method for Comparing Schedules*

MOSM can incorporate numerous objective functions. We assume that the set of objective functions is unlimited and can include various elements with different priorities and optimization directions. One scheduling always have one predefined set of applied objective functions. To compare two feasible schedules, represented by two members of a generation the following approach is applied.

Let s_x and s_y be two candidate feasible schedules. The quality of a given schedule is represented by a given vector containing K real numbers, each vector coordinate represents the actual value of the corresponding objective function.

TABLE IX
RELATIVE QUALIFICATION NOTATION

Notation	Description
u	$u = (u_1, u_2, \ldots, u_k, \ldots, u_K), u_k \in \mathbb{R}$; u denotes the vector containing the values of the objective functions considering the given schedule to be compared.
z	$z = (z_1, z_2, \ldots, z_k, \ldots, z_K), z_k \in \{-1, 1\}$; z denotes the vector containing the optimization directions of the objective functions. The value of z_k is 1 if we want to minimize the k^{th} objective function. The z_k is -1 if the k^{th} objective function must be maximized.
w	$w = (w_1, w_2, \ldots, w_k, \ldots, w_K), w_k \in \mathbb{Z}_0^+$; w denotes the vector containing priorities for the objective functions. Each w_k is a non-negative integer value ($w_k \geq 0$) that expresses the importance of the u_k value of the k^{th} objective function.

A distance function D is defined as follows.

$$D : \mathbb{R}^2 \to \mathbb{R},\ D(a, b) := \begin{cases} 0, & if \max(|a|, |b|) = 0 \\ \frac{b-a}{\max(|a|,|b|)}, & otherwise \end{cases} \quad (1)$$

The relative qualification uses notations defined in Table IX. Let x and y be two vectors with type u. These vectors contain the values of objective functions, and they represent the absolute quality of candidate schedules s_x and s_y to be compared. We define the F function to express the relative quality of y compared to x as a real number.

$$F : u^2 \to \mathbb{R},\ F(x, y) := \sum_{k=1}^{K} (w_k \cdot z_k \cdot D(x_k, y_k)) \quad (2)$$

IV. MULTI-OBJECTIVE, MULTI-PROJECT BENCHMARK SET

A necessary element for conducting the tests of the solution approach is to have multi-project, multi-objective scheduling problems available. For this, the J30 benchmark scheduling problems documented in the PSLIB benchmark set have been used as a basis [26]. The instances j301_1, j301_2, j301_3, ..., j302_10 of the PSPLIB J30 series together was considered as a project portfolio. The project portfolio created with this procedure thus contains 20 projects, each with 30 tasks. The operation times, prerequisite constraints, and resource requirements given in the benchmark tasks were unchanged.

To ensure the comparability of each test, all tests use the same initial parameters and state space. During the search, the initial task priority vector is determined using the parallel generation scheme. The generation scheme used the earliest finish time priority rule.

V. Experimental result

The executed tests are presented in grouped, sequentially organized subsections. Besides detailing the main input parameters of the tests, the test results, and their evaluative descriptions, the step-by-step presentation aims to provide insight into the background of additional conclusions identified during the testing.

A. Test results on individual objective functions

In the first test group, we used the first 20 projects of the PSLIB RCPSP J30 benchmark set. An individual due date (P_{dTime}) has been defined for every fifth project (5th, 10th, 15th, and 20th). Due to the characteristics of the PSLIB benchmark tasks, specifying the completion deadline at the project level can also be modeled by the completion deadline given for the last, virtual closing task of the selected projects. For the selected projects the project completion due date was determined based on the best C_{max} values documented in the PSLIB benchmark results, such that the completion due date of the selected project equals the sum of the C_{max} values of the fewer selected projects, as follows:

$$P_{dTime,i} = \sum_{j \le i} C_{max,j} \mid i,j \in \{5,10,15,20\} \quad (3)$$

The primary aim of the executed tests is to verify the controllability of the search by assigning a non-negative priority to a single designated objective function. The secondary aim of the tests is to use the objective function values obtained from the individual objectives as reference values for evaluating the results of subsequent tests.

Test results are presented in the Table X.

TABLE X
Individual objective function priorities and calculated objective function values

	Objective function priority						Objective function value					
	L_{max}	T_{max}	T_{sum}	U_{sum}	C_{max}	C_{sum}	L_{max}	T_{max}	T_{sum}	U_{sum}	C_{max}	C_{sum}
1	1	0	0	0	0	0	-8	0	0	0	321	79707
2	0	1	0	0	0	0	0	0	0	0	302	79557
3	0	0	1	0	0	0	0	0	0	0	311	80470
4	0	0	0	1	0	0	256	256	436	2	323	84605
5	0	0	0	0	1	0	237	237	637	4	276	79837
6	0	0	0	0	0	1	227	227	548	4	289	72393

The search characteristics of the test #1 execution are depicted in Fig 1. The points on the diagram show the improvements in the values of the selected objective function. The horizontal axis represents the number of search steps. It can be observed that within the first 500 iterations, the search module frequently finds better schedules than the previously known solution. Subsequently, the number of iterations required to find better solutions increases. It is noticeable that within the range of 2000-2500 iterations, it can once again make several improvements. The presented figure confirms that the search module follows an expected characteristic.

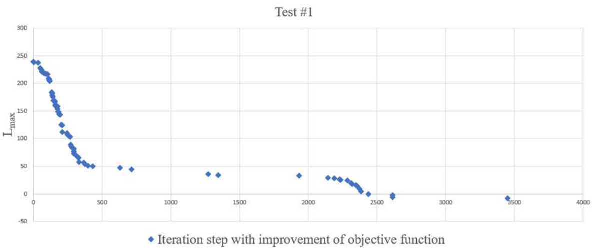


Fig. 1. Magnetization as a function of applied field. Note that “Fig.” is abbreviated. There is a period after the figure number, followed by two spaces. It is good practice to explain the significance of the figure in the caption.

B. Test results on multiple objective functions considered simultaneously

In the second test group, I examined the simultaneous use of multiple objective functions. The scheduling task remained unchanged compared to the first test group. I grouped the objective functions according to three criteria and tested the values of different priorities accordingly. The considered groups are as follows:

- Minimizing the delays (L_{max}, T_{max}, T_{sum}, U_{sum})
- Minimizing the completion time of the last task (C_{max})
- Minimizing the sum of task completion times (C_{sum})

Table XI. presents the results of the defined objective function priorities and the calculated objective function values.

TABLE XI
Simultaneous objective function priorities and calculated objective function values

	Objective function priority						Objective function value					
	L_{max}	T_{max}	T_{sum}	U_{sum}	C_{max}	C_{sum}	L_{max}	T_{max}	T_{sum}	U_{sum}	C_{max}	C_{sum}
7	1	1	1	1	1	1	-7	0	0	0	287	73837
8	0	10	20	8	1	1	0	0	0	0	284	74760
9	0	2	2	2	1	0	0	0	0	0	283	79365
10	5	0	0	0	1	0	-7	0	0	0	280	79193
11	0	0	2	0	0	1	-2	0	0	0	297	73819

During the 7th and 8th tests all three criteria have been considered simultaneously. The aim of these two tests was to determine how close we could approach the minimum of C_{sum} and C_{max} compared to 5th and 6th test. The parameters set for the 7th test favored C_{sum}, while those for the 8th test favored C_{max}. In the 9th and 10th tests the C_{sum} objective function has not been considered, that is why the priority has been set to zero. It can be observed that this favored the computed results of C_{max} (first 287 and 284, then 283 and 280). In the 10th test series, the best value for C_{max} was influenced by the L_{max} objective function. This generated earliness and resulted in an improvement in C_{max}. In the 11th test, only C_{sum} was considered while adhering to the deadlines. Here, T_{sum} alone managed to keep the deadline overrun at zero. The value of C_{sum} decreased quite well. A better value than this was only achieved in previous tests when the C_{sum} objective function was used independently (6th test).

C. *Alternative way to model due date of a project*

The due-date modelling of a project was modified in the next test group. In the alternative modelling the project completion due date was set as task due date for every task associated with the given project. The correctness of this modification is not affected because if task-level due dates were not otherwise specified, the project completion deadline can be considered as the maximum task deadline for all tasks within the given project.

Table XII. presents the considered objective function priorities and the calculated objective function values.

TABLE XII
SIMULTANEOUS OBJECTIVE FUNCTION PRIORITIES AND CALCULATED OBJECTIVE FUNCTION VALUES WITH ALTERNATIVE DUE-DATE MODELLING

	Objective function priority						Objective function value					
	L_{max}	T_{max}	T_{sum}	U_{sum}	C_{max}	C_{sum}	L_{max}	T_{max}	T_{sum}	U_{sum}	C_{max}	C_{sum}
12	0	0	0	1	0	0	-2	0	0	0	304	79814
13	1	0	0	0	0	0	-8	0	0	0	321	79707
14	0	1	0	0	0	0	0	0	0	0	302	79557
15	0	0	1	0	0	0	-1	0	0	0	331	80912
16	0	0	10	0	1	0	213	0	0	0	279	79377
17	0	0	10	0	0	1	0	0	0	0	292	73797

The 12th test confirmed the correctness of the assumption made in the second test group. A different formulation of the same problem with the same model and scheduling method leads to better solutions because the search module can make more effective improvements based on feedback on due date violations.

The result of the 13th test completely matches the result of the 1st test. This was expected since, in terms of L_{max} optimization, the individual deadlines of tasks are not significant.

The result of the 14th test completely matches the result of the 2nd test because, in terms of T_{max} optimization, the individual deadlines of tasks are relevant.

In the 16th test, it was possible to further reduce the value of C_{max} by one unit while adhering to the deadlines. In the first test series, the best value was 280, while here the method achieved 279.

In the 17th test, it was possible to decrease the value of C_{sum} while adhering to the due dates. In the first test, the best value was 73819, while in this test it decreased to 73797.

The achieved results confirm that the formulation of the task greatly influences the quality of the solution, thus justifying the use of sophisticated optimization models.

D. *Evaluation of tardy projects*

In this test group, the investigation focused on how the algorithm handles the scenario where delays are unavoidable. To evaluate this circumstance, in this test group the initial scheduling problem was enhanced with additional data. The due dates of tasks were modified compared to the task used in the second test group. A single common due date was set for all tasks of the four projects with due date. The new due date was 39 time units. This value corresponds to the earliest due date of projects used in the previous test group.

If not all tasks can be completed by due date, then the T_{max}, T_{sum}, and U_{sum} objective functions can become competitors of each other or even reinforce each other's effect. In such cases, the following questions can be formed: If due date overrun is unavoidable, would it be more acceptable to have several small delays or fewer large delays? Alternatively, would it be advisable to minimize the total number of tardy delays?

Table XIII. shows the tested objective function priority considerations and the measured objective function values.

TABLE XIII
MEASUREMENT RESULT OF CERTAINLY DELAYED PROJECTS UNDER DIFFERENT OBJECTIVE FUNCTION PRIORITIES

	Objective function priority						Objective function value					
	L_{max}	T_{max}	T_{sum}	U_{sum}	C_{max}	C_{sum}	L_{max}	T_{max}	T_{sum}	U_{sum}	C_{max}	C_{sum}
18	1	0	0	0	0	0	25	25	553	38	309	80690
19	0	1	0	0	0	0	25	25	553	38	309	80690
20	0	0	1	0	0	0	33	33	492	35	344	82231
21	0	0	0	1	0	0	258	258	2848	28	325	81729
22	0	1	1	1	0	0	30	30	555	37	313	78621
23	0	1	1	10	0	0	35	35	575	29	313	81613
24	0	10	1	1	0	0	26	26	594	41	317	81451
25	0	1	10	1	0	0	29	29	522	39	317	80272

The results of tests 18 and 19 are completely identical. This was expected because if there is a certain delay, then the value of L_{max} is greater than zero and equals the value of T_{max}.

In tests 20, 21, and 22, T_{max}, T_{sum}, and U_{sum} individually provided better results from their own perspectives than the solutions achieved with the other two. When used together in the system, these objective functions can produce compromise solutions. Test 22 provides an example of this situation.

In tests 23, 24, and 25, three selected objective functions were set in such a way that one function has a priority value of 10, while the priority of the other two was set to 1. The results show that the higher priority had an impact on the search, which is reflected in the results as well.

From tests 24 and 25, an important conclusion can be drawn that appropriately chosen priority values can achieve better results in the given scenario than the values achievable by individually applied objective functions.

VI. Conclusion and Future Work

This paper presented an extended model to solve complicated project scheduling problems. An optimization model is proposed to define the mathematical model of the investigated problem class, which involves diverse resources, projects, and tasks characterized by unique attributes, interdependencies, and constraints. The proposed model can also accommodate a flexible system of various objective functions, whose elements can include customized optimization direction and priority value. A novel decision-making framework is also presented. The solving approach integrates metaheuristic search strategies, constructive algorithms, and multi-objective relative comparison models.

In this paper, the focus was set to performance tests that were executed on a new multi- objective resource-constrained multi-project scheduling problem. To evaluate the performance of the proposed method, six objective functions were investigated. The priorities of these objective functions were adjusted, and the numerical result of the tests were evaluated. The summarized numerical results proved that the presented hybrid method performs very well. The calibrated priorities of the objective functions effectively control the searching of the best values for the active objective functions. The obtained results of the performance evaluation tests executed on the created RCMOMPSP problem express that MOSM method is efficient, robust, and flexible. Therefore, our work has achieved the research objectives. The obtained running results demonstrate that the proposed method is effectively capable of solving multi-objective, multi-project scheduling problems.

The proposed model flexibly adapts to changing optimization objectives. The applied many-objective optimization model based on relative qualification is able to solve effectively the many-aspect decision-making problems that focuses on selecting the next candidates in the search method based on previous evaluated results.

The concrete algorithms used in the decision-making phases can be easily exchanged. An excellently important feature of the investigated approach is that the searching process relies only on problem-independent information. A direct consequence of this fact is that the approach can be used in the case of any scheduling problem. In general, it can be stated that any effective permutation-oriented search metaheuristic algorithm is able to iteratively drives the schedule generation scheme by modifying the control sequence of tasks. The generation scheme has a very important role in this approach. Constructing the complete schedule means a reactive simulation that is encapsulates the constraint of the concrete problem type by making the search algorithm independent of the problem-specific constraints.

In practice, software developers can utilize the advantages of this separated and independent resolution of the necessary decision-making phases that support making up flexible software systems and cyber-physical systems. The wide applicability of the proposed optimization model for the extended many-project scheduling problem class is ensured by its formulation in such a way that the individual or system-oriented objective functions, constraints and attributes can also be taken into account. This paper offers a validated solving method for developers to realize practical scheduling applications for solving a wide variety of scheduling problems.

References

[1] A. A. B. Pritsker, W. J. Lawrence and P. M. Wolfe, "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach," *Management Science*, vol. 16, no. 1, pp. 93–108, Sept. 1969.

[2] J. Blazewicz, J. K. Lenstra and A. H. Kan, "Scheduling subject to resource constraints: classification and complexity," *Discrete Applied Mathematics*, vol. 5, no. 1, pp. 11–24, Jan. 1983, DOI: 10.1016/0166-218X(83)90012-4.

[3] Kolisch, R., Hartmann, S.: "Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis", Project scheduling, Project Scheduling: Recent Models, Algorithms and Applications, 147–178 (1999), DOI: 10.1007/978-1-4615-5533-9_7

[4] Hartmann, S., Kolisch, R.: Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research* 127(2), 394–407 (2000).

[5] Pellerin, R., Perrier N., Berthaut, F.: A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research* 280(2), 395–416, (2020).

[6] Hartmann, S., Briskorn, D.: A survey of variants and extensions of the resource-constrained project scheduling problem," *European Journal of Operational Research* 207(1), 1–14 (2010).

[7] Zhang, Y.-H., Gong, Y.-J., Zhang, J., Ling, Y.-b.: A hybrid evolutionary algorithm with dual populations for many-objective optimization. *IEEE Congress on Evolutionary Computation (CEC)*, (2016).

[8] Taha, K.: Methods That Optimize Multi-Objective Problems: A Survey and Experimental Evaluation. *IEEE Access* (8), 80 855–80 878 (2020).

[9] Dai, H., Cheng, W., Guo, P.: An Improved Tabu Search for Multi-skill Resource-Constrained Project Scheduling Problems Under Step-Deterioration. *Arabian Journal for Science and Engineering* 43(6), 3279–3290 (2018).

[10] Schnabel, A., Kellenbrink, C., Helber, S.: Profit-oriented scheduling of resource-constrained projects with flexible capacity constraints. *Business Research* 11(2), 329-356 (2018).

[11] Tirkolaee, E., Goli, A., Hematian, M., Sangaiah, A., Han, T.: Multi-objective multi-mode resource constrained project scheduling problem using Pareto-based algorithms. *Computing* 101(6), 547–570 (2019).

[12] Tabrizi, B.: Integrated planning of project scheduling and material procurement considering the environmental impacts. *Computers & Industrial Engineering* 120, 103–115 (2018).

[13] Dridi, O., Krichen, S., Guitouni, A.: A multiobjective hybrid ant colony optimization approach applied to the assignment and scheduling problem. *International Transactions in Operational Research* 21(6), 935–953 (2014).

[14] S. H. Huang, J. H. Huang, H. C. Lee and Y. Y. Tong, "A new hybrid algorithm for solving the vehicle routing problem with route balancing," *International Journal of Industrial Engineering and Management*, vol. 14, no. 1, pp. 51–62, March 2023.

[15] P. Renna, "Workload control order release with controllable processing time policies: an assessment by simulation," *International Journal of Industrial Engineering and Management*, vol. 13, no. 3, pp. 194–205, Sept. 2022.

[16] M. K. Adeyeri, S. P. Ayodeji, E. O. Olutomilola and O. J. Abayomi, "The Automated Process Control Model for Energy Consumption Optimization within Plantain Flour Processing Facility," *International Journal of Industrial Engineering and Management*, vol. 13, no. 3, pp. 206–214, Sept. 2022, DOI: 10.24867/IJIEM-2022-3-313.

[17] N. Fernandes, M. Thürer, F. Rodrigues, L. Pinto Ferreira, F. J. G. Silva and P. Avila, "Worker Assignment in Dual Resource Constrained Systems Subject to Machine Failures: A Simulation Study," *International Journal of Industrial Engineering and Management*, vol. 13, no. 2, pp. 110–118, 30 June 2022, DOI: 10.24867/IJIEM-2022-2-305.

[18] D. A. Kurniady, Nurochim, A. Komariah, Turwelis, H. T. Hoi and V. H. Ca, "Construction project progress evaluation using a quantitative approach by considering time, cost and quality," *International Journal of Industrial Engineering and Management*, vol. 13, no. 1, pp. 49–57, March 2022.

[19] S. Gao, J. Daaboul and J. Le Duigou, "Layout and scheduling optimization problem for a reconfigurable manufacturing system," *International Journal of Industrial Engineering and Management*, vol. 12, no. 3, pp. 174–186, Sept. 2021, DOI: 10.24867/IJIEM-2021-3-286.

[20] T. Trisna, M. Marimin, Y. Arkeman and T. C. Sunarti, "Fuzzy multi-objective optimization for wheat flour supply chain considering raw material substitution," *International Journal of Industrial Engineering and Management*, vol. 11, no. 3, pp. 182–191, Sept. 2020.

[21] L. Nagy, "Classical and quantum genetic optimization applied to coverage optimization for indoor access point networks", *Infocommunications Journal*, vol. 4, no. 4, 2012.

[22] K. Kubícek, J. Novotný, P. Švenda, and M. Ukrop, "New results on reduced-round tiny encryption algorithm using genetic programming," *Infocommunications Journal*, vol. 8, no. 1, pp. 2–9, 2016.

[23] L. Kovács, E. Varga and P. Mileff, "Application of Neural Network Tools in Process Mining". *Infocommunications Journal,* vol 15. pp. 13–19. ISSN 2061-2079, 2023, DOI: 10.36244/ICJ.2023.5.3

[24] S. K. Karimi, S. J. Sadjadi and S. G. J. Naini, "A bi-objective production planning for a flexible supply chain solved using NSGA-II and MOPSO," *International Journal of Industrial Engineering and Management*, vol. 13, no. 1, pp. 18–37, March 2022, DOI: 10.24867/IJIEM-2022-1-298.

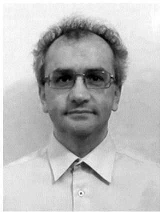
[25] P. Chetthamrongchai, O. Stepanenko, N. Saenko, S. Bakhvalov, G. Aglyamova and A. Iswanto, "A Developed Optimization Model for Mass Production Scheduling Considering the Role of Waste Materials," *International Journal of Industrial Engineering and Management*, vol. 13, no. 2, pp. 135–144, 30 June 2022, DOI: 10.24867/IJIEM-2022-2-307.

[26] K. Mihály and Gy. Kulcsár, "A New Many-Objective Hybrid Method to Solve Scheduling Problems", *International Journal of Industrial Engineering and Management*, vol. 14, no. 4, pp. 326–335, Dec. 2023.

[27] R. V. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 7, no. 1, pp. 19–34, 2016, DOI: 10.5267/j.ijiec.2015.8.004.

[28] PSLIB Single Mode Scheduling Benchmark Dataset, http://www.om-db.wi.tum.de/psplib/getdata_sm.html, last accessed 2024/11/10

Krisztián Mihály is an assistant lecturer at the Institute of Information Science, University of Miskolc (Hungary). He received his M.Sc. in Information Engineering from the Budapest University of Technology and Economics (Hungary) in 2008. He is currently working on his PhD thesis. He is working as a development architect at SAP Hungary Ltd., responsible for SAP PLM for Process Industry solutions. He is recipient of the "Award of the Best Trainer of the Year at SAP". His research interests include production planning and scheduling, project planning and scheduling, many-objective optimization, metaheuristics, and software architectures.



Gyula Kulcsár is an associate professor in the Institute of Information Science at the University of Miskolc (Hungary). He received an M.Sc. in Information Science from the University of Miskolc (Hungary) in 2001. and a Ph.D. in Information Science from the University of Miskolc (Hungary) in 2008. His research interests include production planning and scheduling, project planning and scheduling, many-objective optimization, metaheuristics, discrete event-driven simulation, and manufacturing control. He is a member of the public body of the Hungarian Academy of Sciences. He is the recipient of the Young Researcher Scientific Award from the Hungarian Academy of Sciences in 2008.

Mónika Kulcsárné Forrai an associate professor in the Institute of Information Science at the University of Miskolc (Hungary). She received an M.Sc. in Information Engineering from the University of Miskolc (Hungary) in 2001, and a Ph.D. in Information Science and Technology from the University of Miskolc (Hungary) in 2018. Her research interests include scheduling, search algorithms, optimization, production planning and control, enterprise resource planning, and project scheduling. She is a member of the public body of the Hungarian Academy of Sciences.