An Advanced Reactive Approach to Solve Extended Resource-Constrained Project Scheduling Problems

Krisztián Mihály, Gyula Kulcsár, and Mónika Kulcsárné-Forrai

Abstract—**This paper introduces a fast and configurable method for solving resource-constrained multi-project scheduling problems, using a multi-aspect decision-making procedure that combines a schedule generation scheme with various task-selection values and priorities. The goal of fast scheduling generation is to support reactive scheduling environments. During calculation each decision aspect is computed to produce numerical values, reflecting the importance of each aspect for candidate selection. These priorities can be tailored to specific optimization objectives. The priorities can be customized according to the objective of the optimization problem. The method was tested on the PSPLIB RCPSP J30 benchmark series to minimize project completion time using eight decision aspects. The average relative deviation from lower bounds was used to evaluate the impact of different decision aspect priorities. Although the focus was not on determining optimal priority values, the study explores the effectiveness of using multiple priority rules simultaneously in a configurable way in reactive scheduling environment. Performance tests confirm that the proposed method is flexible, robust, fast, and effective in solving the examined problem type.**

Index Terms—**Scheduling, Reactive Control, Priority Rule, Multi-Aspect Decision Making, Project Management**

I. Introduction

One of the core challenges in project management is creating an effective schedule that ensures the timely and efficient completion of projects. The Resource-Constrained Project Scheduling Problem (RCPSP) is a fundamental issue that impacts various industries, including manufacturing, software development, construction, logistics, and research and development. The primary goal of RCPSP is to allocate limited resources over time to a set of interdependent activities or tasks, optimizing performance indicators such as minimizing project duration, reducing lateness, minimizing costs, or maximizing resource utilization.

Our research focuses on the efficient solution of reactive scheduling and control problems in dynamically changing execution environments with numerous tasks. This paper presents a method suitable for reactive scheduling that simultaneously considers multiple decision aspects without iterative attempts.

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)

In this paper, we first provide a literature review in Chapter II of the Resource-Constrained Project Scheduling Problem (RCPSP). Chapter III presents the base problem extension to multi-project scheduling. Chapter IV discusses generation schemes and their applicability as simulation module for reactive scheduling. Chapter V introduces a multi-aspect qualification method as an enhancement option to the generation schemes. Chapter VI presents the experimental results by implementing the method on the PSPLIB benchmark set. Chapter VII concludes with a summary of the results.

II. Literature review

The definition of the RCPSP problem was introduced in 1969 [1] and was mathematically proven by Blazewicz et al. to be strongly NP-hard [2]. Several survey papers on RCPSP have been published [3]-[5]. Although the original RCPSP model is well-known and sufficient for many cases, practical applications require further extensions. An updated overview of these extensions is provided by Hartmann and Briskorn [6], who categorize model variants based on generalization of activities, alternative precedence constraints, network characteristics, and consideration of multiple projects.

One common approach to solving RCPSP involves using task selection priority rules with schedule generation schemes (SGS). These priority rules determine the task scheduling sequence, impacting the overall schedule efficiency. Priority rules are heuristic methods that order tasks based on specific criteria. Early examples include the earliest start time (EST), earliest finish time (EFT) [7], minimum slack time (MST) [8], and shortest processing time (SPT) [9]. Recent research explores the use of genetic-like evolution of task priority rules [10] and the automatic detection of the best applicable rules for RCPSP problems [11].

The literature presents approaches for handling multiple objective functions simultaneously [12]. Problems with more than three objectives are referred to as many-objective optimization problems [13], presenting new challenges, such as comparing candidate solutions using suitable performance metrics [14]. To address these challenges, researchers develop various methods based on existing approaches [15], advanced methods [16] and hybrid approaches [17].

DOI: 10.36244/ICJ.2025.5.1

III. Multi-Project Scheduling

The resource-constrained multi-project scheduling problem involves a set of activities to be executed on a set of resources, collectively forming projects. Each project is represented by an acyclic directed graph following an activity-on-node model. Nodes represent activities, which must be executed without interruption. Precedence relations, shown as arcs between nodes, indicate that a successor activity cannot start until all predecessors are completed.

The time horizon is divided into elementary time units (e.g., seconds, days, months), chosen based on the project-execution environment. Each activity's processing time is given as multiples of these units. Some activities may belong to multiple projects, creating interdependencies. Projects may have unique release times and due dates, and activities cannot start before their release times. Projects may differ in priority and have various scheduling goals modeled as objective functions, which project management uses for concurrent scheduling.

The execution system has a set of renewable resource types available for project activities. Each resource type has a time-dependent capacity constraint that specifies the available quantity in each time unit. These resources are not consumed but used by activities, then released upon completion, making them available again.

Each activity has specific resource requirements defining the type, quantity, and processing time needed. An activity can start only if the required resources are available for the necessary duration. Resource use begins simultaneously but may end at different times. Activities are non-interruptible, as pre-emption is not allowed. Each project includes virtual start and end activities, which require no resources and have zero processing time.

The investigated scheduling problems may involve multiple objectives with varying values, optimization directions, and importance levels. The objective functions can differ. To solve this extended problem type, a detailed schedule must be created specifying the exact start time for each activity. Our goal is to rapidly generate a feasible, near-optimal schedule that considers the objectives and meets all constraints.

This extended scheduling problem is referred to as ESP in this paper. In describing ESP, we draw on classical project scheduling concepts to establish its relationship to known models and highlight its unique features.

ESP includes RCPSP as a special case, making it also NP-hard. Additionally, ESP encompasses other classical scheduling problems like Single Machine Scheduling, Flow Shop Scheduling, and Job Shop Scheduling. While the literature often uses "operation" instead of "activity" in machine scheduling, and "job" to denote a set of operations, we use "task" to denote the elementary process and "project" to refer to a set of related tasks. This paper avoids the ambiguous term "job".

Our research objective was to develop a solution approach for making real-time decisions for ESP, especially in environments burdened with uncertainty and frequent unexpected events. Such environments require continuous adjustments and rapid scheduling decisions, such as in cyber-physical production systems or agent-based logistical systems. Considering these factors, we chose the reactive scheduling strategy as our fundamental approach.

IV. Generation scheme as a suitable base for reactive scheduling

The Schedule Generation Scheme (SGS) is a well-known type of predefined, rule-based, constructive methods. Starting with an empty schedule, SGS iteratively adds one unscheduled task to the partial schedule until all tasks are scheduled. Table I presents the applied notations of variables used in the algorithm.

TABLE I
NOTATION FOR GENERATION SCHEME

Notation	Description
$\boldsymbol{T}$	Set of tasks to be scheduled
m	Iteration of the generation scheme
$\boldsymbol{TGEN_m}$	$\boldsymbol{TGEN}_m \subseteq \boldsymbol{T}$; the already scheduled tasks in iteration m
$\boldsymbol{D_m}$	$\boldsymbol{D}_m \subseteq \boldsymbol{T}$; the decision set of tasks in iteration m
$SGEN_m$	$SGEN_m = (S_1, S_2, \dots, S_j, \dots, S_{NT})$, the starting time vector of already scheduled tasks in iteration m
$CGEN_m$	$CGEN_m = (C_1, C_2, \dots, C_j, \dots, C_{NT})$, the completion time vector of already scheduled tasks in iteration m

Algorithm 1. presents the pseudo code of generation schemes in general.

Algorithm 1: Generation Scheme Algorithm (SGS)

Input: ESP problem definition, task selection rules and priority values
Output: Feasible schedule
Begin
Create an empty schedule ;
m = 0; $\boldsymbol{TGEN}_0 = \emptyset, SGEN_0 = (0,0,\dots,0); CGEN_0 = (0,0,\dots,0)$;
while ($\boldsymbol{TGEN}_m <> \boldsymbol{T}$)
    m = m + 1;
    Calculate $\boldsymbol{D_m}$
    Select one task from $\boldsymbol{D_m}$
    $\boldsymbol{TGEN}_m = \boldsymbol{TGEN}_m \cap \{t_m\}$
    Calculate $SGEN_m$, $CGEN_m$
end while
Return schedule $SGEN_m$
End

The calculation of $\boldsymbol{D_m}$ depends on the generation scheme variant. This paper uses the serial generation scheme variant, where a task is included in the $\boldsymbol{D_m}$ decision set in the m.-th iteration if and only if the task has not been scheduled and all its predecessor tasks have been scheduled.

$$\boldsymbol{D_m} = (\{\, t_j \,\} | t_j \notin \mathbf{TGEN_m} \wedge \forall t_{pre} \in \boldsymbol{PRE_j} \wedge t_{pre} \in \mathbf{TGEN_m}\ ) \quad (1)$$

V. A multi-aspect qualification method to select the most appropriate task from the decision set

In our reactive scheduling model, multiple task-selection decision aspects (TSDA) can be used simultaneously. We assume the set of applicable TSDAs is not limited and can encompass various items with different priorities and optimization directions. Optimization direction indicates whether a larger or smaller numerical value is desirable for the candidate task. In each execution of the solver, an actual system of applied TSDAs is given, and the actual value of each TSDA can be calculated.

Let s_x and s_y be two candidate tasks to be selected for adding to the partial schedule at iteration m. The calculated values of TSDAs are represented by a given vector containing K real numbers. The notations are given in Table II.

TABLE II
NOTATION FOR RELATIVE QUALIFICATION

Notation	Description
u	$u = (u_1, u_2, \dots, u_k, \dots, u_K), u_k \in \mathbb{R}$; u denotes the vector containing the values of TSDAs considering the given task to be compared.
z	$z = (z_1, z_2, \dots, z_k, \dots, z_K), z_k \in \{-1, 1\}$; z denotes the vector containing the optimization directions of TSDAs. The value of z_k is 1 if the smaller value of the k^{th} TSDA indicates the more favorable task. The z_k is -1 if the larger value pf the k^{th} TSDA indicates the more favorable task.
w	$w = (w_1, w_2, \dots, w_k, \dots, w_K), w_k \in \mathbb{Z}_0^+$; w denotes the vector containing priorities for the TSDAs. Each w_k is a non-negative real value ($w_k \geq 0$) that expresses the importance of the u_k value of the k^{th} TSDA.

A distance function D is defined as follows.

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

The relative qualification uses notations defined in Table II. Let x and y be two vectors with type u. These vectors contain the values of TSDAs, and they represent the absolute qualities of candidate tasks t_x and t_y to be compared. We define the F function to express the relative quality of y compared to x as a real number.

$$F : u^2 \to \mathbb{R}, \ F(x, y) := \sum_{k=1}^{K} (w_k \cdot z_k \cdot D(x_k, y_k)) \tag{3}$$

Using the return value of F(x, y), we can expresses the relative quality of vector y compared to vector x as the following:

- *y* is better than *x* if *F(x, y)* is less than zero.
- *y* is worse than *x* if *F(x, y)* is larger than zero.
- *y* and *x* are equally good if *F(x, y)* is exactly zero.

The presented F-based relative qualification model effectively solves the comparison of the candidate tasks from decision set $\boldsymbol{D_m}$ in the proposed solving approach.

VI. EXPERIMENTAL RESULT

We tested the presented method on the PSLIB benchmark RCPSP J30 problem set [18], which consists of 480 problem instances. The considered objective function was the maximum completion time of all tasks (C_{max}).

A. The applied set of TSDAs

Table III. presents the applied task selection decision aspects (TSDAs). During testing, eight TSDAs were examined. The RC_DEST is a dynamic (time-dependent) TDSA, whose value is updated by the construction algorithm through recalculation in changing decision situations. The others are static (time-independent) TDSAs, which are calculated once at the start of the construction algorithm.

TABLE III
USED TASK-SELECTION ASPECTS

Notation	Description
NSucc	Number of successors
ProcT	Processing time of the task
CPM_EST	Earliest start time calculated by Critical Path Method (CPM)
CPM_EFT	Earliest finish time calculated by CPM
CPM_LST	Latest start time calculated by CPM
CPM_LFT	Latest finish time calculated by CPM
RC_DEST	Dynamic earliest start time considering dynamically the actual resource constraints at the given time of decision making
DD	Due date of the task

B. Numerical results

To evaluate the performance of the reactive solver, we used the lower bound LB_p as reference value for each benchmark instance p. LB_p is provided for all instances in the PSLIB dataset. The result of the reactive solver executed on problem instance p is denoted by $C_{max,p}$.We calculated the the average relative deviation (ARD) of the reactive solver execution for the complete J30 benchmark dataset by Equation (4), where smaller ARD value indicates the better result.

$$\mathrm{ARD} = \frac{\sum_{p=1}^{P} \frac{C_{max,p} - LB_p}{LB_p}}{P} 100 \ [\%] \tag{4}$$

Table IV presents the measurement results of tests considering only one individual TSDA. The priority values are set to 1 or -1 depending on the optimization direction. For simplicity, we presented the multiplication of z_k and w_k as the priority of the k^{th} TSDA.

The CPM_LST achieved the lowest ARD value, which was approximately 0.04923.

TABLE IV
ARD EVALUATION RESULTS FOR INDIVIDUAL TASK-SELECTION ASPECTS

Test	Task selection priority multiplied with optimization direction								ARD
#	*RC_DEST*	*NSucc*	*ProcT*	*CPM_EST*	*CPM_EFT*	*CPM_LST*	*CPM_LFT*	*DD*	
1	1	0	0	0	0	0	0	0	0.078260
2	0	1	0	0	0	0	0	0	0.140896
3	0	0	1	0	0	0	0	0	0.174179
4	0	0	0	1	0	0	0	0	0.091479
5	0	0	0	0	1	0	0	0	0.115464
6	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0.049226**
7	0	0	0	0	0	0	1	0	0.051196
8	0	0	0	0	0	0	0	1	0.095355
9	0	-1	0	0	0	0	0	0	0.100995
10	0	0	-1	0	0	0	0	0	0.124379

TABLE V
ARD EVALUATION RESULTS FOR COMBINED TASK SELECTION ASPECTS

Test	Task selection priority multiplied with optimization direction								ADR
#	*RC_ DEST*	*NSucc*	*ProcT*	*CPM_ EST*	*CPM_ EFT*	*CPM_ LST*	*CPM_ LFT*	*DD*	
11	1	0	0	0	0	2	0	0	0.045571
12	1	-1	-1	1	1	1	1	0	0.061593
13	1	1	1	1	1	1	1	1	0.092054
14	1	1	-1	1	1	1	1	1	0.072527
15	1	-1	1	1	1	1	1	1	0.077187
16	1	-1	-1	1	1	1	1	1	0.061593
17	1	-1	-1	1	1	10	1	1	0.051461
18	1	-1	-1	1	1	20	1	1	0.049511
19	1	-1	-1	1	1	5	1	1	0.052672
20	2	0	0	0	0	1	0	0	0.038408
21	5	0	0	0	0	4	0	0	0.040886
22	5	-1	-1	1	1	1	1	1	0.05537
23	6	0	0	0	0	3	0	0	0.038408
24	6	-0.1	0.1	0	0	3	0	0	0.03776
25	6	-0.5	0.5	0	0	3	0	0	0.037249
26	6	-0.5	0.5	0	0	3	0.2	0	0.038329
27	6	-0.5	0.5	0	0	3	-0.2	0	0.037479
28	6	-0.5	0.5	0	0	3	0.5	0	0.037442
29	6	-0.5	0.5	0	0	3	1	0	0.037848
30	6	-0.5	0.5	0	0	3	-1	0	0.038671
31	6	-0.5	0.5	0	0.1	3	0	0	0.037556
32	6	-0.5	0.5	0	-0.1	3	0	0	0.037324
33	6	-0.5	0.5	0	0.5	3	0	0	0.037772
34	6	-0.5	0.5	0.05	0.05	3	0.05	0.05	0.037635
35	6	-0.5	0.5	0.1	0	3	0	0	0.037843
36	6	-0.5	0.5	-0.1	0	3	0	0	0.037999
37	6	-0.5	0.5	0.1	0.1	3	0.1	0.1	0.038126
38	6	-0.5	0.5	0.5	0	3	0	0	0.039375
39	6	-0.5	0.5	-0.5	0	3	0	0	0.041056
40	6	-0.5	0.5	1	0	3	0	0	0.041141
41	6	1	1	0	0	3	0	0	0.048826
42	6	-1	1	0	0	3	0	0	0.043549
43	6	-1	-1	0	0	3	0	0	0.056383
44	6	-1	-1	1	1	3	1	1	0.050003

Table V presents the results of 34 different tests where multiple TSDAs were used simultaneously. The table includes the applied priority values for each TSDA and the calculated ARD values.

The third-best result was given by RC_DEST, with an ARD value of approximately 0.07826 (Table IV Test #1). When these two selection criteria were applied together, an even better result was obtained than when used separately. For example, with priority value of RC_DEST 1 and CPM_LST 2 (Test #11), the ARD value was approximately 0.04557. Swapping the priorities of the two selection criteria (Test #20) reduced the ARD to approximately 0.03841.

An interesting observation was that in the combined solution, it was advantageous to assign a higher priority to the criterion that performed worse individually. The specific values of the TSDAs were not as important as their relative ratio. For instance, increasing the priorities threefold (e.g., from 2:1 to 6:3) resulted in an unchanged ARD value.

This experimental finding can be proven mathematically. Considering the F function (3) used for relative comparison, multiplying the priority values by a constant is equivalent to multiplying the final result of the F function by the same constant. Since the result of F is compared to zero, multiplying by any non-negative real number does not change its relation to zero. If F was greater, less, or equal to zero, it remains so. This should be considered when fine-tuning the priority values of TSDAs.

Involving additional selection criteria further improved the results. Table V examples show that the ARD value could be significantly reduced with different priority values. RC_DEST and CPM_LST remained dominant, but other criteria, such as NSucc and ProcT, also proved useful with smaller priority values. These criteria influence the decision when the F function value is close to zero based on higher-priority criteria. The priority of NSucc is negative because the method favors candidate tasks with a higher number of successor tasks. For other criteria, smaller numbers are more favorable.

In this experiment, the best result was achieved with the following priorities: 6; -0.5; 0.5; 0; 0; 3; 0; 0, (Test #25). The ARD value achieved this way was approximately 0.037249. Table V also shows that an ARD value below 0.038 could be reached with many settings. This is favorable because, for a given specific system, the set priority value scheme can be used effectively, providing a sufficiently sharp solution while maintaining flexibility.

While lower ARD values can be achieved with search algorithms, they come with significantly longer computation times due to the need to generate a large number of solutions iteratively. In contrast, a reactive construction algorithm produces only one solution. Reactive scheduling quickly adapts to real-time changes, as it operates based on predefined priority rules and decision criteria, allowing for immediate decision-making. In the investigated situations, the reactive approach may be more advantageous, as it does not require waiting for the search algorithms to respond, enabling faster reactions to environmental changes.

The experiment demonstrated that combining multiple task-selection aspects yields better solutions than using a single aspect. The decision-making method based on relative qualification can handle any finite number of selection criteria together. Assigning priority values to TSDAs is straightforward and flexible.

VII. CONCLUSION

In this paper, we present a novel reactive scheduling approach for extended project scheduling problems, aiming to create feasible and fast schedules for multiple projects with detailed resource requirements. The model uses a serial generation scheme with a new multi-priority decision-making procedure that considers many different decision aspects for deterministically selecting tasks.

The proposed extension is adaptable to a wide range of scheduling problems due to its problem-independent nature and ability to incorporate diverse decision aspects. These aspects can be calibrated and incorporated similarly to classical priority rules.

Our performance tests, conducted on the PSLIB J30 benchmark series, demonstrated that the combination of multiple decision aspects outperforms single aspects in minimizing the latest completion time. This supports our hypothesis that combining decision aspects using the relative qualification model is advantageous.

Future research will focus on applying extensions in the model to address the reactive decision-making requirements related to scheduling in cyber-physical systems.The proposed approach can be flexibly adapted to various optimization objectives and effectively applied to other selection or optimization problems due to its problem-independent elements. The schedule generation scheme, directed by the next task selection method, handles problem-specific constraints and can work with any generation scheme.

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] R. Kolisch, S. Hartmann, "Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis", *Project scheduling*, pp. 147–178, 1999., DOI: 10.1007/978-1-4615-5533-9_7

[4] R. Kolisch, S. Hartmann, "Experimental investigation of heuristics for resource-constrained project scheduling: An update", *European journal of operational research*, vol. 174, no. 1, pp. 23–37, 2006., DOI: 10.1016/j.ejor.2005.01.065

[5] R. Pellerin, N. Perrier and F. Berthaut, "A survey of hybrid metaheuristics for the resource-constrained project scheduling problem," *European Journal of Operational Research*, vol. 280, no. 2, pp. 395–416, 16 Jan. 2020, DOI: 10.1016/j.ejor.2019.01.063.

[6] S. Hartmann and D. Briskorn, "An updated survey of variants and extensions of the resource-constrained project scheduling problem," *European Journal of Operational Research*, vol. 297, no. 1, pp. 1–14, 16 Feb. 2022, DOI: 10.1016/j.ejor.2021.05.004.

[7] J. E. Kelley and M. R. Walker, "Critical-path planning and scheduling," *Proceedings of the Eastern Joint Computer Conference*, 1959. DOI: 10.1145/1460299.1460318

[8] A. A. Pritsker and W. Happ, *GERT: Graphical Evaluation and Review Technique*, Wiley, 1966.

[9] R. W. Conway, W. L. Maxwell, and L. W. Miller, *Theory of Scheduling*, Addison-Wesley, 1967.

[10] J. Luo, M. Vanhoucke, J. Coelho, W. Guo, „An efficient genetic programming approach to design priority rules for resource-constrained project scheduling problem", *Expert Systems with Applications*, vol. 198, 2022, DOI: 10.1016/j.eswa.2022.116753.

[11] W. Guo, M. Vanhoucke, J. Coelho, J. Luo, "Automatic detection of the best performing priority rule for the resource-constrained project scheduling problem", *Expert Systems with Applications*, vol. 167, 2021, DOI: 10.1016/j.eswa.2020.114116.

[12] K. Taha, "Methods That Optimize Multi-Objective Problems: A Survey and Experimental Evaluation," *IEEE Access*, vol. 8, pp. 80 855–80 878, 2020, DOI: 10.1109/ACCESS.2020.2989219.

[13] Y.-H. Zhang, Y.-J. Gong, J. Zhang and Y.-b. Ling, "A hybrid evolutionary algorithm with dual populations for many-objective optimization," *2016 IEEE Congress on Evolutionary Computation (CEC)*", Vancouver, BC, Canada, 2016, pp. 1610–1617, DOI: 10.1109/CEC.2016.7743981

[14] I. R. Meneghini, M. A. Alves, A. Gaspar-Cunha and F. G. Guimarães, "Scalable and customizable benchmark problems for many-objective optimization," *Applied Soft Computing*, vol. 90, May 2020, DOI: 10.48550/arXiv.2001.11591.

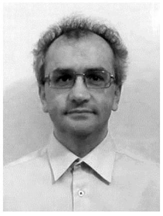
[15] H. Dai, W. Cheng and P. Guo, "An Improved Tabu Search for Multi-skill Resource-Constrained Project Scheduling Problems Under Step-Deterioration," *Arabian Journal for Science and Engineering*, vol. 43, no. 6, pp. 3279–3290, Jan. 2018, DOI: 10.1007/s13369-017-3047-4.

[16] S. Mane and M. R. Narasingarao, "A chaotic-based improved many-objective jaya algorithm for many-objective optimization problems," *International Journal of Industrial Engineering Computations*, vol. 12, no. 1, pp. 49–62, Oct. 2020, DOI:10.5267/j.ijiec.2020.10.001.

[17] 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.

[18] PSLIB Single Mode Scheduling Benchmark Dataset, last accessed 2024/04/03, https://www.om-db.wi.tum.de/psplib/getdata_sm.html.

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.