ISSN 2683-345X
journal homepage: http://ijiemjournal.uns.ac.rs/

International Journal of Industrial Engineering and Management

Volume 14 / No 4 / December 2023 / 326 - 335

Original research article

A New Many-Objective Hybrid Method to Solve Scheduling Problems

K. Mihály[a], G. Kulcsár[a,*]

[a] *University of Miskolc, Department of Information Engineering, 3515 Miskolc-Egyetemváros, Hungary*

ABSTRACT

This paper proposes a new hybrid method to solve extended scheduling problems that can include multiple resources, projects, and tasks with specific attributes, relationships, and constraints, as well as many objective functions with individual optimization directions and variable importance. A new decision-making procedure is proposed to combine metaheuristic search strategies, constructive algorithms, and many-objective comparison models. This paper also presents a concrete realization in which an advanced mixture search algorithm drives a flexible controllable constructive algorithm, and a relative qualification model ensures multi-objective optimization. Some numerical results illustrate the effectiveness of the proposed hybrid method.

ARTICLE INFO

Article history:

Received January 30, 2023
Revised November 3, 2023
Accepted November 14, 2023
Published online November 21, 2023

Keywords:
Multi-project;
Many-objective;
Resource-constrained;
Scheduling

*Corresponding author:
Gyula Kulcsár
gyula.kulcsar@uni-miskolc.hu

1. Introduction

Research and development of methods to solve practice-oriented multi-project scheduling problems is gaining an increasingly significant role. The motivation for our research originates from the management needs of discrete production systems. In manufacturing and services, the managers require efficient solutions to generate detailed schedules for the available resource components of the system to carry out activities, tasks, jobs, processes, and projects in flexible changing and resource-constrained execution environments.

In practice, many extended project scheduling problems occur when the companies follow the paradigms of intelligent manufacturing, cyber-physical production, and virtual enterprise. Many-project scheduling problems consists of multiple different decision-making subproblems that focus on resource allocating, sequencing, timing, costing and other optimization aspects. They can be defined by different optimization models. Diverse input data, parameters, decision variables, constraints must be considered, and many different key performance indicators must be optimized when creating detailed solutions for the managers.

One of the generalized features of the multi-project scheduling problems is that a given execution system performs many projects by carrying out a given set of tasks (activities). The elementary tasks can de-

Published by the University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia.

http://doi.org/10.24867/IJIEM-2023-4-342

pend on each other, they require many resources, and they can be connected to one or more projects at the same time. The resources can be classified into groups or types that are available with time-dependent strict capacity constraints. Enterprise application systems (MIS, EDB, ERP, MES, SCADA etc.) can provide detailed input data for the given scheduling problem. It is assumed that the necessary information is given. Managers defines the actual set of projects with their attributes, and they provide the current optimization objectives by specifying the set of objective functions and priorities. The scope of scheduling does not extend to the selection of projects, because all the given projects must be scheduled in the given time horizon.

This paper presents the results of our research focused on elaborating a general and flexible approach to solve many-objective resource-constrained multi-project scheduling problems.

2. Literature Review

The literature on optimization models and methods is very extensive. Many researchers propose new models and algorithms to solve different industrial optimization problems. This category includes, for example, vehicle routing [1], workload control [2], process control [3], worker assignment [4], project progress evaluation [5], layout optimization [6], supply chain optimization [7], production planning [8], production scheduling [9], and many other problems.

Scheduling plays an important role in managing of various systems. Many review papers can be found on the topic of scheduling. The main variants of the project scheduling problems are summarized for example in [10], [11], and [12]. Several important models and methods for project scheduling can be found in [13], and [14].

After reviewing papers focused on project scheduling, the resource-constrained project scheduling problem (RCPSP) was selected as an initial model to be extended and generalized. The first optimization model of RCPSP was published in [15]. RCPSP is an NP-hard problem [16].

Many further papers on RCPSP have been published. A survey of variants and extensions of RCPSP can be found in [17]. Even though RCPSP is sufficiently effective for many cases, further extensions are required by complicated situations that occur in practice. Hartmann & Briskorn give an updated review on the fundamental types of extensions related to RCPSP [18]. The main extension categories are the following: (1) generalization of the activity, (2) alternative precedence constraints and network characteristics, and (3) consideration of multiple projects. Our research is related to all three categories because the investigated problems can include elementary tasks, generalized activities, tasks related to several projects at the same time, many projects, and many objective functions.

Approaches to support optimization of multiple objective functions can be found in the literature [19]. The optimization problems with more than three objective functions are called as many-objective optimization problems. Such problems generate new challenges when we want to elaborate generalized, flexible, and efficient methods. New challenges come up when the methods create candidate solutions, and they must be compared by considering many objective functions in suitable performance metrics [20]. The many-objective optimization methods can be classified into the following main categories: (1) diversity-based, (2) indicator-based, (3) relaxed-dominance-based, (4) preference-based, (5) aggregation-based, (6) reference-set-based, and (7) dimensionality-reduction-based methods.

A combination of minimization of makespan and costs is applied in [21]. Considering optimization on individual project level and combined project portfolio level is presented in [22]. Tabrizi used two optimization objectives [23]. The first objective combines the project completion time and due date, while the second one focuses on the ecological impact of orders applied in the material procurement.

The multi-objective and many-objective optimization problems and the proposed algorithms are surveyed for example in [19] and [24]. Researchers proposed different methods to solve such many-objective optimization problems by using existing approaches, advanced approaches, hybrid approaches and new approaches. For example, Mane at al. presented a new approach that relies on a chaotic-based improved many-objective Jaya algorithm [24]. In this paper, we also propose a new generalized many-objective approach for comparing candidate solutions. Our comparison model can be used in any search algorithm.

Many metaheuristic search algorithms can be used to solve scheduling problems. For example, the set of commonly applied metaheuristics includes particle swarm optimization [25], tabu search [26], genetic algorithm [27], memetic algorithm [28], differential evolution algorithm [29], Jaya algorithm [30], and many other algorithms.

In recent years researchers proposed many hybrid methods for solving the RCPSP. Hybrid methods rely on different metaheuristic strategies. A survey and a comparative analysis of different hybrids are presented in [31]. The investigated hybrid methods were executed on the well-known PSPLIB benchmark data instances.

The primary purpose of our research was to develop a generalized many-objective multi-project scheduling method to serve flexibly the scheduling needs of production management and project management in practice.

The remaining part of this paper is organized as follows: Section 3 shows the problem description. Our approach and the proposed method are presented in Section 4. Some numerical results of performance tests are given in Section 5. Finally, the conclusions are included in Section 6.

3. Problem Description

The investigated scheduling problem is introduced as follows. We have a given set of tasks to be executed on a given set of resources. The related tasks together form a job that can also be considered as a specific project. A specific set of projects is given in the scheduling problem. Each project is represented by an acyclic directed graph, which can be considered as a task-on-node model. Each node (vertex) represents a task that means an operation or a process to be executed without preemption. Precedence relations may occur as conjunction arcs among the nodes in the graph. Each arc indicates that the successor task cannot be started until the predecessor task is completed.

The time horizon is given as a series of elementary time units. The time intervals can be regarded as a chosen elementary time units or period (e.g., second, minute, day, week, month, year etc.). This parameter can be specified by the real project-execution (business, manufacturing, and service) environment. The processing time of the task or the duration of the process are given as discrete multiples of one elementary time unit.

In the scheduling problem, there may be tasks that are connected to many projects at the same time, so this means that the projects may be interdependent. Projects may have unique release (starting) time constraints and specific due dates (deadlines). The tasks of the project cannot start before the constrained release time, and it would be advisable to avoid exceeding the project deadline. Projects may also have different priorities because the importance of the projects is not necessarily at the same level. In addition, project management can use a variety of different performance indicators to schedule projects simultaneously, because projects are also allowed to have a particular set of different scheduling goals (objective functions).

In the execution system of the projects, a given set of renewable resource types is available to perform the tasks of projects. Each resource type has a time-dependent non-negative capacity constraint that specifies the available quantity of resources in each elementary time unit (period) of the time horizon.

The resources are renewable. This means that the tasks do not consume the resources, just use them. The task allocates the needed quantities of the required resource types at the start and releases the allocated concrete resources at the end, so the released resources are available again.

Each task has a unique set of resource requirements. This set consists of elements that refer to different resource types. Such an element specifies the type of resource, the quantity of the concrete resources, and the processing time. The task can start if the required quantity of each required resource type is available for the relevant processing time. The use of different resource types starts at the same time, but it may not end at the same time. The execution of each task cannot be interrupted because pre-emption is not allowed.

There are two virtual tasks corresponding to the start and the end of each project. These virtual starting and virtual finishing tasks have no resource requirement. These can be represented by zero length processing time.

The investigated scheduling problems can contain many different scheduling goals at the same time. The value set, the optimization direction and the importance of the objective functions can be different. The predefined actual set of such objective functions may vary.

The detailed schedule of the tasks must be created by specifying the concrete starting time for each task to solve this extended scheduling problem. The purpose is to find the best schedule considering many objective functions simultaneously and subject to all constraints. This introduced extended scheduling problem will be denoted by the ESP abbreviation in this paper. ESP includes the RCPSP as a special subproblem, and the RCPSP is NP-hard, therefore, ESP also is an NP-hard problem.

4. A New Hybrid Method to Solve the ESP Problem

4.1. The Proposed Scheduling Approach

We present a hybrid method for searching the best schedule by considering many objective functions at the same time. Our solving approach has three main decision-making phases. These are the following:

(1) create a priority sequence of tasks to be scheduled,

(2) insert the highest priority task into the schedule by considering the availabilities of resources and the requirements of the chosen task,

(3) choose the best candidate schedule from a given set of feasible solutions by considering the value of each objective function simultaneously.

We developed the MOSM hybrid method to solve the ESP. The novelty of this method is that an advanced many-objective searching algorithm iteratively provides a candidate priority sequence of tasks (PST), and an advanced constructive algorithm generates a feasible schedule based on the PST in each iteration. Using this approach, we transform the ESP to a reduced searching problem, in which the primary decision variable is a vector that contains the priority sequence of tasks.

4.2. An Advanced Search Algorithm to Create the Priority Sequence of Tasks

In this subsection, we propose an advanced searching algorithm for driving the constructive algorithm. Our search strategy was inspired by the Jaya algorithm [30] and the local search principle. We developed a combined mixture of evolutionary and local search techniques. This algorithm has three novelties:

(1) A new searching strategy supports to explore the search space.

(2) An advance procedure creates new candidate members for the next generation.

(3) Many objective functions with individual priorities and optimization directions can be used at the same time.

The operation of MOSM is presented in Algorithm 1. The $M_{i,g}$ symbol represents the control vector as the i^{th} candidate member of the g^{th} generation. $M_{i,g}$ contains the priority sequence of tasks. The search algorithm iteratively specifies the new $M_{i,g}$ and calls the schedule construction algorithm (SC) with the actual $M_{i,g}$ to create a feasible schedule: SC($M_{i,g}$). NM means the number of members in the population, and NG denotes the number of generations. NM and NG are the input parameters of the algorithm.

The search algorithm follows a generation-based evolutionary strategy, but it does not use any crossover operator, only one mutation operator is applied. In any g^{th} generation, *NM* number of new candidate solutions are created ($i=1,2,\ldots,NM$). The mutation operator swaps two randomly selected elements in the priority sequence of tasks. This random exchange can also give a sequence that does not meet the precedence requirements. We have developed a normalization algorithm that corrects the sequence if necessary. The normalization algorithm (NA) runs in the given task sequence. If NA achieves a task whose predecessor tasks have not yet been affected, then NA skips it. When NA has affected a task, NA will look for the successor task with the lowest position number. If this position is smaller than the current position, NA jumps back there and continues running, otherwise NA step over the next position of the actual position. The corrected task sequence is given by the traversal order of tasks.

When producing the i^{th} new element, the starting base of the mutation can be the member with the same *i* serial number or the best member of the previous generation. The probability of mutating the best member increases linearly during the running of the search algorithm.

The new candidate $M_{i,g}$ results in a new feasible solution. If this is better than the previous $M_{i,g-1}$ based solution without mutation, then the new $M_{i,g}$ is accepted, otherwise the previous $M_{i,g-1}$ replaces the new $M_{i,g}$.

4.3. An Advanced Constructive Method for Transforming the Priority Sequence of Tasks into the Feasible Schedule

MOSM uses a constructive algorithm to create a feasible schedule by applying predefined building algorithms iteratively. The schedule construction algorithm (SC) starts from an empty schedule. In each iteration, a newer feasible partial-schedule is constructed by inserting the chosen task into the previous partial-schedule. This procedure is repeated until all tasks are scheduled. Algorithm 2 presents the pseudo code for SC.

Algorithm 1: Many-Objective Scheduling Method (MOSM)

```
{
Load the input data and initialize the variables;
i = 1; g = 1;
while (i <= NM)
{
    Create the M_{i,g} vector by the parallel schedule generation scheme with the earliest starting task rule;
    Construct the SC(M_{i,g}) schedule based on the M_{i,g} vector;
    Evaluate the SC(M_{i,g}) schedule;
    Add the M_{i,g} vector as the i^th new  member to the g^th generation;
    i = i + 1;
}
Select the best member of the g^th generation;
g = g + 1;
while (g <= NG)
{
    i = 1;
    limit = 1 – g / NG;
    while (i < NM)
    {
        number = Generate a pseudo random number from the interval [0, 1] with uniform probability;
        if (number < limit)
            Create a new candidate M_{i,g} vector by mutating the ith member of the previous generation;
        else
            Create a new candidate M_{i,g} vector by mutating the best member of the previous generation;
        Construct the SC(M_{i,g}) schedule based on the M_{i,g} vector;
        Evaluate the SC(M_{i,g}) schedule;
        Add the better version of M_{i,g} and M_{i,g-1} as the i^th new member to the g^th generation;
        i = i + 1;
    }
    Select the best member of the g^th generation;
    g = g + 1;
}
Return best member of the latest generation;
}
```

Algorithm 2: Schedule Construction Algorithm (SC)

```
{
Load the input data;
Create an empty schedule;
j=1;
while (j <= number of tasks to be scheduled)
{
    Choose the t_c task from the j^th position of the vector M_{i,g};
    Insert the chosen t_c task into the schedule with the earliest allowable start time;
    j = j + 1;
}
Return the schedule;
}
```

The schedule generation process covers the following three main decisions:

(1) Select schedulable tasks.

(2) Choose the best candidate task from the decision set.

(3) Insert the chosen task into the schedule.

Our SC solves these subproblems by deterministic rule-based procedures. We use a special generation scheme that integrates the first and second decision-making issues. The $M_{i,g}$ priority sequence of tasks is given by the searching algorithm, and the tasks are scheduled in the given order. The normalization algorithm ensures that the precedence constraints are fully respected.

SC method calculates the earliest time when the chosen task can be started. This calculation considers all the constraints: the actual availabilities of the needed resources, the latest finishing time of the related predecessor tasks, the releasing time constraints, and the resource requirements of the chosen task.

These presented decision-making procedures ensure great flexibility and high speed for the construction of the feasible schedule, while the input priority sequence of tasks contribute to achieve a high level of efficiency, robustness, and performance.

4.4. A New Qualification Model for Comparing Schedules

Many objective functions can be used in the MOSM. We suppose that the set of objective functions are not limited, and it can consist of various items with different priorities and optimization directions.

We assume that the actual system of objective functions (SOF) is given, and the actual value of each objective function can be calculated by an *Evaluate* algorithm that can be called by MOSM with a given candidate schedule as an input data structure.

Let s_x and s_y be two candidate schedules selected from the set of feasible schedules. The quality of a given schedule is represented by a given vector containing K real numbers. The k^{th} item of the vector means the actual value of the k^{th} objective function. We introduce the following notations:

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.

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.

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.

We define the following relative distance function:

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

The number a and b can be replaced by two values of a given objective function. In this case, the return value of function D expresses how much the second number has changed compared to the first number from the point of view of the given objective function.

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.

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

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 schedules in the proposed solving approach, so the proposed MOSM can carry out many-objective optimization by considering the actual system of objective functions.

5. Performance Tests and Some Numerical Results

The proposed MOSM hybrid method has been implemented in C++ programming language. Various performance tests were performed and evaluated.

First, we solved some problem types that have known optimal solutions. The purpose of these tests was to evaluate how well MOSM can obtain the optimal solutions. MOSM achieved excellent results in these tests.

In the second test phase, we solved benchmark instances of NP-hard scheduling problems. In this paper, we present only one example of them.

The chosen case study is focused on single-mode RCPSP problem instances originated from the well-known PSLIB benchmark datasets [32]. The RCPSP problem is one of the special cases of ESP. The RCPSP problem type has only one project consisting of many tasks to be scheduled, and different resource types are available in the system with individual constant capacity-constraints. In this problem type, the set of objective functions consists of only one item. The optimization objective is to minimize the maximum completion time of tasks.

We present the result of tests executed on the J30, J60 and J120 datasets from the PSLIB benchmarks. Each problem instance of the J30, J60, and J120 datasets defines a single project that consists of 30, 60, and 120 real and 2 virtual tasks. In addition, four resource types are defined by each problem instance. The proposed MOSM hybrid method is used to solve these problem instances. The searching parameters are listed in Table 1.

Datasets J30, J60, and J120 contain 480, 480, and 600 problem instances respectively. To compare the proposed MOSM method with other methods, we used the lower bound LB_p as reference value for each instance p. LB_p is provided by the critical path method (CPM) for instances of J60 and J120, while LB_p is given as optimal value of the objective function for instances of J30 in the PSPLIB dataset.

We considered the known LB_p values as reference values, and we solved each problem instance (p=1, 2, ..., P) exactly 15 times by using the MOSM. The best objective value $C_{best,p}$ of our 15 attempts on the instance p and the given LB_p reference value were compared. We calculated the value of the average relative deviation (ARD) by the following formula (3).

$$ARD = \frac{\sum_{p=1}^{P} \frac{C_{best,p} - LB_p}{LB_p}}{P} 100\ [\%] \quad (3)$$

The *ARD* values were calculated for all three J30, J60, and J120 datasets. Based on the limits of 1000, 5000 and 50000 commonly used in the literature, we also limited the allowed number of evaluations of the objective function when the algorithm ran. The *ARD* results are shown in Table 2-4, and some earlier published results of other methods are also listed for comparison.

In the case of J30, the ARD value is approximately 0,003% at the schedule limit of 50000. MOSM achieved the reference value of the objective function for 479 out of 480 problem-instances. In the case of J60, the ARD value of MOSM is higher with only 0.42 than the best value at limit of 50000. The obtained value for the J120 exceeds the best value by only 3.35.

The results obtained on J30, J60, and J120 benchmarks of PSPLIB are very encouraging because they are remarkably close to the best results, and they are better than many other results. The methods giving better results were developed specifically for the RCPSP problem type and the researchers fine-tuned the parameters of the methods focusing on the concrete problem instances.

In contrast, we solved the test problems as one of the special cases of a wider and more complicated problem area. Our method can be applied not only in the case of minimizing the completion time, but also works effectively in cases of arbitrarily chosen sets of objective functions. We did not use any problem-specific feature when creating the solution, so the proposed method is a generalized scheduling

Table 1. Search parameters of MOSM for the J30, J60, and J120 datasets of PSLIB benchmark

Parameter	Value
The number of generations	500, 2500, and 25000
The number of member solutions in the population	2
The number of attempts for solving a given problem instance	15

Table 2. Comparison of the average relative deviation values for J30

Algorithm	Schedule		
	1000	5000	50000
MOSM (this study)	0.12	0.03	0.003
QIGA (2021) [29]	0.20	0.12	0.06
TS-MODE (2020) [29]	0.06	0.01	0.00
HGA (2008) [33]	0.27	0.06	0.02
GRASP-FBI-SS (2013) [34]	0.57	0.39	0.23
Sequential(SS(FBI)) (2018) [35]	0.10	0.02	0.00
EQIGA (2022) [27]	0.06	0.02	0.00
Memetic algorithm (2020) [28]	-	0.00	0.00
JPSO (2011) [25]	0.29	0.14	0.04

Table 3. Comparison of the average relative deviation values for J60

Algorithm	Schedule		
	1000	5000	50000
MOSM (this study)	12.046	11.42	10.97
QIGA (2021) [29]	12.36	12.10	11.77
TS-MODE (2020) [29]	11.90	11.21	10.629
HGA (2008) [33]	11.56	11.10	10.7
GRASP-FBI-SS (2013) [34]	12.88	12.42	11.96
Sequential(SS(FBI)) (2018) [35]	11.38	10.93	10.58
EQIGA (2022) [27]	11.70	11.22	10.74
Memetic algorithm (2020) [28]	-	10.72	10.55
JPSO (2011) [25]	12.03	11.43	11.00

Table 4. Comparison of the average relative deviation values for J120

Algorithm	Schedule		
	1000	5000	50000
MOSM (this study)	38.03	34.16	33.94
QIGA (2021) [29]	36.30	36.02	35.08
TS-MODE (2020) [29]	34.40	32.86	30.59
HGA (2008) [33]	34.07	32.54	31.24
GRASP-FBI-SS (2013) [34]	38.16	37.30	36.32
Sequential(SS(FBI)) (2018) [35]	34.01	32.52	31.16
EQIGA (2022) [27]	36.22	35.12	33.55
Memetic algorithm (2020) [28]	-	32.76	31.12
JPSO (2011) [25]	35.71	33.88	32.89

procedure that is able to solve different scheduling problems. We did not generate trial solutions for calibrating parameters in advance, and we did not use any empirical constant to improve the performance. Even without them, our method is very flexible, robust, simple, fast, efficient and it can be easily applied in practice.

6. Conclusions

The paper focused on solving an extended class of scheduling problems. The presented solutions are suitable for predictively, reactively, and proactively scheduling many projects in different execution environments.

The proposed approach is based on different decision-making phases, so the solution generation is very flexible because each decision-making phase relies only on problem-independent information and each concrete algorithm can be easily changed and replaced independently. This approach utilizes the advantages of the separated and independent resolution of the necessary decision-making phases that make up the hybrid solving method. The generalized essence of the proposed hybridization is that a suitable many-objective searching metaheuristic algorithm iteratively drives a constructive algorithm by modifying the priority sequence of tasks. The constructive process plays the role of a reactive simulation that is embedded in the searching process to make the detailed solution based on the control priority sequence of tasks and all the given constraints.

Consequently, the proposed approach and the hybrid method can be used to solve any scheduling problem. In addition, the presented advanced many-objective relative qualification can be applied in solving any optimization problem.

The proposed approach can also flexibly adapt to changing requirements, constraints, and many optimization objectives, so it can be used in advanced planning and scheduling applications, enterprise management systems, on-demand flexible manufacturing and service systems.

The proposed hybrid method is well-suited to prepare project execution schedules for short, medium, and long-time intervals, while it can also provide updated schedules from time to time to meet the needs of the rolling planning pattern. In addition, the proposed flexible controllable constructive algorithm can also be realized in cyber-physical systems, because the real project execution system and the simulation-oriented cyber system can work together in real time, and the constructive decision-making phases are able to directly control the real events and activities.

The proposed method can also be used to control automated manufacturing systems, because the actual priority values that express the importance to perform operations earlier can be continuously refined by using artificial intelligence methods. Similarly, to manufacturing processes, the presented approach can also be adapted to any service process using appropriately designed abstract objects, scheduling levels, periods, constraints, and optimization objectives.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] 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," Int. J. Ind. Eng. Manag., vol. 14, no. 1, pp. 51-62, March 2023, doi: 10.24867/IJIEM-2023-1-324.

[2] P. Renna, "Workload control order release with controllable processing time policies: an assessment by simulation," Int. J. Ind. Eng. Manag., vol. 13, no. 3, pp. 194-205, Sept. 2022, doi: 10.24867/IJIEM-2022-3-312.

[3] 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," Int. J. Ind. Eng. Manag., vol. 13, no. 3, pp. 206-214, Sept. 2022, doi: 10.24867/IJIEM-2022-3-313.

[4] 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," Int. J. Ind. Eng. Manag., vol. 13, no. 2, pp. 110-118, 30 June 2022, doi: 10.24867/IJIEM-2022-2-305.

[5] 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," Int. J. Ind. Eng. Manag., vol. 13, no. 1, pp. 49-57, March 2022, doi: 10.24867/IJIEM-2022-1-300.

[6] S. Gao, J. Daaboul, and J. Le Duigou, "Layout and scheduling optimization problem for a reconfigurable manufacturing system," Int. J. Ind. Eng. Manag., vol. 12, no. 3, pp. 174-186, Sept. 2021, doi: 10.24867/IJIEM-2021-3-286.

[7] T. Trisna, M. Marimin, Y. Arkeman, and T. C. Sunarti, "Fuzzy multi-objective optimization for wheat flour supply chain considering raw material substitution," Int. J. Ind. Eng. Manag., vol. 11, no. 3, pp. 182-191, Sept. 2020, doi: 10.24867/IJIEM-2020-3-263.

[8] 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," Int. J. Ind. Eng. Manag., vol. 13, no. 1, pp. 18-37, March 2022, doi: 10.24867/IJIEM-2022-1-298.

[9] 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," Int. J. Ind. Eng. Manag., vol. 13, no. 2, pp. 135-144, 30 June 2022, doi: 10.24867/IJIEM-2022-2-307.

[10] P. Brucker, A. Drexl, R. Möhring, K. Neumann, and E. Pesch, "Resource-constrained project scheduling: Notation, classification, models, and methods," European Journal of Operational Research, vol. 112, no. 1, pp. 3-41, Jan. 1999, doi: 10.1016/S0377-2217(98)00204-5.

[11] P. Brucker, "Scheduling and constraint propagation," Discrete Applied Mathematics, vol. 123, no. 1, pp. 227-256, Nov. 2002, doi: 10.1016/S0166-218X(01)00342-0.

[12] V. L. Tavares, "A review of the contribution of Operational Research to Project Management," European Journal of Operational Research, vol. 136, no. 1, pp. 1-18, 1 Jan. 2002, doi: 10.1016/S0377-2217(01)00097-2.

[13] C. Schwindt and J. Zimmermann, Handbook on Project Management and Scheduling, Vol.1, 1st ed., Switzerland, Cham: Springer International Publishing, 2015, doi: 10.1007/978-3-319-05443-8.

[14] C. Schwindt and J. Zimmermann, Handbook on Project Management and Scheduling Vol. 2, 1st ed., Switzerland, Cham: Springer International Publishing, 2015, doi: 10.1007/978-3-319-05915-0.

[15] 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.

[16] 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.

[17] S. Hartmann and D. Briskorn, "A survey of variants and extensions of the resource-constrained project scheduling problem," European Journal of Operational Research, vol. 207, no. 1, pp. 1-14, 2010, doi: 10.1016/j.ejor.2009.11.005.

[18] 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, 2022, doi: 10.1016/j.ejor.2021.05.004.

[19] K. Taha, "Methods That Optimize Multi-Objective Problems: A Survey and Experimental Evaluation," IEEE Access, vol. 8, pp. 80855-80878, 2020, doi: 10.1109/ACCESS.2020.2989219.

[20] 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.

[21] A. Schnabel, C. Kellenbrink, and S. Helber, "Profit-oriented scheduling of resource-constrained projects with flexible capacity constraints," Business Research, vol. 11, no. 2, pp. 329-356, 2018, doi: 10.1007/s40685-018-0063-5.

[22] E. B. Tirkolaee, A. Goli, M. Hematian, A. K. Sangaiah, and T. Han, "Multi-objective multi-mode resource constrained project scheduling problem using Pareto-based algorithms," Computing, vol. 101, no. 6, pp. 547-570, 2019, doi: 10.1007/s00607-018-00693-1.

[23] B. H. Tabrizi, "Integrated planning of project scheduling and material procurement considering the environmental

impacts," Computers & Industrial Engineering, vol. 120, pp. 103-115, June 2018, doi: 10.1016/j.cie.2018.04.031.

[24] 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.

[25] R.-M. Chen, "Particle swarm optimization with justification and designed mechanisms for resource-constrained project scheduling problem," Expert Systems with Applications, vol. 38, no. 6, pp. 7102-7111, June 2011, doi: 10.1016/j.eswa.2010.12.059.

[26] 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.

[27] H. Saad, R. Chakrabortty, S. Elsayed and M. Ryan, "Quantum-Inspired Genetic Algorithm for Resource-Constrained Project-Scheduling," IEEE Access, vol. 9, pp. 38488-38502, 2021, doi: 10.1109/ACCESS.2021.3062790.

[28] H. F. Rahman, R. K. Chakrabortty, and M. J. Ryan, "Memetic algorithm for solving resource constrained project scheduling problems," Automation in Construction, vol. 111, March 2020, doi: 10.1016/j.autcon.2019.103052.

[29] K. M. Sallam, R. K. Chakrabortty, and M. J. Ryan, "A two-stage multi-operator differential evolution algorithm for solving Resource Constrained Project Scheduling problems," Future Generation Computer Systems, vol. 108, pp. 432-444, July 2020, doi: 10.1016/j.future.2020.02.074.

[30] 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.

[31] 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, 2020, doi: 10.1016/j.ejor.2019.01.063.

[32] R. Kolisch and S. Hartmann, "PSLIB Single Mode Scheduling Benchmark Dataset," 2006. [Online]. Available: http://www.om-db.wi.tum.de/psplib/getdata_sm.html. [Accessed 11-Oct-2023.].

[33] V. Valls, F. Ballestín, and S. Quintanilla, "A hybrid genetic algorithm for the resource-constrained project scheduling problem," European Journal of Operational Research, vol. 495-508, no. 2, pp. 495-508, 2008, doi: 10.1016/j.ejor.2006.12.033.

[34] J. C. Rivera, L. F. V. Moreno, F. J. S. Díaz, and G. E. Z. Peña, "A hybrid heuristic algorithm for solving the resource constrained project scheduling problem (RCPSP)," Revista EIA, vol. 10, no. 20, pp. 87-100, 2013, doi: 10.14508/reia.2013.10.20.87-100.

[35] F. Berthaut, R. Pellerin, A. Hajji, and N. Perrier, "A path relinking-based scatter search for the resource-constrained project scheduling problem," International Journal of Project Organisation and Management, vol. 10, no. 1, pp. 1-36, 2018, doi: 10.1504/IJPOM.2018.090372.