Evolutionary Algorithm for the Earliness-Tardiness Project Scheduling Problem

Hao Nguyen Thi[1], Huu Dang Quoc[2], Loc Nguyen The[3]
[1] Hung Vuong University, Nong Trang, Viet Tri, Phu Tho
[2] Thuong Mai University, 79 Ho Tung Mau, Cau Giay, Ha Noi, Viet Nam
[3] Hanoi National University of Education
Correspondence: Huu Dang Quoc, huudq@tmu.edu.vn
Received: 29/06/2024, revised: 31/08/2024, accepted: 13/09/2024
Digital Object Identifier: 10.32913/mic-ict-research-vn.v2024.n3.1288

Abstract: Completing too early or too late can cause damage to the project. For example, in transporting goods to the seaports, if the shipping arrives too early, it will take time and cost to stay at the dock. In the other case, shipping too late will reduce the quality of the product and break the contract. The goal of this paper is to minimize the tardiness and earliness of those projects. First, some related studies are briefly introduced, thereby showing the role and practical applications of the MS-RCPSP (Multi-Skill Resource Constrained Project Scheduling Problem) problem and the two factors of earliness and delay in scheduling problems. The roles of earliness and tardiness in scheduling problems are also described. At the same time, the new contribution of this research is also highlighted, which is the proposal for the latest problem E-RCPSP (Earliness tardiness MS-RCPSP). This problem is the first problem in the MS-RCPSP family that considers the sum of earliness and tardiness as the objective, which has never been considered in previous studies. Then, the mathematical model of the proposed E-RCPSP problem, which includes the objective function, project components and constraints is introduced. To solve the proposed problem, two approximate algorithms based on the evolutionary algorithms GA (Genetic Algorithm) and DE (Differential Evolution) are introduced. In the last section of the article, the performance of these two algorithms is verified and compared with each other through experiments conducted on the iMOPSE benchmark dataset.

Keywords: *Network resources management, earliness-tardiness cost, resource-constrained project scheduling, evolutionary algorithms, differential evolution algorithms.*

Tiêu đề: **Thuật toán tiến hóa cho bài toán lập lịch dự án Earliness-Tardiness**

Tóm tắt: Hoàn thành quá sớm hoặc quá muộn có thể gây thiệt hại cho dự án. Ví dụ, trong việc vận chuyển hàng hóa đến cảng biển, nếu hàng hóa đến quá sớm, sẽ mất thời gian và chi phí để lưu lại tại bến tàu. Ở trường hợp khác, việc vận chuyển quá muộn sẽ làm giảm chất lượng sản phẩm và phá vỡ hợp đồng. Mục tiêu của bài báo này là giảm thiểu tình trạng các dự án được hoàn thành quá trễ hay quá sớm. Đầu tiên, bài báo giới thiệu tóm tắt một số nghiên cứu liên quan, qua đó chỉ ra vai trò và ứng dụng thực tế của bài toán MS-RCPSP (Multi-Skill Resource Constrained Project Scheduling Problem) và hai yếu tố Độ sớm và Độ trễ trong bài toán lập lịch. Đồng thời, tác giả nêu bật đóng góp mới của nghiên cứu này, đó là đề xuất một bài toán mới tên là E-RCPSP (Earliness tardiness MS-RCPSP). Bài toán mới này là bài toán đầu tiên trong họ MS-RCPSP đặt tổng Độ sớm và Độ trễ thành hàm mục tiêu, điều chưa từng xuất hiện trong các nghiên cứu trước đây về MS-RCPSP. Sau đó, bài báo giới thiệu mô hình toán học của bài toán E-RCPSP bao gồm hàm mục tiêu, các thành phần của dự án và các ràng buộc, đặc biệt là ràng buộc về thời gian. Để giải quyết bài toán E-RCPSP, bài báo xây dựng hai thuật toán gần đúng dựa trên thuật toán tiến hóa GA (Genetic Algorithm) và thuật toán tiến hóa DE (Differential Evolution). Phần cuối của bài báo trình bày những thực nghiệm trên tập dữ liệu chuẩn iMOPSE nhằm kiểm chứng và so sánh hiệu năng của hai thuật toán đề xuất.

Từ khóa: *Network resources management, earliness-tardiness cost, resource-constrained project scheduling, evolutionary algorithms, differential evolution algorithms.*

I. INTRODUCTION

The project schedule is the deciding factor in project efficiency, but finding the optimal schedule is very difficult because of constraints on task priority and resource limitations. For that reason, MS-RCPSP (Multi-Skill Resource Constrained Project Scheduling Problem) [1], one of the typical project scheduling problems, has long been proven to be an NP-Hard problem. This is a scheduling problem with many practical applications, so it has been of interest to many researchers [2], [3], [4], [5].

The MS-RCPSP scheduling problem assumes that the project is conducted by a given set of resources, each with a different skill level. A given project consists of tasks that are related to each other in order of priority, meaning that only after the parent task is completed can the child task begin to be processed. For a resource to perform a task, it must have the skill required by the task at a skill level greater than or equal to the level required by the task.

This problem was soon proven to be NP-Hard [1], meaning that its optimal solution cannot be found in polynomial time. Therefore, the practical approach often used by researchers is to find an approximate solution with an acceptable deviation.

Myszkowski and colleagues proposed the Ant colony algorithm [3] and GA algorithm [5] to solve the MS-RCPSP problem. In addition, a very important contribution of Myszkowski's research group is the construction of the iMOPSE dataset [2] which is considered a standard dataset and widely used by many other studies.

However, MS-RCPSP has limitations that prevent this problem from fully representing real projects. The limitation is that this problem aims to minimize makespan and cost without taking into account two other important factors that determine project efficiency: tardiness and earliness.

In the family of scheduling problems, tardiness and earliness are common parameters that have been widely defined and used [6], [7], [8].

Tardiness is the time elapsed between the deadline and the finishing time of a task, which represents the delay in task execution. The earliness of a task is defined as:

$$E_i = max(0, f_i - h_i) \quad (1)$$

We denote the prescribed due date for the task by h_i, in other words, f_i is the deadline for completion of the task.

With the above notation, tardiness is defined as:

$$T_i = max(0, h_i - f_i) \quad (2)$$

Let's look at a specific example of the harmful effects of completing a project early in the frozen food industry. If the product is made too early, the manufacturer must bear additional costs for renting a warehouse for storage. In addition, when products reach users, their shelf life is reduced.

In the past, there have been studies on tardiness and earliness applied to scheduling problems, but these two factors have never been applied to a multi-skill scheduling problem like MS-RCPSP. The new contribution of this paper compared to previous studies is to consider delay and earliness as the objective function for the MS-RCPSP problem, instead of makespan or cost like previous studies [3], [4],[5].

This paper proposes a novel problem called E-RCPSP (Earliness tardiness MS-RCPSP), which is the problem of finding a project schedule that minimizes total tardiness and earliness. Figure 1 depicts a project schedule consisting of 5 tasks, whereby the total delay and advance of this schedule is 6 + 8 + 5 + 15 + 6 = 40.

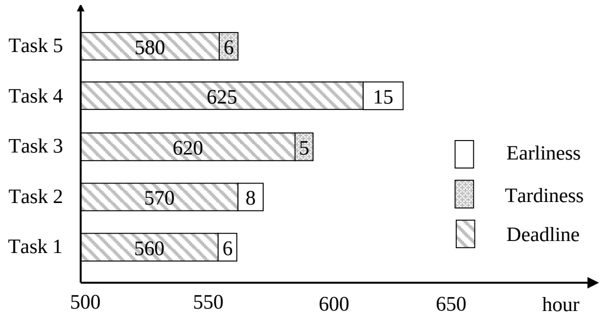

Figure 1. Tardiness and earliness of a project consisting of 5 tasks.

The rest of the paper is structured as follows. Section 2 introduces previous research related to the MS-RCPSP problem as well as the concepts of delay and earliness. Section 3 presents the mathematical model of the E-RCPSP problem. Section 4 introduces two solutions to solve the proposed problem including the GA-E algorithm and DE-E differential evolution algorithm. To verify the effectiveness of the two proposed algorithms, section 5 presents experimental results, and analyzes and evaluates them to clarify the performance of those algorithms.

II. RELATED WORKS

1. Approximate Algorithm for MS-RCPSP Problem

Myszkowski solved the original version of the MS-RCPSP problem using a variety of methods. Initially, he used a heuristic method of sorting tasks based on the order of their processing time, and then assigning resources in that order. Later, Myszkowski turned to meta-heuristic algorithms such as the Ant Colony algorithm [3] and the GA algorithm [5] and obtained higher-quality solutions. In addition to proposing algorithms, another important contribution of Myszkowski et al. is the iMOPSE dataset [2]. This dataset is recognized as a standard benchmark and is therefore widely used by many other studies to verify the performance of algorithms for the MS-RCPSP problem.

A variant of MS-RCPSP was named MMSRCPSP (Multi-mode Multi-skilled Resource-Constrained Project

Scheduling Problem) by Hosseinian [9]. This variant of the MS-RCPSP problem adds the constraint that each task can only be executed in certain modes, and the mode cannot be changed once the task's execution has been started.

To find an efficient schedule for the MMSRCPSP problem, Hosseinian et al. [10] used a traditional GA algorithm combined with a decision-making method based on the Shannon-entropy information measure to search for better schedules for the next generation. Then, with the original MS-RCPSP problem, Hosseinian turned to using the Dandelion Algorithm [11] and verified its effectiveness on the iMOPSE dataset. In this research, Hosseinian solved the multi-objective MS-RCPSP problem with two objectives: makespan and execution cost.

Similar to Hosseinian, H. Davari-Ardakani [1] also studied a multi-objective variant of the MS-RCPSP problem with an objective function including two components: project implementation time and cost. In the proposed problem, called MSPSP, H. Davari-Ardakani only considers projects with the following two characteristics:

- Project implementation time is arbitrary. The project can even be done in the evening or on weekends.
- Energy costs are enormous, equal to employee salaries.

To solve the MS-RCPSP problem in a Grid environment, Youni and colleagues proposed a hybrid heuristic algorithm based on a task arrangement mechanism [12]. The works of Maghsoudlou [13] and Bibiks [14] are similar in that they apply the Cuckoo Search algorithm to search for the optimal schedule of multi-risk projects based on three different evaluation objectives. Zhu et al. proposed an evolutionary algorithm based on the multi-verse algorithm [15].

2. Solutions to Minimize Earliness and Tardiness

Mario Vanhoucke and colleagues [16] proposed the RCPSPWET problem, an extension of the RCPSP problem. The goal of the RCPSPWET problem is to minimize the total Earliness-Tardiness penalty cost. Mario Vanhoucke uses prioritization constraints between tasks and constraints on continuously available renewable resources. The authors solved the RCPSPWET problem using the search tree algorithm and branching algorithm.

Yazdan Khoshjahan [6] proposed a new problem, named RCPSP-DET, whose goal is to minimize the net present value of project Earliness-Tardiness penalties. The author solves this problem through a meta-heuristic algorithm.

Maarten Otten [17] minimized the Earliness-Tardiness cost across multiple machines for surgical scheduling. The author has proposed a solution to minimize two factors: temporary time and overtime of doctors and medical staff.

Saheed Akande [18] proposed three versions of the RCPSP problem related to earliness and tardiness. The first version of the problem deals with the total costs incurred due to tardiness work. The second version of the problem maximizes earliness and minimizes tardiness during project execution. The third version of the problem minimizes the linear objective function of earliness and tardiness. The results obtained from the three versions of the problem represent a measure of the degree of penalty arising from each type of problem. In another paper, Saheed Akande solved the multi-objective scheduling problem, including minimum total completion time, maximum tardiness, and earliness [19].

The general limitation of the above studies is that they only aim to reduce the total early-late penalty cost of the RCPSP problem; There is no solution for minimizing the early-late penalty on the MS-RCPSP problem.

III. PROBLEM FORMULATIONS

Given a set of tasks (denoted by Job) to be executed that are related to each other in order of priority, meaning that the child task is only allowed to start after the parent task has been completed. Given a set of resources (denoted by Res), each resource possesses different skills. To execute task Job_i, resource Res_i must have the skills that match Job_i's requirements and one skill level greater than or equal to the skill level required by Job_i.

The goal of the E-RCPSP problem is to find a project schedule that minimizes total earliness and tardiness. Earliness and tardiness are defined by formulas (1) and (2), respectively. E-RCPSP can be conceptually formulated based on the following notations:

- Tas_i: set of tasks that need to be completed before the time task i is executed.
- Fis: the set of skills.
- Fis^i: the subset of skills owned by resource i, $Fis^i \subseteq Fis$.
- Fis_i: skill i.
- tim_j: the time required to perform task j.
- Res: the set of resources.
- Res^m: the set of resources that could handle task m, $Res^m \subseteq Res$.
- Res_i: resource i.
- Job: the set of all resources.
- Job^k: the set of tasks that resource k could handle, $Job^k \subseteq Job$.
- Job_i: task i.
- req^i: set of skills required to perform the task i.
- Sta_k, Clo_k: start time and finish time of task k.

- $Var_{u,v}^{t}$: a boolean variable; it equals 1 means that task u will be executed by resource v at time t, it equals 0 in other cases.
- lev_i: the level of skill i.
- gen_i: type of skill i.
- mak: makespan (execution time) of schedule.
- TSc: a candidate schedule.
- TSc_{all}: the set of candidate schedules.
- $f(TSc)$: makespan (execution time) of scheduled TSc.
- cou: number of tasks, num: number of resources.

E-RCPSP problem could be defined as follows:

$$f(TSc) \quad \rightarrow \quad min$$

where

$$f(TSc) = \sum_{i=1}^{n} (e_i E_i + t_i T_i) \tag{3}$$

e_i and t_i are the penalty units for the earliness and tardiness times of task i, respectively. E_i , T_i are the earliness and the tardiness of task i, respectively.

Subject to:

$$Fis^k \neq \emptyset \quad \forall Res_k \in Res \tag{4}$$

$$tim_i \geq 0 \quad \forall Job_i \in Job \tag{5}$$

$$Clo_i \geq 0 \quad \forall Job_i \in Job \tag{6}$$

$$Clo_i \leq Clo_k - tim_k \quad \forall Job_k \in Job, k \neq 1, Job_i \in Task_k \tag{7}$$

$$\forall Job_i \in Job^k, \exists Fis_q \in Fis^k : gen_{Fis_q} = gen_{r^i}, \quad lev_{F_q} \geq lev_{r^i} \tag{8}$$

$$\forall Res_k \in Res, \forall q \in mak : \sum_{i=1}^{n} Var_{i,k}^{q} \leq 1 \tag{9}$$

$$\forall Job_i \in Job, \exists! q \in [0, mak], \quad \exists! Res_k \in Res : Var_{i,k}^{q} = 1 \;\; with \; Var_{i,k}^{q} \in \{0, 1\} \tag{10}$$

Note that:

- Formulation (4) means that every resource has at least one skill.
- Formulation (5, 6) means that task execution time is greater than or equal to 0.
- Formulation (7) forces the parent task to be finished before the start time of the children's task.
- Formulation (8) means that for every task, there is always at least one resource that has enough skill level to handle that task.
- Formulation (9) ensures that each resource (k) can only perform at most one task at any time (q).
- Formulation (10) means that each task is assigned to a single resource and executed only by that resource.

IV. PROPOSED ALGORITHM

1. DE algorithm for E-RCPSP problem

Differential Evolution (DE) is the information-driven evolutionary algorithm introduced by R. Storn and K. Price [20]. DE is the directed evolutionary algorithm that uses mutation methods to find better solutions in the next generation. DE has been applied effectively to solve NP-Hard problems.

This paper proposes a DE-E which is algorithm to solve the E-RCPSP problem, its operating steps are presented in Figure 2 and the source codes are detailed in Algorithm 1 below.

Algorithm 1: DE-E algorithm

Data: N (size of population), numGen (number of generations)
Result: the best schedule
Load and valid dataset; count = 0; P = initialize(N);
gbest = fbest(P); */*the best individual in the population*/*
while *count < numGen* **do**
  for *i < N* **do**
    $x_{r_1} \neq x_{r_2} \neq x_{r_3} \neq x_i$; */*Random different individuals*/*
    $F \leftarrow rand(0, 1)$; */*mutation constant*/*
    $v_i(t) = x_{r_1} + F * (x_{r_2} - x_{r_3})$; */*mutation*/*
    I_{rand} = randi(1,tCount); */*I_{rand} ensures $x_i \neq v_i$*/*
    CR ∈ [0, 1]; */*probability of crossover*/*
    for *j < tCount* **do**
      if *i = I_{rand} OR rand(0,1) <= CR* **then**
        $u_{i,j} = v_{i,j}$;
      end
      else
        $u_{i,j} = x_{i,j}$;
      end
    end
    $u1$ = simplebound(u); $z = f(u1)$; $fxi = f(x_i)$;
    */*select population*/*
    if *fxi < z* **then**
      $P[i] = u1$;
    end
    */*update makepan*/*
    if *z<makespan* **then**
      $makespan = z$; ;
      CopyMPS($u1$);
    end
    if *z <fitness[i]* **then**
      $fitness[i] = z$; */*update fitness*/*
    end
  end
  */*update gbest*/*
  gbest = fbest(P); count = count +1;
end
return gbest;

Where, some functions are used in Algorithm 1:

f: objective function of E-RCPSP problem.

$fbest$: the function returns the best individual in the population. $tCount$: number of tasks.

$simplebound$: validate current particle/individual.

$CopyMPS$: update population.

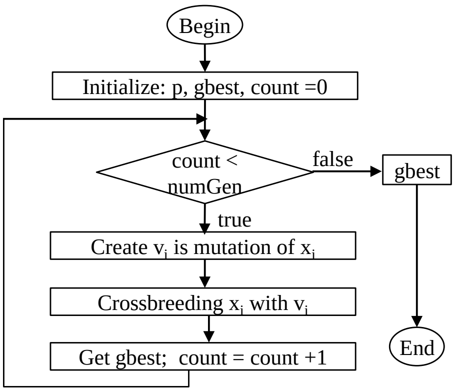

Figure 2. DE-E algorithm for the E-RCPSP problem.

2. GA algorithm for E-RCPSP problem

Genetic algorithm (GA) is a computer science technique that seeks to find the best solution to combination optimization problems. GA is an evolutionary algorithm that applies principles of evolution such as mutation, natural selection, and crossover. This paper proposes a GA-E, which is a GA algorithm for E-RCPSP problem, its operating steps are presented in Figure 3. GA-E algorithm are details as follows.

Algorithm 2: GA-E algorithm

```
Data: N (size of population), numGen (number of generations)
Result: the best schedule
Load and valid dataset;
count = 0;     p = initialize(N);     gbest = fbest(p);
while count < numGen do
    initialize(fitness);
    /*Evaluate fitness*/
    for i < N do
        fitness[i] = f(p[i]);
        if finess[i]<makespan then
            makespan = fitness[i];     CopyMPS(p[i]);
        end
    end
    /*select population*/
    for i < N do
        ichon1 = rand(1,N);     ichon2 = rand(1,N);
        if fitness[ichon1] > fitness[ichon2] then
            Pselect [i] = p[ichon1];
        end
        else
            Pselect [i] = p[ichon2];
        end
    end
    /* Crossover */
    Crossbreeding of randomly selected individuals;
    /*mutation*/
    muta = 0.1*rand(0,1);     /*mutation probability*/
    Create mutation with mutation probability "muta"
     and get Pselect;
    p = Pselect;     count = count+1;
end
return fbest(p);
```

Some functions are used in algorithm 2:

f: objective function of E-RCPSP problem.

$fbest$: the function returns the best individual in the population.

$CopyMPS$: update population.

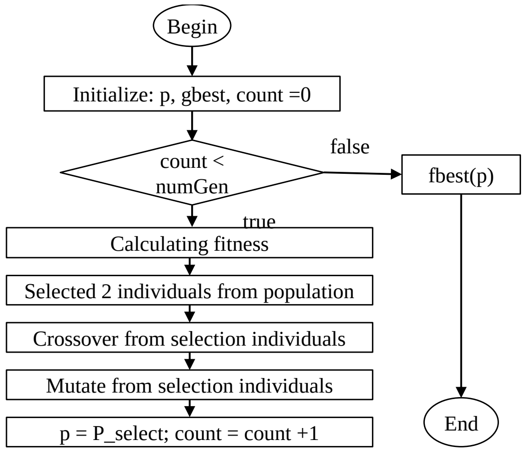

Figure 3. Proposed GA-E algorithm for E-RCPSP problem

V. PERFORMANCE EVALUATION

1. Experimental settings

To verify the performance of the proposed algorithms, experiments are conducted on C# Visual Studio 2022, processor Core i5 2.4GHz, RAM 6GB, OS Windows 11.

Experiments were conducted iMOPSE dataset [2]. As detailed in Table I, iMOPSE's data is stored in the file .def, which includes the following information fields:

- Number of the project's tasks and resources.
- Priority relationship between tasks.
- The number of relationships between tasks.
- Number of skills owned by resources.
- The subset of skills owned by each resource.

2. Experimental Results

To find the optimal schedule for the E-RCPSP problem, this article proposes two algorithms DE-E and GA-E. Experiments were conducted to evaluate the performance of the proposed algorithms. The experimental results of the two algorithms are described in Table II and Table III, respectively, in which column E represents the earliness value, column T represents the tardiness value and column ET represents the value of the objective function, all of these values are in seconds.

Figure 4 shows the experimental results of the GA-E algorithms with the number of generations being 1000,

Table I
IMOPSE DATASET

No.	Dataset	Name	Tasks	Resources	Precedence Relations	Skills
1	iM1	100_5_22_15	100	5	22	15
2	iM2	100_5_46_15	100	5	46	15
3	iM3	100_5_48_9	100	5	48	9
4	iM4	100_5_64_15	100	5	64	15
5	iM5	100_5_64_9	100	5	64	9

Table II
EXPERIMENTAL RESULTS OF THE GA-E ALGORITHM

Dataset	1000 generations			3000 generations			5000 generations		
	E(s)	T(s)	ET(s)	E(s)	T(s)	ET(s)	E(s)	T(s)	ET(s)
iM1	4148	3607	**7755**	5992	3210	**9202**	6501	2203	**8704**
iM2	3547	1936	**5483**	4850	1017	**5867**	3113	2359	**5472**
iM3	2543	2332	**4875**	2219	2731	**4950**	3711	1236	**4947**
iM4	2889	1836	**4725**	1197	2892	**4089**	2642	1304	**3946**
iM5	1052	1950	**3002**	1363	1479	**2842**	1102	1408	**2510**

Table III
EXPERIMENTAL RESULTS OF THE DE-E ALGORITHM

Dataset	1000 generations			3000 generations			5000 generations		
	E(s)	T(s)	ET(s)	E(s)	T(s)	ET(s)	E(s)	T(s)	ET(s)
iM1	459	513	**972**	530	114	**644**	908	797	**1705**
iM2	860	1058	**1918**	1965	378	**2343**	539	541	**1080**
iM3	1548	1030	**2578**	1358	1153	**2511**	1151	757	**1908**
iM4	1594	1497	**3091**	557	504	**1061**	575	1221	**1796**
iM5	2663	1808	**4471**	2665	1777	**4442**	2681	1732	**4413**

3000, and 5000. Figure 5 shows the experimental results of DE-E with the corresponding number of generations. Through this comparison, it can be seen that the GA-E algorithm is more effective when performed on datasets that have a high number of precedence relations and a small number of skill levels. In other words, for the same number of tasks and resources, on a dataset with a higher number of precedence relations and a lower number of skill levels, the GA-E algorithm finds better results than DE-E.

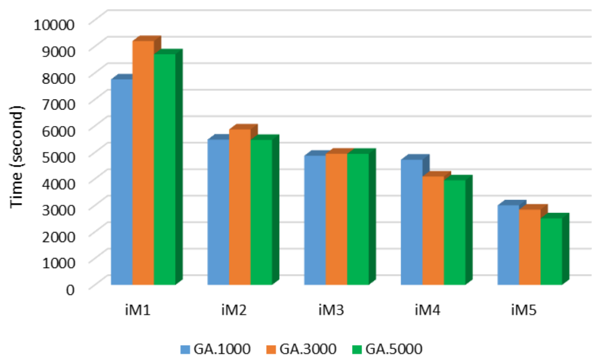

Figure 4. The efficiency of the GA-E algorithm

Figure 5 depicts that the DE-E algorithm is more effective when the dataset has a small number of precedence relations and a large number of skill levels. That is, with the same number of tasks and resources, on a dataset with a smaller number of precedence relations and a larger number of skill levels, the results of the DE-E algorithm are better than the GA-E.

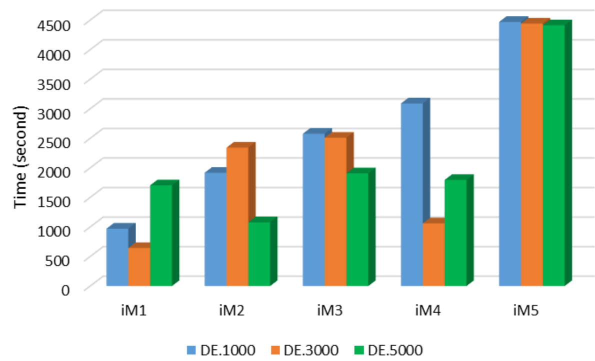

Figure 5. The efficiency of the DE algorithm

Experimental results with different numbers of generations (1000, 3000, 5000) show that the results of the DE-E algorithm are better than GA-E when performed on the iM1, iM2, iM3, and iM4 datasets. In contrast, the GA-E algorithm outperforms DE-E on the remaining datasets. A comparison of experimental results of GA-E and DE-E algorithms with 1000 generations is depicted in Figure 6:

• With the iM1 dataset, the ET value of DE-E is only 12.53% of the GA-E.

• With the iM2, iM3, iM4, iM5 dataset, the ET value of DE-E is 34.98%, 52.88%, 65.42%, and 148.93% of GA-E, respectively.

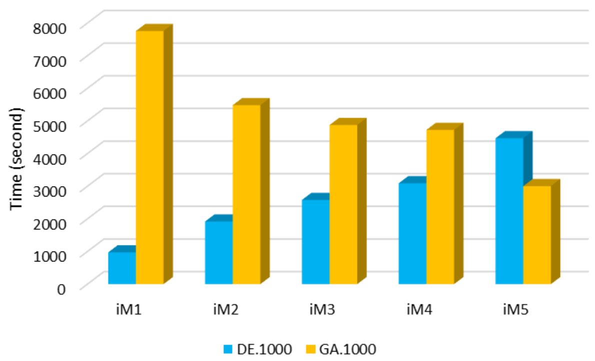

Figure 6. The effectiveness of GA-E and DE-E when experimenting with 1000 generations

A comparison of experimental results of GA-E and DE-E algorithms with 3000 and 5000 generations is depicted in Figure 7 and Figure 8, respectively.

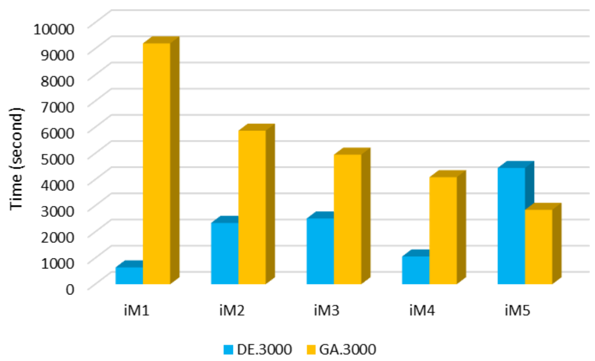

Figure 7. The effectiveness of GA-E and DE-E when experimenting with 3000 generations

A comparison of experimental results of GA-E and DE-E algorithms with 3000 generations is depicted in Figure 7:

• With the iM1 dataset, the ET value of DE-E is only 7.00% of the GA-E.

• With the iM2, iM3, iM4, iM5 dataset, the ET value of DE-E is 39.94%, 50.73%, 25.95%, and 156.30% of GA-E, respectively.

A comparison of experimental results of GA-E and DE-E algorithms with 5000 generations is depicted in Figure 8:

• With the iM1 dataset, the ET value of DE-E is only 19.59% of the GA-E.

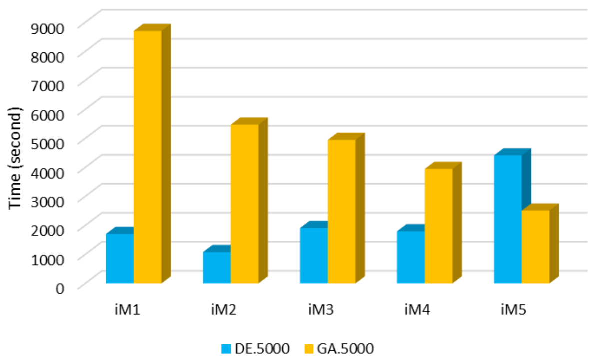

Figure 8. The effectiveness of GA-E and DE-E when experimenting with 5000 generations

• With the iM2, iM3, iM4, iM5 dataset, the ET value of DE-E is 19.74%, 38.57%, 45.51%, and 175.82% of GA-E, respectively.

VI. CONCLUSION AND FUTURE WORK

Earliness and tardiness are very important factors that allow evaluation of the effectiveness of schedules. In the family of scheduling problems, the MS-RCPSP problem has many practical applications, so it attracts the attention of many researchers. However, the application of earliness and tardiness to the MS-RCPSP problem has never been mentioned in previous studies.

This research has built the foundation for project scheduling so that the execution of tasks matches or does not deviate much from the scheduled deadline. We proposed a new problem called E-RCPSP and officially stated its mathematical model. Two evolutionary algorithms based on GA and DE have been built to find the optimal schedule for the proposed problem. Experimental results show that among the two proposed algorithms, each has its own advantages, meaning that no algorithm outperforms the other on all datasets.

In the future, the authors hope to find a comprehensive solution capable of finding the optimal schedule in all cases. To achieve that goal, we plan to apply deep reinforcement learning (DRL) to help the proposed algorithm learn from the knowledge of previous generations to make correct decisions for the next generations.

ACKNOWLEDGMENT

This research was supported by the Ministry of Education and Training, Viet Nam, under the science and technology project no. B2024-SPH-14 supervised by the Hanoi National University of Education (HNUE), Viet Nam.

REFERENCES

[1] H. Najafzad, H. Davari-Ardakani, R. Nemati-Lafmejani, "Multi-skill project scheduling problem under time-of-use electricity tariffs and shift differential payments", *In Energy Journal*, vol. 168, pp. 619-636, Elsevier, 2019.

[2] P.B. Myszkowski, M. Laszczyk, I. Nikulin, M. Skowro, "iMOPSE: a library for bicriteria optimization in Multi-Skill Resource-Constrained Project Scheduling Problem", *In Soft Computing Journal*, 23: 32397, 2019.

[3] P.B. Myszkowski, M. Skowroński, L. Olech, K. Oślizło, " Hybrid Ant Colony Optimization in solving Multi–Skill Resource–Constrained Project Scheduling Problem", *In Soft Computing Journal*, Volume 19, Issue 12, pp.3599–3619, 2015.

[4] S. Javanmard, B. Afshar-Nadjafi, S.T. Niaki, "Preemptive multi-skilled resource investment project scheduling problem: Mathematical modeling and solution approaches," *In Computers & Chemical Engineering*, 96, pp. 55-68, 2017.

[5] Skowronski M., Myszkowski P.B., " Specialized genetic operators for Multi–Skill Resource–Constrained Project Scheduling Problem", *In 19th Inter, Conference on Soft Computing - Mendel 2013*, pp. 57-62, 2013

[6] Y. Khoshjahan, A.A. Najafi, B.A. Nadjafi, "Resource constrained project scheduling problem with discounted earliness-tardiness penalties: Mathematical modeling and solving procedure", *In Computers & Industrial Engineering*, 2013.

[7] V. A. Strusevich, K. Rustogi, "Scheduling with Times-Changing Effects and Rate-Modifying Activities", *In Springer International Publishing*, 2017.

[8] Kramer, A., Subramanian, A., " A unified heuristic and an annotated bibliography for a large class of earliness–tardiness scheduling problems", *In Journal of Scheduling*, vol. 22, 2019.

[9] A.H. Hosseinian, V. Baradaran, " An Evolutionary Algorithm Based on a Hybrid Multi-Attribute Decision Making Method for the Multi-Mode Multi-Skilled Resource-Constrained Project Scheduling Problem", *In Journal of Optimization in Industrial Engineering*, 12.2, pp. 155-178, 2019.

[10] A.H. Hosseinian, V. Baradaran," Detecting communities of workforces for the multi-skill resource-constrained project scheduling problem: A dandelion solution approach", *In Journal of Industrial and Systems Engineering*, pp. 72-99, 12.2019.

[11] A.H. Hosseinian, V. Baradaran," P-GWO and MOFA: two new algorithms for the MSRCPSP with the deterioration effect and financial constraints (case study of a gas treating company)", *In Applied Intelligence*, 50, pp. 2151-2176, 2020.

[12] Younis, Muhanad Tahrir, and Shengxiang Yang, "Hybrid meta-heuristic algorithms for independent job scheduling in grid computing", *In Applied soft computing*, vol.72, pp. 498-517, 2018.

[13] Maghsoudlou, Hamidreza, Behrouz Afshar-Nadjafi, and Seyed Taghi Akhavan Niaki," Multi-skilled project scheduling with level-dependent rework risk; three multi-objective mechanisms based on cuckoo search", *In Applied Soft Computing*, vol. 54, pp. 46-61, 2017.

[14] Bibiks, Kirils, et al.," Improved discrete cuckoo search for the resource-constrained project scheduling problem", *In Applied Soft Computing*, vol. 69, pp. 493-503, 2018.

[15] Zhu, Lei, Jian Lin, Zhou-Jing Wang, "A discrete oppositional multi-verse optimization algorithm for multi-skill resource constrained project scheduling problem", *In Applied Soft Computing*, vol. 85, 2019.

[16] Mario Vanhoucke, Erik Demeulemeester, and Willy Herroelen," An Exact Procedure for the Resource-Constrained Weighted Earliness–Tardiness Project Scheduling Problem", *In Kluwer Academic Publishers. Manufactured in The Netherlands*, 2001.

[17] Maarten Otten, Aleida Braaksma, Richard J. Boucherie, " Minimizing Earliness/Tardiness costs on multiple machines with an application to surgery scheduling", *Operations Research for Health Care*, vol. 22, 2019.

[18] S. Akande, G. Ajisegiri, " Three classes of Earliness-Tardiness (E/T) scheduling problems", *In Fuel Communications*, vol. 10, 2022.

[19] S. Akande, Ajisegiri GO, " Single machine multi-criteria scheduling problem: total completion time, maximum lateness, and maximum earliness performance measures", *In International Journal of Planning and Scheduling*, vol. 3, 2021.

[20] K.Price, R.Storn, J.Lampinen: "Differential Evolution - A Practical Approach to Global Optimization", *In Springer, Berlin, Germany*, 2005.

Hao Nguyen Thi received Bachelor degree at Hung Vuong University, Viet Nam, in 2009. She received M.S. degree at Ha Noi University of Science and Technology, Viet Nam, in 2013. She worked at Hung Vuong University, Phu Tho, Viet Nam from 2009. Her research interests include optimization algorithms, approximation algorithms, deep learning.
Email: haont@hvu.edu.vn

Huu Dang Quoc received Bachelor and M.S. degree in School of Information Technology, Vietnam National University, Ha Noi, Viet Nam, in 2000 and 2015. He received Ph.D. degree in Military Institute of Science and Technology, Ha Noi, Vietnam, Viet Nam, 2022. He worked at Thuong Mai University, Ha Noi, Viet Nam from 2006. His research interests include Computer Network and Software Engineering, Evolution Algorithm, Optimization Algorithm.
Email: huudq@tmu.edu.vn

Loc Nguyen The received Bachelor and M.S. degree in School of Information and Communication Technology, Hanoi University of Science and Technology, Viet Nam, in 1998 and 2001, respectively. He received Ph.D. degree in School of Information Science, Japan Advanced Institute of Science and Technology, Japan, 2007. He worked at Hanoi National University of Education (HNUE), Ha Noi, Viet Nam from 1997 and is currently a professor, head of Department of Computer Engineering, Faculty of Information Technology, HNUE. His research interests include Optimization Algorithms, Approximation Algorithms, IoT, Networking Infrastructure.
Email: locnt@hnue.edu.vn