MIMEM: The Hybrid Metaheuristic Method for the Resource Constrained Scheduling Problem

Huu Dang Quoc[1], Loc Nguyen The[2]

[1] Thuong Mai University, 79 Ho Tung Mau, Cau Giay, Ha Noi, Viet Nam

[2] Hanoi National University of Education

Correspondence: Loc Nguyen The, locnt@hnue.edu.vn

Received: 16/04/2023, revised: 21/06/2023, accepted: 30/6/2023

Digital Object Identifier:: 10.32913/mic-ict-research-vn.v2023.n1.1218

Abstract: The Multi-Skill Resource-Constrained Project Scheduling Problem (MS-RCPSP) is an NP-Hard problem that involves scheduling activities while accounting for resource and technical constraints. This paper aims to present a novel hybrid algorithm called MIMEM, which combines the Memetic algorithm with the Migration method to solve the MS-RCPSP problem. The proposed algorithm utilizes the migration method to identify local extremes and then relocates the population to explore new solution spaces for further evolution. The MIMEM algorithm is evaluated on the iMOPSE benchmark dataset, and the results demonstrate that it outperforms. The solution of the MS-RCPSP problem using the MIMEM algorithm is a schedule that can be used for intelligent production planning in various industrial production fields instead of manual planning.

Keywords: *memetic algorithm, evolutionary computing, optimization, MS-RCPSP problem.*

Tiêu đề: **Thuật Toán Lai MIMEM Giải Bài Toán MS-RCPSP**

Tóm tắt: Bài toán lập lịch thực hiện dự án với tài nguyên giới hạn và đa kỹ năng (Multi Skill-Resource Constrained Project Scheduling Problem: MS- RCPSP) là một bài toán thuộc lớp NP-Hard, mục tiêu của bài toán này là tìm ra lịch biểu để thiết lập các tài nguyên thực hiện mọi công việc của dự án trong thời gian ngắn nhất. Bài báo trình bày thuật toán lai ghép mới để giải bài toán này, thuật toán mới gọi là MIMEM, là sự kết hợp giữa thuật toán Memetic với phương pháp Migration. Trong quá trình tiến hóa của thuật toán Memetic, khi quần thể rơi vào cực trị địa phương, phương pháp Migration sẽ được sử dụng để di chuyển cả quần thể sang vùng không gian nghiệm khác giúp quần thể thoát khỏi cực trị địa phương và tiếp tục quá trình tiến hóa. Để kiểm chứng tính hiệu quả của thuật toán đề xuất, bài báo đã thực nghiệm trên bộ dữ liệu iMOPSE, kết quả cho thấy MIMEM mang lại các kết quả tốt đối với bài toán MS-RCPSP. Lời giải của bài toán là lịch biểu thể hiện việc bố trí các tài nguyên thực hiện các công việc của dự án, do vậy, có thể sử dụng để lập kế hoạch điều phối sản xuất tự động thay vì các phương pháp làm truyền thống trước đây.

Từ khóa: *thuật toán Memetic, tính toán tiến hóa, tính toán tối ưu, bài toán MS-RCPSP*

I. INTRODUCTION

The MS-RCPSP is a complex scheduling problem widely studied in operations re-search and project management. It is a type of project scheduling problem that considers the order in which tasks should be performed and the resources required to perform each task, such as labor, equipment, and materials. The goal of solving the MS-RCPSP is to minimize the project's completion time, cost, or both (multi-objective [1, 2]) while ensuring that all resource constraints are satisfied. It is a widely studied problem with numerous real-world applications, including construction, software development, and manufacturing.

The MS-RCPSP problem has an important constraint: a resource can only perform the task if it possesses the right skill type and the skill level is greater than or equal to the required skill level. Moreover, resources have many skill types and skill levels, so it is necessary to plan for allocating resources to perform practical tasks to improve the efficiency of project implementation. MS-RCPSP is a problem of class NP-Hard, so the optimal solution cannot be found in polynomial time. However, metaheuristic methods can solve the problem to find approximate solutions

in less time. In order to find a suitable solution for the MS-RCPSP problem, this paper proposes the MIMEM algorithm developed from the Memetic algorithm [3–5] integrated with the Migration method.

The rest of this paper is presented as follows: Part II introduces some research on evolutionary methods and the Memetic algorithm to solve NP-Hard problems. Part III describes the mathematical statement of the problem. Part IV presents the MIMEM algorithm to solve the MS-RCPSP problem, a hybrid algorithm between Memetic and Migration to improve the solution's efficiency by finding and detecting local extremes and moving populations away from that. Part V presents the implementation of experiments to verify the MIMEM algorithm and analyze and evaluate the quality of the solution. Finally, section VI summarizes the paper's main results and future research directions.

II. RELATED WORKS

1. Approximate methods to solve the Scheduling problems

The most well-known scheduling problems are NP-Hard, meaning that their solutions cannot be found in polynomial time. Therefore, it is common to use approximate, evolutionary algorithms to find an acceptable solution quickly often use evolutionary, approximate algorithms to find an acceptable solution quickly. Many algorithms have been proposed for the scheduling problem, such as GA [8–10], PSO [11–13]], DE[14, 15], ANT [16],...

Garg et al. [17] studied and solved the scheduling problem in the cloud computing environment to improve the energy efficiency of virtual machines. The authors have proposed the PESVMC algorithm to schedule the execution of workflows. In [18], the authors study the scheduling algorithm combined with deep learning to optimize the environment that combines fog-cloud computing. The multi-objective scheduling problem is also studied and solved by parallel processing algorithms by the author of the [19] paper.

Besides, many research groups also use the GA algorithm to solve the scheduling problem. In [9], the authors use the GA integrated with the Immune to solve the NPV-Based RCPSP problem. The algorithm is also applied with two additional variables to improve efficiency: local search and forward-backward improvement. Also, using a hybrid GA algorithm to solve problems in a cloud computing environment, Hussain and colleagues [10] improved GA; this algorithm combines a blend of a dynamic round-robin and a local search algorithm and has been deployed experimentally and verified in the Cloudsim environment.

Another effective metaheuristic commonly used to solve the scheduling problem is the Particle Swarm Optimization (PSO) method. In [11], the authors studied the mod-ified PSO algorithm combined with deep learning models to apply in the image diagnosis of eye diseases. The application of modified PSO is used for both supervised and unsupervised machine learning models to tune the CLAHE parameter. In [12], PSO was studied to reduce energy consumption and the working time of stand-alone tasks in a cloud computing environment. The proposed algorithm, EE-APSO, also permits the population to eliminate the local extremes and expand the search space. In a grid computing environment, the authors use a three-PSO balanced PSO algorithm[13] to schedule workflows and improve the utilization of system resources.

Besides the above solutions, many research groups also use Differential Evolution (DE) algorithm. In [14], the authors proposed an effective method of selecting the presenter feature to improve performance and reduce the processing time of IDS systems. The proposed method uses a DE algorithm to select valuable features during an extreme machine learning classifier is applied after feature selection to evaluate the selected features. Experiments were performed on the NSL-KDD dataset. In [15], pseudo-descriptors focus on solving the individual mutation strategy to improve discoverability, convergence accuracy, and convergence speed. The authors give out a new algorithm called NBOLDE with neighborhood mutation and opposition-based learning operators. The proposed NBOLDE includes new evaluation parameters and weighting factors in the neighborhood model to propose a new neighborhood strate-gy. On this basis, a new neighbor mutation strategy based on DE to best. Usually, to experiment with the proposed algorithms, the authors often use two primary datasets, PSLIB[20] and iMOPSE [6, 7], which P.B. Myszkowski suggested

2. Memetic algorithm

Memetic algorithms (MAs)[3–5] are optimization algorithms that incorporate both evolutionary and local search procedures to solve complex optimization problems. MAs have gained increasing popularity in recent years due to their ability to over-come the limitations of traditional evolutionary algorithms (EAs) by incorporating problem-specific knowledge to guide the search process.

In a memetic algorithm, individuals in the population are encoded as solutions to the optimization problem and evolve through selection, crossover (recombination), and mutation. The genetic algorithm generates new candidate solutions, while the local search method is used to refine the solutions generated by the genetic algorithm.

Using local search methods in memetic algorithms can lead to faster convergence and improved solution quality compared to traditional genetic algorithms. This is because local search methods can exploit the structure of the problem space to find better solutions quickly. At the same time, the genetic algorithm provides a mecha-nism for exploring the search space and avoiding getting trapped in local optima.

However, there are also some disadvantages to using MAs. One of the main draw-backs is that MAs can be computationally expensive, as they require both global and local search procedures. Another challenge is designing effective, problem-specific, and efficient local search procedures can be complex. Furthermore, MAs can also suffer from premature convergence, where the algorithm becomes stuck in a local optimum and cannot escape finding the global optimum.

III. MS-RCPSP PROBLEM

The MS-RCPSP has a wide range of real-world applications, such as in project management, scheduling of manufacturing processes, and resource allocation in large-scale engineering projects. The ability to effectively solve the MS-RCPSP has significant implications for the efficiency and success of these real-world applications.

The MS-RCPSP problem can be conceptually formulated based on the notations in Table I.

The MS-RCPSP problem could be state as follow:

$$f(P) \rightarrow min \tag{1}$$

Subject to the following constraints:

$$S^k \neq \oslash \quad \forall L_k \in L \tag{2}$$

$$t_j \geq 0 \quad \forall W_j \in W \tag{3}$$

$$E_j \geq 0 \quad \forall W_j \in W \tag{4}$$

$$E_i \leq E_j - t_j \quad \forall W_j \in W, j \neq 1, W_i \in C_j \tag{5}$$

$$\forall W_i \in W^k \quad \exists S_q \in S^k : g_(S_q) = g_(r^i) \text{ và } h_(S_q) \geq h_(r^i) \tag{6}$$

$$\forall L_k \in L, \forall q \in m : \sum_{i=1}^{n} A_{i,k}^q \leq 1 \tag{7}$$

$$\forall W_j \in W \exists! q \in [0-m], !L_k \in L : A_{j,k}^q = 1; \ |A_{j,k}^q \in \{0;1\} \tag{8}$$

In the MS-RCPSP problem, each task has additional skill requirements of the renew-able resource needed to perform. Each resource is also divided into different skill types and skill levels. Figure 1 depicts an example of the resource skill constraints required to complete a task.

IV. PROPOSED ALGORITHM

This section describes the proposed evolutionary algorithm for the MS-RCPSP prob-lem named MIMEM, a variation of the Memetic algorithm [3–5] enhanced with the Migration method.

In the MS-RCPSP problem, the population is calculated and evolved over several generations to improve the makespan of the project. Representing each individual in the population is a vector whose elements correspond to the total number of project tasks. The value at each vector element denotes the resource index assigned to perform the corresponding task.

1. Migration method

The traditional memetic algorithm operates by evolving a population over several generations through several steps, including selection, crossover, mutation, local search, and recombination to create a new population. However, while evolving, the algorithms may fall into the local extreme and iterate infinitely without leaving, so finding a better result is impossible. Therefore, the Migration technique moves the population to another solution space far from the local extreme. Integrating the Migration technique into the memetic algorithm leads to improved results.
The Migration method runs over steps as follows:

Step 1: Catching the local extreme

In this step, the algorithm uses a marked variable c_{fail} to count the times the makespan is not changed over generations; if c_{fail} was more significant than a specified threshold (c_{max}), then the population has fallen into the local extreme. Equation (10) represents the calculation of c_{fail}. (10)

Step 2: Changing the population

Once the population falls into a local extreme, the MIMEM algorithm tries to move the entire population to a new area by using the Migration method as follows:

- Repeating with each particle
- Considering each task i of the particle, get out the L_i of resources to execute the current task, reordering the resources index in L_i.

Table I
THE NOTATIONS

Symbol	Description
C_i	The set of tasks need to be completed before task i can be executed
S	The set of all resource's skills S_i: the subset of skills owned by the resource i, $S^i \subseteq S$;
S_i	The skill i;
t_j	The duration of task j
L	The resources used to execute tasks of the project
L^k	The subset of the resources which can be performed task k; $L^k \subseteq L$
L_i	The resource i
W	The tasks of the project need to do
W^k	The subset of task which can be executed by the resource k , $W^k \subseteq W$
W_i	The task i
r^i	The subset of the skill required by task i. A resource has the same skill and skill level equal to or greater than the requirement that can be performed.
B_k, E_k	The begin time and end time of the task k
Au, vt	The variable to identify the resource v is running task u at time t; *1*: yes, *0*: no;
hi	The skill level i;
gi	Type of skill i;
m	Makespan of the schedule
P	The feasible solution
P_{all}	The set of all solution
$f(P)$:	The function to calculate the makespan of P solution
n	Task number
z	Resource number

- Replacing the current resource with the resource which mirrors in L_i. The formula (11) shows this action.

Figure 1 presents the Migration technical by replacing the execution resource. Primarily, the task being performed by resource L_2 will be changed to L_{m-1}, and resource L_m does the task will be allocated to L_1.

After applying the Migration method, the population will be moved to a new location, allowing the algorithm MIMEM to escape the local extreme, continue the evolution-ary computation, and expand the solution space.

The Migration method is present in Algorithm 1 below.

2. MIMEM Algorithm

The MIMEM algorithm is a hybrid algorithm from the memetic algorithm and the Migration method. The improvement of MIMEM is demonstrated in "Migration steps", where the algorithm checks for local extrema within the population and redirects it to another region if it has fallen into one. This step is executed by applying the Migration method.

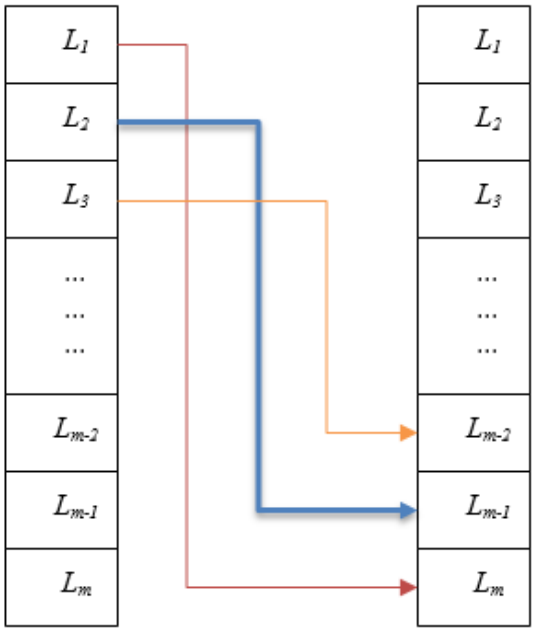

Figure 1. Replace the resource that performs the task

Algorithm 1: Migration

```
1  Input: dataset, g_max: number of generation, mutation-
     Probability, crossoverProbability, localSearchProbability
2  Output: best and duration
3  begin
4      P_new = {};
5      i = 1;
6      p_size = size(P);
7      while i < p_size do
8          P_i ← P_all[i];
9          j_task = 1;
10         n_task = size(P_i);
11         for j_task = 1 to n_task do
12             L^i ← the resources matching with L(j_task)
                 requirement;
13             L^i ← Reorder(L^i) ;
14             i_current ← current resource index to run
                 task i;
15             i_current = size(L^i) – idx + 1;
16             L_jtask = L^i[i_current];
17         end
18         P_new = P_new + {P_i};
19     end
20     return P_new;
21 end
```

Algorithm 2: MIMEM

```
1  Input: P_all – the population needs to move
2  Output: P_new – the result population
3  begin
4      Read dataset;
5      g = 0, c_fail = 0, c_max = 50;
6      P_all = {population initallation} ;
7      {makespan, best, fitness} = {Caculating the fitness of
         population and the best individual} ;
8      while g < g_max do
9          parents = Selection(population);
10         offspring = Crossover(population, crossoverProb-
             ability);
11         Mutate(offspring, mutationProbability);
12         LocalSearch(offspring, localSearchProbability);
13         P_all = CombinePopulation(P_all, offspring);
14         {new_makespan, new_best, fitness} = {Cacu-
             lating the fitness of population and the best
             individual from P_all} ;
15         if new_makespan # makespan then
16             c_fail = 0 ;
17         else
18             c_fail+ = 1;
19         end
20         if c_fail = c_max then
21             P_all = Migration(P_all);
22             c_fail = 0;
23         end
24         if makespan < new_makespan then
25             makespan = new_makespan; ;
26             best = new_best ;
27         end
28         g = g+1 ;
29     end
30     return {best, makespan } ;
31 end
```

The MIMEM algorithm is present in Algorithm 2.

At each evolution generation of the population, the local search technique is performed by calling the LocalSearch function, which relies on the localSearchProbability parameter to calculate the number of neighbor individuals ($localSearchProbability * sizeof(P_{all})$) which they are used to be calculated to find the best one and return it. The best individual among the neighbors has been selected if a better makespan exists than the current.

In Algorithm 2, lines 15-19 aim to detect local extremes, while commands in lines 20-23 move the population by calling the Migration function. After moving the population, the algorithm continues on the new solution area.

V. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed MIMEM algorithm, we exper-imented on the iMOPSE [6, 7] dataset presented in Table II.

1. Experimental parameters

Experiments were conducted with the following parameters and data:

- Benchmark instances: iMOPSE dataset (iM01 to iM15), each instance is a project with full input parameters such as the number of tasks, the number of resources used to execute the project, the number of precedence constraints, and the number of resource skills.
- The initial number of particles (i.e., solution, individual, schedule): 50
- The number of generations: 30,000
- The number of times the experiment was carried out on each element of the dataset: 30
- Experiment tools: Matlab version 2014
- PC configuration: CPU Core i5 2.2 GHz, 6GB RAM, Windows 10

2. Performance Analysis

Experimental results with the iMOPSE dataset are presented in Table III. The results are aggregated, compared, and evaluated based on three factors:

- BEST - best makespan values - this is the shortest time to complete the project
- AVG - average execution time of 50 experiments with each instance dataset
- STD values - the standard deviation value between experiments on the same data instance

Table II
THE IMOPSE DATASET

Name	Tasks	Resources	Precedence Relations	Skills
iM01	100	5	22	15
iM02	100	5	46	15
iM03	100	5	48	9
iM04	100	5	64	15
iM05	100	5	64	9
iM06	100	10	26	15
iM07	100	10	47	9
iM08	100	10	48	15
iM09	100	10	64	9
iM10	100	10	65	15
iM11	100	20	22	15
iM12	100	20	46	15
iM13	100	20	47	9
iM14	100	20	65	15
iM15	100	20	65	9

The average execution time of the GA algorithm and the MIMEM algorithm are 2.7 hours and 2.3 hours, respectively. Note that normally the time it takes to find a schedule makes almost no sense because the quality of its solution is what matters.

Myszkowski has published GArunner tool [21] that allows other research groups to implement his GA algorithm, which is considered one of the most efficient algorithm. Thanks to that tool, we implemented the GA algorithm and the MIMEM on the iMOPSE dataset [6, 7]. The results are shown in Table III.

The experimental results show that the proposed algorithm MIMEM is more efficient than the GA algorithm on the following parameters:

- In term of BEST values, the MIMEM algorithm outperforms GA in 12 out of 15 projects, achieving improvements ranging from 0 to 22.33%. However, in 3 projects, MIMEM performs worse than GA. Figure 2 illustrate the BEST values comparisons detail.
- Regarding AVG values, MIMEM yields better results than GA in 11 out of 15 projects, achieving improvements ranging from 0 to 23.31%. However, in 4 projects, MIMEM performs worse than GA. Figure 3 detail the AVG values comparisons for 15 projects.
- Moreover, MIMEM exhibits greater stability than GA with a total standard deviation of 49.8, compared to GA's 68.8 on 15 datasets of iMOPSE. Figure 4 present the STD values comparisons for 15 projects.

Experimental results demonstrate that the MIMEM algorithm is less effective for projects with fewer tasks, but it achieves higher efficiency for larger projects. This suggests that MIMEM is well-suited for scheduling real-world projects consisting of a series of jobs with a significant number of tasks. For instance, MIMEM can be applied in various fields, such as equipment manufacturing and industrial sewing lines,...

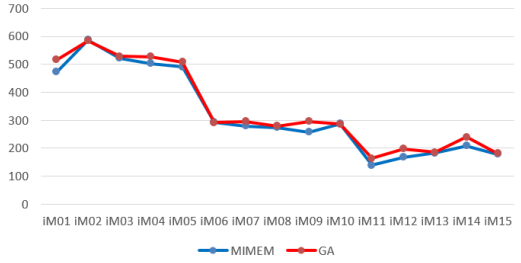

Figure 2. Comparison of the BEST parameter between MIMEM and GA

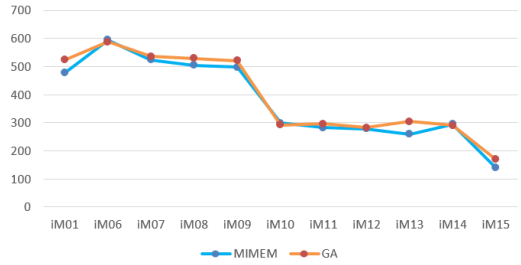

Figure 3. Comparison of the AVG parameter between MIMEM and GA

Table III
THE RESULTS FROM THE IMOPSE DATASET

Dataset	GA			MIMEM		
	Best	**Avg**	**Std**	**Best**	**Avg**	**Std**
iM01	517	524	5.3	473	477	2.5
iM02	584	587	4.7	587	594	4.3
iM03	528	535	9.7	522	523	10.4
iM04	527	530	2.5	503	504	1.6
iM05	508	521	9.9	490	497	4.7
iM06	292	292	1.7	294	299	0.1
iM07	296	296	2.3	280	283	1.7
iM08	279	282	2.9	274	278	3.4
iM09	296	305	6.6	258	259	0.7
iM10	286	290	5.0	287	294	3.5
iM11	163	169	5.8	139	139	1.9
iM12	197	207	6.9	168	169	8.2
iM13	185	186	0.5	183	188	0.4
iM14	240	240	0.5	208	211	5.3
iM15	181	187	4.5	179	185	1.1

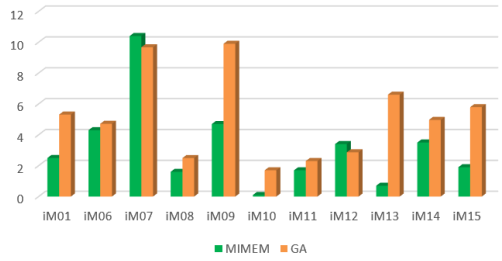

Figure 4. Comparison of the STD parameter between MIMEM and GA

VI. CONCLUSION

In this paper, we have presented a mathematical formulation of the MS-RCPSP problem and proposed a new algorithm, MIMEM, that combines the strengths of the Memetic algorithm and the Migration method. Our experiments on the IMOPSE dataset demonstrate that MIMEM outperforms previous algorithms regarding BEST and AVG values, achieving up to a 22.33% improvement on BEST values and a 23.31% improvement on AVG values. The total STD value suggests that the pro-posed algorithm exhibits more excellent stability, meaning that it produces less difference between experiment times.

The results suggest that MIMEM is a promising approach for solving complex scheduling problems with a large number of tasks, which is critical for various real-world industrial production scenarios such as industrial machine lines or some other manufacturing sectors. In future work, we plan to investigate other approximation methods, such as random moves based on Gauss, Cauchy, etc., to enhance further the quality of the schedules produced by MIMEM.

REFERENCES

[1] Zhang, Xin, et al. "A coevolutionary algorithm based on the auxiliary population for con-strained large-scale multi-objective supply chain network.", *Mathematical Biosciences and En-gineering*, 19.1 (2022): 271-286.

[2] Zhang, Wenqiang, et al. "Multi-stage hybrid evolutionary algorithm for multiobjective distrib-uted fuzzy flow-shop scheduling problem.", *Mathematical Biosciences and Engineering*, 20.3 (2023): 4838-4864.

[3] Wilson, Allan J., et al. "A review on memetic algorithms and its developments.", *Electrical and Automation Engineering*, 1.1 (2022): 7-12.4.

[4] Afsar, Sezin, et al. "Multi-objective enhanced memetic algorithm for green job shop schedul-ing with uncertain times.", *Swarm and Evolutionary Computation*, 68 (2022): 101016.

[5] Seo, Wangduk, et al. "Effective memetic algorithm for multilabel feature selection using hy-bridization-based communication.", *Expert Systems with Applications*, 201 (2022): 117064.

[6] Myszkowski, Paweł B., and Maciej Laszczyk. "Investigation of benchmark dataset for many-objective Multi-Skill Resource Constrained Project Scheduling Problem.", *Applied Soft Computing*, 127 (2022): 109253.

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

[8] Saad, Hatem MH, Ripon K. Chakrabortty, Saber Elsayed, and Michael J. Ryan. "Quantum-Inspired Genetic Algorithm for Resource-Constrained Project-Scheduling.", *IEEE Access*, 9 (2021): 38488-38502.

[9] Asadujjaman, Md, Humyun Fuad Rahman, Ripon K. Chakrabortty, and Michael J. Ryan. "An Immune Genetic

Algorithm for Solving NPV-Based Resource Constrained Project Schedul-ing Problem.", *IEEE Access*, 9 (2021): 26177-26195.
[10] Hussain, Adedoyin A., and Fadi Al-Turjman. "Hybrid Genetic Algorithm for IOMT-Cloud Task Scheduling.", *Wireless Communications and Mobile Computing*, 2022 (2022).
[11] Aurangzeb, Khursheed, Sheraz Aslam, Musaed Alhussein, Rizwan Ali Naqvi, Muhammad Arsalan, and Syed Irtaza Haider. "Contrast Enhancement of Fundus Images by Employing Modified PSO for Improving the Performance of Deep Learning Models.", *IEEE Access*, 9 (2021): 47930-47945.
[12] Rani, Rama, and Ritu Garg. "Energy efficient task scheduling using adaptive PSO for cloud computing.", *International Journal of Reasoning-based Intelligent Systems*, 13.2 (2021): 50-58.
[13] Sahana, Sudip Kumar. "Ba-PSO: A Balanced PSO to solve multi-objective grid scheduling problem.", *Applied Intelligence*, (2021): 1-13.
[14] Al-Yaseen, Wathiq Laftah, Ali Kadhum Idrees, and Faezah Hamad Almasoudy. "Wrapper feature selection method based differential evolution and extreme learning machine for intru-sion detection system.", *Pattern Recognition*, 132 (2022): 108912.
[15] Deng, Wu, et al. "An improved differential evolution algorithm and its application in optimiza-tion problem.", *Soft Computing*, 25.7 (2021): 5277-5298.
[16] Zhao, Suyao, et al. "Large-Scale Scheduling Model Based on Improved Ant Colony Algo-rithm." *Mobile Information Systems* 2022 (2022).
[17] Garg, Neha, et al. "Energy-Efficient Scientific Workflow Scheduling Algorithm in Cloud En-vironment.", *Wireless Communications and Mobile Computing* 2022 (2022).
[18] Shruthi, G., et al. "Mayfly taylor optimisation-based scheduling algorithm with deep rein-forcement learning for dynamic scheduling in fog-cloud computing.", *Applied Computational Intelligence and Soft Computing* 2022 (2022).
[19] Shams Lahroudi, Seyed Hassan, Farzaneh Mahalleh, and Seyedsaeid Mirkamali. "Multiobjec-tive Parallel Algorithms for Solving Biobjective Open Shop Scheduling Problem.", *Complexity* 2022 (2022).
[20] R. Kolisch, A. Sprecher, "PSPLIB-a project scheduling problem library: OR software-ORSEP operations research software exchange program.", *European journal of operational research*, 96(1), pp.205-216, 1997.
[21] GArunner tool: http://imopse.ii.pwr.wroc.pl/rcpsp_spsp _library.html

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 Software Engineering, Evolution Algorithm, Optimization Algorithm.
E-mail: 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, Viet Nam from 1997 and is currently a professor, vice dean of the Department of Information Technology. His research interests include Computer Network and Computer.
E-mail: locnt@hnue.edu.vn.