C. Ratanavilisagul, "A Novel Modified Ant Colony Optimization for Solving Multi-Skill Resource-Constrained Project Scheduling Problem," in 2023 8th International Conference on Computational Intelligence and Applications (ICCIA), Haikou, China, 2023, pp. 64-68, doi: 10.1109/ICCIA59741.2023.00020.

A novel modified Ant Colony Optimization for solving Multi-skill Resource-constrained Project Scheduling Problem

Chiabwoot Ratanavilisagul
Department of Computer and Information Science of Applied Science
King Mongkut's University of Technology North Bangkok (KMUTNB)
Bangkok, Thailand
chaibwoot@gmail.com, chaibwoot.r@sci.kmutnb.ac.th

***Abstract*— multi-skills resource-constrained project scheduling problem (MS-RCPSP) is assigned to employees under the predecessor tasks constraint and the multi-skills constraint in order to create the feasible schedule with both the cheapest cost and the shortest completion time. MS-RCPSP has multi-goals and complexity so Ant Colony Optimization (ACO) is applied to solve MS-RCPSP. Many researchers can find either the cheapest total cost plans or the shortest completion time plans. However, the practical applications want them both together. Hence, this paper proposes modifying ACO solving MS-RCPSP and obtaining good results both the cheapest cost and the shortest duration. The proposed algorithm was tested on thirty-six test cases from an MS-RCPSP iMOPSE datasets and provided more satisfactory results in comparison with other ACO algorithms.**

***Keywords*— Ant Colony Optimization, Project Scheduling Problem, Metaheuristics, Multi-objective Optimization**

I. INTRODUCTION

Resource-constrained project scheduling problem (RCPSP) is assigned all to employees to create the feasible schedule with the shortest completion time [1]. Mostly, the skills domain is considered in practice. In the business and the industry, project managers could encounter to manage effective project schedules within constraints such as duration, cost, skills of employee and other constraints [2]. Hence, algorithms for soling project scheduling are required by both the industry and the business. Moreover, the good algorithm could be good to assign into the project as it is less time-consuming than developing by hand.

Multi-skills Resource-constrained Project Scheduling Problem (MS-RCPSP) is RCPSP that is added into the multi-skill constraint. Both RCPSP and MS-RCPSP are also known as combinatorial optimization problems and are NP-hard problems [1-3]. These problems cannot be solved by brute force algorithms in case of having a lot of tasks within acceptable time. Heuristic algorithms and Meta-heuristic algorithms are widely used to solve them.

In recent years, many researchers have proposed the usage of heuristic algorithms to solve MS-RCPSP or any problem similar to this case. Heuristic and Meta-heuristic algorithms such as Differential Evolution (DE) [4, 5], Genetic Algorithm (GA) [6], and Ant Colony Optimization (ACO) [2, 7] can solve highly complex problem effectively. ACO has successfully solved MS-RCPSP. Hence, this paper focuses only on the ACO that solves MS-RCPSP.

ACO is motivated by the foraging behavior of an ant colony [8, 9]. This algorithm has been widely used for solving optimization problems [10] such as resource-constrained project scheduling problem (RCPSP), traveling salesman problems (TSP), network routing and vehicle routing problem (VRP) etc. Moreover, advantages of ACO [10, 11] are rapid discovery of good solutions and good efficiency for RCPSP or similar problems.

In paper of [7] proposed ACO applied to solve RCPSP. This algorithm is the standard ACO for applying to solve RCPSP. The experimental results show that ACO obtains better results than genetic algorithms and simulated annealing algorithms.

In paper of [2] proposed applying the heuristic based on priority rules with ACO or a hybridization of ACO approach (HACO) to solve MS-RCPSP. The experimental results show that HACO obtains better results than the standard ACO in term of either the cheapest total cost or the shortest completion time.

In practical applications, selecting the cheapest total cost may take the ultimate completion time until the results may be not able to apply to real practice. On the other hand, selecting the shortest completion time may have the ultimate total cost until the results may be not able to apply to real practice. The algorithm should allocate resources that can obtain both the cheapest total cost and the shortest completion time. These values should be balanced and optimum. The difficulty of MS-RCPSP is assigned to employees in order to obtain the optimum value by the balance of the total cost and the completion time.

The creation answers of ACO gets influence from pheromone of ants. [12-14]. As mentioned earlier, this paper proposed modifying ACO (MACO) to solve MS-RCPSP to apply in practical applications suitably. The main concept of the proposed algorithm is the pheromone of ants are updated by the standard ACO, the heuristic is guided by the shortest completion time, and the heuristic is guided by the cheapest total cost. The pheromone of ants are updated by heuristic and ACO. It can create better answers by both good the total cost and good the completion time.

MS-RCPSP iMOPSE datasets are benchmark problems [3] and were used to compare the ACO, HACO [2], and the proposed algorithm. The results show that the solution quality of the proposed algorithm is better than others comparing to ACO algorithms in the MS-RCPSP iMOPSE datasets.

The rest of this paper is organized as follows. Section 2 explains the related works (RCPSP, MS-RCPSP, ACO apply with MS-RCPSP, and HACO). Section 3 explains the proposed technique. Section 4 explains the experiment setup and presents the experiment results. Section 5 and concludes the project with a brief summary and directions for development algorithm in future work.

II. Related Work

A. *Resource-constrained Project Scheduling Problem*

RCPSP composes a set of tasks and a set of employees. Each employee is represented as integer numbers. Each task is represented as integer numbers. Each task composes starting time and finishing time. All tasks cannot be stopped if they have been started. Some tasks may have the previous task or the predecessor task. It means the previous task of this task must have been finished before this task started. The goal of RCPSP is assigned all tasks to employees under the precedence tasks constraint in order to create the feasible schedule with the shortest completion time.

B. *Multi-skill Resource-constrained Project Scheduling Problem*

MS-RCPSP is RCPSP that is added the multi-skills constraint of tasks and employees. Each employee has his skills and each his skill has a level. Each task requires a skill and its skill requires a level in order to perform this task. If an employee has a skill matching the skill of the task and the level of the employee is equal or more than the level of the task, this employee can perform this task. On the other hand, if an employee has a skill not matching the skill of the task or the level of the employee is less than the level of the task, this employee cannot perform this task.

Moreover, MS-RCPSP has multi-goals and complexity. The first goal is assigning all tasks to employees under the predecessor tasks constraint and the multi-skill constraint in order to create the feasible schedule with the shortest completion time. It is called the duration optimization (DO). The second goal is assigning all tasks to employees under the predecessor tasks constraint and the multi-skill constraint in order to create the feasible schedule with the cheapest total cost. It is called the cost optimization (CO). The third goal is assigning all tasks to employees under the predecessor tasks constraint and the multi-skill constraint in order to create the feasible schedule with both the shortest completion time and the cheapest total cost. It is called both duration and cost optimization (BO).

$$\min f(PS) = 0.5 \times \frac{T}{T_{max}} + 0.5 \times \frac{\sum_{i=1}^{J} c_j}{c_{max} - c_{min}} \tag{1}$$

$$\forall_{k\in K}(s_k) \geq 0, \forall_{k\in K}(Q^k) \neq \emptyset \tag{2}$$

$$\forall_{j\in J}(F_j) \geq 0, \forall_{j\in J}(d_j) \geq 0 \tag{3}$$

$$\forall_{j\in J, j\neq 1, I\in P_j}(F_i) \leq (F_j - d_j) \tag{4}$$

$$(\forall_{i\in J^k}\exists_{q\in Q^k}(h_q) = h_{q_i}) \wedge (l_q \geq l_{q_i}) \tag{5}$$

$$\forall_{k\in K}\forall_{t\in T}\sum_{i=1}^{n} U_{i,k}^{t} \leq 1 \tag{6}$$

$$\forall_{j\in J}\exists_{!t\in T, !k\in K}(U_{j,k}^{t}) = 1, U_{j,k}^{t} \in \{0,1\} \tag{7}$$

PS is Project Schedule. PS composes a set of tasks (J = 1,..., n) and a set of employees (K = 1,…, m). n is a number of tasks. m is a number of employees. T is the completion duration of the answer. T_{max} is the sum of the duration from all tasks. c_{min} is the total cost of all tasks assigned by the wage of an employee is the cheapest without skill constraint. c_{max} is the total cost of all tasks assigned by the wage of an employee is the most expensive without skill constraint. The task j has the hourly duration time of task j (d_j), the start time of task j (S_j) and the finish time of task j (F_j). P_j is the predecessors of task j. s_k is the hourly rate salary of employee k. Q is all skills. Q^k is skills of employee k (Q^k = 1,…., r). Q^k is a subset Q. r is a number of skills of employee k. q_j is a skill required by tasks j. l_q is the level of skill q. h_q is skill q. J^k is a subset of all tasks that employee k can be performed. c_j^k is the cost of performing task j by employee k ($c_j^k = d_i \times s_k$). If $U_{j,k}^t = 1$, employees k is assigned to tasks j in time t. On the other hand, If $U_{j,k}^t = 0$, employees k is not assigned to tasks j in time t. Equation (1) is the objective function or the fitness function of MS-RCPSP. Equation (2) is all salary employees with positive values and every employee must perform at least one task. Equation (3) is every task has a positive finish time and duration. Equation (4) is the precedence constraint. Equation (5) is the skills constraint. Equation (6) is an employee cannot perform two tasks at the same time. Equation (7) is each task must be performed by an employee.

C. *Ant Colony Optimization apply with MS-RCPSP*

Ant Colony Optimization is a stochastic algorithm that is used for solving combinational optimization problem. ACO can be described as follows: Initially, each edge has an initial pheromone $\tau_{ij}(0)$ between employees and tasks. The next step is to select a task of ant. The first task of each ant is randomly selected, and then each ant selects the next task according to the probability function as follows:

$$P_{ij}^{k} = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha}[n_{ij}]^{\beta}}{\sum_{t\in allowed}[\tau_{ij}(t)]^{\alpha}[n_{ij}]^{\beta}}, j \in allowed \\ 0\,, otherwise \end{cases} \tag{8}$$

Where P_{ij}^{k} is the probability of ant k choosing to employee i to task j, $\tau_{ij}(t)$ is the amount of pheromones, which will be found in employees and tasks in iteration t, $n_{ij} = \frac{1}{d_{ij}}$ is the inverse of the distance, β is a parameter which determines the relative importance of pheromone versus distance (β > 0). The result from formula Equation (8) is selection of a path that is shorter and has a greater amount of pheromone. After the ant selection process is completed, the fitness of the ant is calculated by Equation (1). The fitness of each ant is used to update pheromones. $\Delta\tau_{ij}^{k}(t)$ is the amount of pheromones left by an ant on a route that it is calculated by Equation (9). Q is a consistent value. k represents the k^{th} ant in the colony. L_k is the fitness of ant k.

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L_k}, \text{if the } k^{th} \text{ant uses edge}(i,j)\text{in its tour} \\ 0, \text{otherwise} \end{cases} \tag{9}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{10}$$

$$\tau_{ij}(t) = (1-\rho)\,\tau_{ij}(t-1) + \Delta\tau_{ij}(t) \tag{11}$$

Where $\Delta\tau_{ij}(t)$ is total amount of pheromones that ants use the route between employee i and task j have left. m is the amount of ant population. ρ is the coefficient of evaporation and receives a value between [0-1]. The next step is to repeat this until the stop condition is reached.

In terms of MS-RCPSP, each ant is an answer. The answer is composed of all employees. The process of ant assigned all tasks to employees under the predecessor tasks constraint and the multi-skill constraint.

D. Hybridization of Ant Colony Optimization approach

To improve searching performance and obtain the better answers, the standard ACO was modified by using simple heuristics based on priority rules. This technique was called a hybridization of ACO approach (HACO). The process of this technique can be summarized as follows: this Algorithm has two modes. The first mode is the duration optimization mode (DM). The second mode is cost optimization mode (CM). For DM, initial ant population using heuristics with considering from the earliest time and a number of successors. Then the standard ACO is executed. For CM, initial ant population using heuristics with considering salary rate. Then the standard ACO is executed. Experimental results showed that HACO got a good answer in terms of either the duration optimization or the cost optimization.

III. The Proposed Work

Selecting the cheapest total cost only or selecting the shortest completion time only may be unsuitable for practical applications. For example, selecting the cheapest total cost may hire only one employee that has the cheapest salary while other employees may not hire. The result is that the time of this project is very much such as 2 years. If capital is increased, the used time of this project may rest for 3 months. On the other hand, selecting the shortest completion time may hire all employees. The capital is very high, such as 2 million. If completion time is increased, the capital may decrease rest to 5 hundred thousand. As mentioned earlier, a good algorithm should allocate resources to obtain both the cheapest total cost and the shortest completion time in order that it is suitable for practical applications. The difficulty of MS-RCPSP is assigning all tasks to employees in order to obtain the optimum values by the balance of the total cost and the completion time.

The creation answers of ACO get influenced by the pheromone of ants while the pheromone of ants is updated from answers [12-14]. The answers of ACO are created by Heuristic that is guided by the cheapest total cost. The pheromone of ACO which is updated from this answer tries to create answers that get a few total cost in the next iterations. The answers of ACO are created by Heuristic that is guided by the shortest completion time. The pheromone of ACO which is updated from this answer tries to create answers that get a little completion time in the next iterations. If the answers of ACO are created by Heuristic that are guided by both the cheapest total cost and the shortest completion time, it will create good answers that are both a few total cost and a little completion time.

Hence, this paper proposed the pheromones of ants are updated by the standard ACO, Heuristic is guided by the cheapest total cost, and Heuristic is guided by the shortest completion time. The pheromones are updated with a variety guide. It can create better solutions that get the optimum value by the balance of the total cost and the completion time. The process of the proposed algorithm is as follows: the population of ants is divided into three groups.

The first group is Heuristic that is guided by the cheapest total cost. The first step, employees are sorted ascending by their salary rate. Then, all predecessor tasks are assigned to employees in sorted order. It means the employee has a cheap salary. This employee is assigned to tasks before. Until this employee cannot be assigned to tasks. Then, all rest tasks are assigned to employees in sorted order. Assigning tasks must preserve the multi-skill constraint and the predecessor tasks constraint.

The second group is Heuristic that is guided by the shortest completion time. The first step, the predecessor tasks are sorted by a number of successors in descending order. Then, all predecessor tasks are assigned to employees in sorted order. It means a task has many successor tasks. This task was assigned before. Moreover, assigning each predecessor task must affect the completion time that is the least increased. Then, all rest tasks try assigning to all employees. Any task that affects the completion time that is the least increased. That task is assigned. The process is performed repeatedly until all rest tasks are assigned. Assigning tasks must preserve the multi-skill constraint and the predecessor tasks constraint.

The third group is searching by the standard ACO. The largest population of ants is the third group because the first group and the second group are used to adjust a little pheromone. Most searches are still using the standard ACO. The results from the first group and the second group, the pheromones are updated with a variety guide. It can create better answers. The proposed algorithm is called modified ACO for solving multi-skill resource-constrained project scheduling problems or MACO. Pseudo code of MACO is shown below:

```
Initialize edges, solution, and pheromone table
While termination condition ≠ true do
  For ant x begin 1 to a number of all ants
    If x > 0 and x < NC
      Sort all employees by salary rate as ascending order
      All predecessor tasks are assigned to employees as sorted order into list of ant x
      All rest tasks are assigned to employees as sorted order into list of ant x
    Else If x > NC and x < (NC + NT)
      Sort all predecessor tasks by a number of successors as descending order
      Each task is assigned; the completion time is the least increased.
      All predecessor tasks are assigned to employees as sorted order into list of ant x
      All rest tasks are assigned to employees into list of ant x
    Else
      For until all tasks are assigned into list of ant x
        Random an employee that uses to assign task
        The assigned task is not repeat tasks in list of ant x
        The assigned task passes condition of the skill constraint
        The assigned task passes condition of the predecessor task constraint
        Assign a task with probability according to formula (8)
      End For
    End If
    Evaluate the fitness of ant x according to formula (1)
    If the fitness of ant x is better than that of solution
      Solution = ant x
    End If
    Update pheromone according to formula (10) by ant x
  End For
  Apply the evaporation
End While
```

Where NC is the population of Heuristic is guided by the cheapest total cost. NT is the population of Heuristic guided by the shortest completion time.

IV. Experiments and Results

A. *The measures of algorithm performance*

Comparing algorithm performance uses the average value from runs because ACO is a stochastic algorithm. The measures of algorithm performance in the experiments are as follows: the average best total cost value (ABC) is the average of best total cost in the final iteration from all runs. The average best completion time value (ABT) is the average of best completion time in the final iteration from all runs. The average of all datasets (AD) is the average of ABC or ABT from all datasets. ABC, ABT, and AD indicate the answer searching efficiency of an algorithm. The more these values are the less value, the better they are. The experiment was tested on thirty-six datasets from iMOPSE dataset benchmark problems. This dataset is available for download at [15]. There are two groups of created project instances. The first group has 100 tasks. The second group has 200 tasks. It is a standard dataset and can be used to compare the performance of algorithms.

B. *Parameters Setting*

The parameters are as follows for all experiments: $\beta = 0$, $Q = 0.05$, $\alpha = 1$, $\rho = 0.1$. For these parameters are suggested by this paper [2]. For MACO, NC = 5 and NT =5. This value of NC and NT gets from the results from experiment. The experiments start defining NC and NT equal 0 then increases 5 until 100 and select values of NC and NT that can create the best result. The number of ants used is 100. The number of experiments of each datasets is 10 runs. The maximum number of iterations is set as 200.

C. *Experiment of Proposed Algorithm*

From the experimental results in Table 1, overall quality answers of MACO are better than that of ACO and HACO because of its lowest AD in terms of both the completion time and the total cost. It shows that the pheromones of ants are updated by using Heuristic that is guided by both the cheapest total cost and the shortest completion time. It can increase the search performance of ACO and get better answers.

For HACO with CM and HACO with DM, overall total costs of HACO with CM are better than that of HACO with DM because the pheromones of ants obtain influence from Heuristic that are guided in terms of the total cost. While, overall completion time of HACO with DM is better than that of HACO with CM because the pheromones of ants obtain influence from Heuristic that are guided in terms of time.

ABC or ABT of MACO in some datasets is more than that of ACO and HACO such as 100_10_47_9, 200_10_135_9_D6 and 200_20_54_15. In this case, ABT of MACO is more than ABT of HACO with DM but ABC of MACO is less than ABC of HACO with DM. It means that MACO may select the total cost more than the completion time. Surely, ABT or ABC of MACO is less than those of both ACO and HACO. Mostly, both ABT and ABC of MACO are less than those of ACO and HACO. Hence, MACO outperforms ACO and HACO.

TABLE I. COMPARATIVE RESULTS OF ACO, HACO AND MACO.

Name Datasets	ACO		HACO (CM)	
	ABC	ABT	ABC	ABT
100_10_26_15	80158.32	2472.00	110649.20	2401.30
100_10_27_9_D2	34942.15	2468.80	41461.36	2351.50
100_10_47_9	127541.30	3240.50	124464.20	3226.60
100_10_48_15	112234.04	2868.30	119309.30	2945.20
100_5_64_9	93149.56	2751.00	101108.00	2912.80
100_10_65_15	135570.10	3011.60	143515.40	2689.60
100_20_22_15	71495.38	2682.30	97492.30	2576.00
100_20_23_9_D1	40197.46	2424.50	46690.17	3297.10
100_5_46_15	193754.40	2842.20	197315.10	2750.80
100_20_47_9	93166.80	3093.00	100413.56	4007.10
100_20_65_15	89134.26	3146.00	93601.29	2981.10
100_20_65_9	109757.00	3435.30	96246.49	3879.40
100_5_20_9_D3	34859.14	1992.70	38784.27	2241.80
100_5_22_15	114421.60	2427.30	117180.80	2419.00
100_20_46_15	107452.69	3386.10	112279.00	3263.70
100_5_48_9	188375.40	3014.40	188933.10	2476.50
100_5_64_15	122885.30	2812.80	135678.00	2593.80
100_10_64_9	93228.02	3224.60	97265.53	3323.50
200_10_128_15	178108.40	7260.40	166123.50	5473.40
200_10_135_9_D6	98356.74	9138.10	98586.81	8812.60
200_10_50_15	107315.65	5828.40	164144.40	4860.10
200_10_50_9	159042.50	7063.60	221877.00	5199.60
200_10_84_9	181258.00	8338.30	191878.30	8262.00
200_10_85_15	269313.60	7057.10	278855.30	6088.70

200_20_145_15	255135.80	9188.20	226758.40	7459.40
200_20_150_9_D5	85977.79	14385.80	87511.00	22256.40
200_20_54_15	237462.10	6572.70	272421.50	5319.70
200_20_55_9	131765.16	6804.10	190965.10	7726.30
200_20_97_15	264327.90	8561.60	248525.00	6956.60
200_20_97_9	217644.30	8388.80	217506.40	6873.40
200_40_130_9_D4	104853.70	18092.00	103109.63	13590.40
200_40_133_15	223598.40	10485.10	203535.40	10059.20
200_40_45_15	203568.30	6518.80	216507.00	6673.50
200_40_45_9	186190.00	7646.80	232588.50	7859.00
200_40_90_9	319481.10	132996.40	215039.30	10465.50
200_40_91_15	258344.30	8757.20	213871.90	9220.40
AD	147890.74	9288.24	153116.43	5708.14

TABLE I. (CONTINUED).

Datasets	HACO (DM)		MACO	
	ABC	ABT	ABC	ABT
100_10_26_15	121483.00	2340.90	78540.01	1855.50
100_10_27_9_D2	42831.28	2312.80	34256.50	1701.90
100_10_47_9	142728.70	2518.20	92771.80	3410.00
100_10_48_15	135912.20	2428.70	108612.00	1884.00
100_5_64_9	100549.17	2380.00	97120.47	2122.00
100_10_65_15	151919.30	2446.50	130112.30	1704.80
100_20_22_15	109462.70	2518.60	65923.89	2025.00
100_20_23_9_D1	47777.53	3012.40	36057.49	1813.70
100_5_46_15	201369.40	2643.20	189742.50	1784.20
100_20_47_9	124852.40	2573.30	99032.42	2082.10
100_20_65_15	109503.00	2602.20	58134.70	2648.00
100_20_65_9	123004.30	2592.90	60954.00	3167.00
100_5_20_9_D3	39313.26	2089.10	31642.32	1547.00
100_5_22_15	118950.80	2420.10	111189.00	1991.00
100_20_46_15	138651.60	2736.40	103470.28	2028.10
100_5_48_9	193370.30	2450.50	182397.60	1918.10
100_5_64_15	138991.50	2431.60	120709.60	1605.50
100_10_64_9	114350.80	2444.60	79685.37	2870.70
200_10_128_15	182817.00	4725.00	180849.60	4612.00
200_10_135_9_D6	103887.80	5757.30	73207.80	7423.00
200_10_50_15	180209.10	4861.50	112870.50	4179.80
200_10_50_9	243683.90	4876.30	132471.40	4146.90
200_10_84_9	226091.80	5050.00	220967.70	5202.10
200_10_85_15	304689.60	4730.80	199585.00	5496.00
200_20_145_15	277246.20	4774.20	151585.50	5489.70
200_20_150_9_D5	86958.51	16327.80	52542.80	10384.00
200_20_54_15	291189.40	5125.10	164143.00	5384.00
200_20_55_9	230596.10	4885.90	131872.90	4371.50
200_20_97_15	291789.20	5278.90	175306.60	6513.20
200_20_97_9	276638.90	4808.90	100422.00	6182.00
200_40_130_9_D4	93092.90	13922.80	91192.76	11582.80
200_40_133_15	278846.60	5297.70	257799.10	4823.00
200_40_45_15	257434.70	4945.00	142692.90	4181.70
200_40_45_9	268131.80	5316.20	148008.70	4517.50
200_40_90_9	220086.80	8669.80	291468.10	5314.80
200_40_91_15	189275.60	8800.40	244208.10	5091.10
AD	171046.87	4530.43	126431.85	3973.71

V. CONCLUSION

MS-RCPSP has multi-goals that are the shortest completion time and the cheapest total cost. Using heuristics based on either the cheapest cost or the shortest completion time is unsuitable to solve MS-RCPSP. The algorithm is used for solving MS-RCPSP. It should create a feasible schedule with both the cheapest total cost and the shortest completion time. The creation answers of ACO gets influenced by the pheromone of ants. Hence, this paper proposed a novel modifying ACO can solve MS-RCPSP effectively. The proposal technique has the main idea as the pheromone of ants are updated by the standard ACO, Heuristic is guided by the shortest completion time, and Heuristic is guided by the cheapest total cost. The pheromones are updated from ACO and Heuristic. It can create better answers by both optimum total cost and optimum completion time. The proposed technique is called MACO. From the iMOPSE benchmark problem datasets, ACO, HACO with DM, HACO with CM and MACO were tested and results were compared. The results indicated that the proposed MACO outperforms other comparative algorithms in terms of the quality of the solutions in all experiments.

In the future, the local optimum problem of ACO will be studied to solve From the ACO experimental results, It can be observed that the answer has not improved in a long time. I think adding novel techniques such as the mutation operation technique and the rest technique into the process of ACO may solve or decrease local optimum problems and obtain better results.

ACKNOWLEDGMENT

This research was funded by Faculty of Applied Science, King Mongkut's University of Technology North Bangkok, Thailand contract no. 661127

REFERENCES

[1] Blazewicz J, Lenstra JK, and Rinnooy Kan AHG, “Scheduling subject to resource constraints: classification and complexity,” Discret Appl Math, vol 5, pp. 11–24, 1983.

[2] Pawel B. Myszkowski, Marek E. Skowronski, Lukasz P. Olech, and Krzysztof Oslizlo, “Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem,” Soft Comput 19, pp. 3599–3619, 2015.

[3] Kolisch R, and Sprecher A, “PSPLIB a project scheduling problem library,” Eur J Oper Res 96, pp. 205–216, 1996.

[4] Maciej Laszczyk, and Paweł B. Myszkowski, “Improved selection in evolutionary multi–objective optimization of multi–skill resource–constrained project scheduling problem,” Information Sciences, vol. 481, pp. 412-431, 2019.

[5] H. D. Quoc, L. N. The, C. N. Doan, and T. P. Thanh, “New Effective Differential Evolution Algorithm for the Project Scheduling Problem,” 2020 2nd International Conference on Computer Communication and the Internet (ICCCI), pp. 150-157, 2020.

[6] Jian Lin, Lei Zhu, Kaizhou Gao, “A genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem,” Expert Systems with Applications, vol. 140, 2020.

[7] Merkle D, Mittendorf M, Schmeck H, “Ant Colony Optimization for resource-constrained project scheduling,” IEEE Trans Evol Comput 6, vol. 4, pp. 333–346, 2002.

[8] A. Colorni, M. Dorigo and V. Maniezzo, “distributed optimization by ant colonies,” proceedings of european conference on artificial life, Paris, France, pp. 134-142, 1991.

[9] M. Dorigo and L. M. Gambardella, “ant colony system: a cooperative learning approach to the traveling salesman problem,” IEEE transactions on evolutionary computation, vol. 1, pp. 53 - 66, 1997.

[10] V. Selvi, and R. Umarani, “comparative analysis of ant colony and particle swarm optimization techniques,” International journal of computer applications, vol. 5, pp.1–6, 2010.

[11] M. Dorigo and L. M. Gambardella, “a study of some properties of ant-Q,” proceedings of the 44th international conference on parallel problem solving from nature, Springer, Berlin, pp. 656–665, 1996.

[12] C. Ratanavilisagul, “Modified Ant Colony Optimization for Vehicle Routing Problem with Time Windows using Limited Search Space and Novel Updating Pheromone and Re-initialization Pheromone Techniques,” ICIC Express Letters, Part B: Applications (ICIC-ELB), Vol. 1, pp. 571-580, 2022.

[13] C. Ratanavilisagul, and B. Pasaya, “Modified Ant Colony Optimization with Updating Pheromone by Leader and Re-Initialization Pheromone for Travelling Salesman Problem,” 2018 International Conference on

Engineering, Applied Sciences, and Technology (ICEAST), pp. 1-4, 2018.

[14] C. Ratanavilisagul, "A Novel Modified Ant Colony Optimization Algorithm by Resetting and Updating Pheromone for Vehicle Routing Problem with Time Windows," 2021 7th International Conference on Computer and Communications (ICCC), pp. 1194-1198, 2021.

[15] http://imopse.ii.pwr.edu.pl/download.html [accessed December 11, 2022].

AUTHORS' BACKGROUND

Your Name	Title	Research Field	Personal website
Chiabwoot Ratanavilisagul	assistant professor Dr.	Artificial Intelligence, Natural language processing, Machine learning, Searching Technique, Optimization	http://chaibwoot.no-ip.org/Chiabwoot.jsp