Vietnam Journal of Science and Technology ***62*** *(1) (2024) 140-155*

doi:10.15625/2525-2518/17919

AdaL-PSO - a new adaptive algorithm for the Multi-Skilled resource-constrained project Scheduling problem

Phan Thanh Toan, Do Van Tuan

Samsung Display Vietnam, Yen Phong Industrial Zone, Bac Ninh, Viet Nam

*Email: *pttoan@hnue.edu.vn*

Received: 14 June 2023; Accepted for publication: 4 April 2023

Abstract. MS-RCPSP is a combinatorial optimization problem with many practical applications, this problem has been proven to belong to the NP-hard class, the approach to solving this problem is to use algorithms to find an approximate solution. In this paper, we adopted a new adaptive nonlinear weight update strategy based on fitness value and new neighborhood topology for Particle Swarm Optimization algorithm, thereby helping to prevent PSO from falling into local extremes. The new algorithm is called AdaL-PSO. A numerical analysis is carried out using iMOPSE benchmark dataset and is compared with some other early algorithms. The results presented suggest the prospect of our proposed algorithm.

Keywords: Swarm intelligence, evolutionary computing, pptimization methods, resource constrained Project Scheduling, particle Swarm optimization.

Classification numbers: 4.7.1, 4.7.4

1. INTRODUCTION

The resource-constrained projects cheduling problem (RCPSP) has been an important optimization problem over the past few decades with many applications in manufacturing, production planning, resource scheduling in cloud computing, and some other areas [1 - 3]. Over the past few decades, there has been increasing interest in the study of the MS-RCPSP problem, which is an extension of the RCPSP problem. MS-RCPSP is a combinatorial optimization problem that was inspired by a scheduling problem in software development industry, where staff had several different skills, such as programming, data analysis, debugging and so on. In the MS-RCPSP problem, employees have a number of skills and the skill levels vary significantly. Each activity requires a certain skill level, which increases the flexibility of the scheduling problem, but it also makes the problem more complex.

The Resource Constrained Project Scheduling Problem has been proven to belong to NP-hard class, which is one of the most investigated topics in the class of scheduling problems [4].

The paper includes six major sections: some methods for solving the MS-RCPSP problem are described in section 2, where an overview of the particle swarm optimization algorithm is also presented. Particle Swarm Optimization is a metaheuristic algorithm which is based on the

behavior of social animals, this algorithm has been applied to efficiently solve optimization problems. The mathematical formulation of the MS-RCPSP problem and system model will be presented in section 3 of this paper. The proposed algorithm is presented in section 4, which is based on the Particle Swarm Optimization strategy [4]. We also focus on presenting a new neighborhood topology called LNDS (local neighborhood descent search) and inertia weight update strategy based on sigmoid function, these are the two main components of the proposed algorithm. The proposed algorithm is conducted on the iMOPSE benchmark dataset [5], the simulation results are analyzed in detail in section 5 of this paper. We also conducted the algorithm on the factory's data set collected from TNG, this is the data from a large textile company in Viet Nam. Conclusions and directions for future research are presented in section 6 of the paper.

2. RELATED WORK

2.1. Approximation methods for MS-RCPSP problem

Scheduling problem is an issue of great concern to many companies and is applied in many fields [6 - 11]. The MS-RCPSP problem, which we mentioned in this paper, is also proven to belong to the NP-hard class [4, 12, 13]. In addition, MS-RCPSP is multimodal or discontinuous and very challenging to solve with traditional optimization stratery. In the past few years, many research works and different approaches have been proposed to solve the MS-RCPSP problem [14 - 17]. The approach based on metaheuristic algorithms is the most common, the authors in papers [18 - 22] used the Genetic Algorithm (GA), Ant Colony Optimization (ACO) and Particle Swarm Optimization.

Z. Chen and C. Chyu [23] proposed a scheduling algorithm for Resource-Constrained Project Scheduling Problem based on Evolutionary Algorithm, in the paper authors combine the Evolutionary Algorithm and Local Search method to minimize the execution time of schedule. P. Das [24] proposed a scheduling algorithm for RCPSP based on the Simulated Annealing algorithm.

A hybrid algorithm was created by Myszkowski *et al*. [5, 13], this is a combination of two methods: Differential Evolution method and the Greedy algorithm. The goal of the authors is to minimize the makespan and costs, and the authors also build a new benchmark dataset for the MS-RCPSP problem called iMopse [5], this dataset is created based on real-world instances of the problem.

Mejia *et al*. [3] proposed a new variant of the MS-RCPSP problem to optimize the schedule of research activities in a nuclear laboratory. In the article [25], the authors proposed two new evolutionary algorithms for the MS-RCPSP problem with the deterioration effect and financial constraints. Hosseinian [26] proposed an extended model of the MS-RCPSP problem, this model has a dual objective, the author’s purpose is to simultaneously minimize the makespan and total cost of the project.

Resource-Constrained Project Scheduling Problem and its extention, Multi-Skill Resource-Constrained Project Scheduling Problem, have become important research topics over the last few decades. Many authors have proposed different methods for this problem [27 - 30].

2.2 Mathematical formulation of PSO

PSO (Particle Swarm Optimization) is one of the most well-known population-based stochastic optimization algorithms which was proposed by Kennedy and Eberhart [32]. Velocity and position are updated as follows:

$$v_i^{t+1} = w^t \times v_i^t + C_1 \times rand_1 \times \left(p_{best_i}^t - x_i^t\right) + C_2 \times rand_2 \times (p_{gbest}^t - x_i^t) \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^t \quad (2)$$

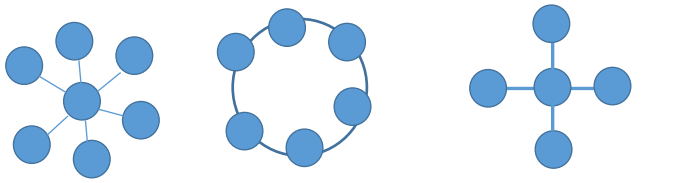
2.3. Topological neighborhood for PSO

Particles have been studied in two general types of neighborhoods: 1) global best and 2) local best. There are many topological neighborhoods [33]. The formula for updating the position vector is

$$v_i^{t+1} = w^t \times v_i^t + C_1 \times rand_1 \times \left(p_{best_i}^t - x_i^t\right) + C_2 \times rand_2 \times (p_{lbest_i}^t - x_i^t) \quad (3)$$

where p_{lbest_i} is the local best position of particle i with the best fitness value among its neighbors.

As shown in Figure 1, three neighborhood topologies are widely used.



Star topology Ring topology Von neumann topology

Figure 1. Neighborhood topologies.

3. MATHEMATICAL MODEL AND DEFINITIONS

3.1. Classical RCPSP

MS-RCPSP is an extension of the RCPSP problem, where each resource has multiple skills and each activity requires one or more specific skills to be executed. The description of the classical RCPSP would be introduced as follows:

(i) We represent each project by a directed non-cyclical graph G(V, E), where each node on the graph represents a task and the arcs represent the ordinal relationship between the tasks. The arc (u, v) ∈ E shows that task u must be completed before task v begins (Figure 2).

(ii) We add two empty tasks to the project. A task is placed at the beginning of the project, which is the predecessor task of all other tasks in the project and the second empty task is placed at the end of the project, which is the successor of all other tasks.

(iii) The duration of a specific task is the length of the task in execution time, from its start time to its finish time.

(iv) Tasks must not be stopped during execution, until it is done.

(v) Each task requires some resources to execute.

3.2. Mathematical model of the MS-RCPSP

The MS-RCPSP is an important scheduling problem that is applied to many real-world problems [5, 13, 31]. In MS-RCPSP each task requires some skills at a specified level to be excuted, whereas resources have multiple skills at different levels.

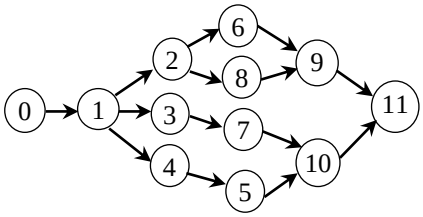


Figure 2. The relationship between tasks in a workflow.

Some notations of the MS-RPSP are given as follows:

Pre_i	The set of tasks that need to be completed before task i can be executed
SK	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;
d_j	The duration of task j
t_{jk}	The time to complete task j by resource k
Re	The resources used to execute tasks of the project
Re^k	The subset of the resources which can perform task k; $Re^k \subseteq Re$
Re_i	The resource i
W	The project tasks to be done
W^k	The subset of the task which can be executed by resource k , $W^k \subseteq W$
W_i	The task i
r^j	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 task k
$A_{u,v}^{\ t}$	The variable to identify resource v running task u at time t (1: yes, 0: no)
h_i	The skill level i
h_{S_q}	Level of skill q of a resource
g_i	Skill type i
g_{S_q}	Type of skill q of a resource
Mk	Makespan of the schedule
P	The feasible solution
P_{all}	The set of all solutions
$f(P)$:	The function to calculate the makespan of P solution
N	Number of tasks to execute
M	Number of tasks available

The Real-RCPSP problem could be stated as follows:

$$f(P) \rightarrow min \tag{4}$$

where:

$$f(P) = max_{W_i \in W}\{E_i\} - min_{W_k \in W}\{B_k\} \tag{5}$$

subjected to the following constraints:

$$S^k \neq \varnothing \qquad \forall Re_k \in Re \tag{6}$$

$$t_{jk} \geq 0 \qquad \forall W_j \in W, \ \forall Re_k \in R \tag{7}$$

$$E_j \geq 0 \qquad \forall W_j \in W \tag{8}$$

$$E_i \leq E_j - t_j \qquad \forall W_j \in W, j \neq 1, W_i \in C_j \tag{9}$$

$$\forall W_i \in W^k \ \exists S_q \in S^k : \ g_{S_q} = g_{r_i} \text{ and } h_{S_q} \geq h_{r_i} \tag{10}$$

$$\forall L_k \in L, \forall q \in m : \sum_{i=1}^{n} A_{i,k}^q \leq 1 \tag{11}$$

$$\forall W_j \in W \ \exists! \, q \in [0, m], ! \, L_k \in L : A_{j,k}^q = 1; \text{where} A_{j,k}^q \in \{0; 1\} \tag{12}$$

$$t_{ik} \geq t_{il} where h_k \leq h_l \forall \, (r^k, r^l) \in \{S^k \times S^l\} \tag{13}$$

Example 1

In Figure 3, $S_{i.j}$ denotes S_i skill type and it has j skill level. The resources of the project have skill types: S_1 and S_2, and each of them can have a skill level from 1 to 3. The resource constraints for the tasks are presented as follows:

Resources	
$S_{1.3}$, $S_{2.2}$	Re_1
$S_{2.1}$, $S_{3.2}$	Re_2
$S_{1.2}$, $S_{2.1}$	Re_3
$S_{2.2}$, $S_{3.3}$	Re_4

Tasks			
$S_{2.2}$	$S_{3.1}$	$S_{2.2}$	$S_{1.1}$
J_1	J_2	J_3	J_4
✓	♦	✓	✓
♦	✓	♦	♦
♦	♦	♦	✓
✓	✓	✓	♦

✓ Can be assigned
♦Can not be assigned

Figure 3. The relation matrix of resources-tasks assignment.

4. PROPOSED ALGORITHM

4.1. Schedule Representation

The MS-RCPSP problem consists of two important components: resource allocation and task sequencing. In this paper we use the Resource-only scheme method to represent the solution of the problem. Each solution is represented as an vector $S = \{R_{\pi 1}, R_{\pi 2}, .. R_{\pi N}\}$; where N is the number of task .

Example 1

Consider a project with tasks as: J={J_1, J_2, J_3, J_4, J_5, J_6, J_7, J_8, J_9, J_{10}} and project resources RS = {Re_1, Re_2, Re_3}, as show in Figure 2.

Our goal is to find a solution with a minimum project completion time (makespan) and satisfy two constraints: the order of execution of tasks and the skill level of the resources.

The duration of the tasks are given in Table 1

Table 1. Task duration.

Task	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}
Duration	3	5	2	4	1	2	4	4	3	4

Assume that the resources are assigned to the tasks as shown in the following table:

Table 2. Resource – task assignment.

Task	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}
Resource	Re_1	Re_1	Re_2	Re_3	Re_3	Re_1	Re_2	Re_1	Re_1	Re_3

A feasible schedule for the project is demonstrated in Figure 2, where it can be seen that the makespan is 17.

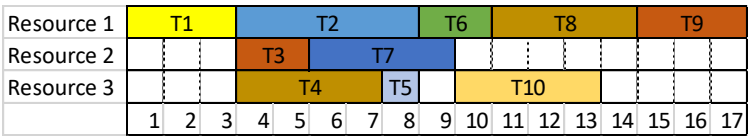


Figure 4. A feasible schedule.

The above schedule is demonstrated in Table 2, where resource 1 performs tasks J_1, J_2, J_6, J_8, and J_9; resource 2 is assigned to execute tasks J_3 and J_7; resource 3 deals with the rest tasks. According to this scheme, the project completion time (makespan) is 17.

4.2. Proposed algorithm AdaL-PSO

In the PSO algorithm, two parameters C_1 and C_2 provide local/global exploration, while the inertia weight parameter determines the relative rate of convergence. Many studies have shown that the algorithm convergence speed is significantly reduced when using a fixed inertia value [34]. This is a consequence of excessive momentum in the population, so the large step size exceeds the best search regions. In this paper, we proposed an adaptive inertia weight update strategy based on the fitness value. When the fitness value tends to increase, a large inertia weight is needed in order for the design space to be searched thoroughly. As fitness value decreases, which means that the most promising area of the search space has been discovered and the speed of convergence tends to slow down, the inertia weight should be reduced, in order for the momentum of the particles to decrease, allowing them to concentrate in the best search areas.

To accomplish the above strategy, we proposed a fitness-dependent value of the inertia weight. A commonly used inertia update rule is the non-linear decreasing based on sigmoid function, calculated by the formula as below:

$$\theta = -10 + \frac{(f^t - f_{min}) \times 20}{f_{max} - f_{min}} \tag{14}$$

where: f^t is the fitness value at the t^{th} iteration; f_{min}: the minimum value of the fitness function up to time t; f_{max}: the maximum value of the fitness function up to time t.

The inertia weight parameter was updated by the following formula:

$$w^t = sigmoid(\theta) = \frac{1}{1+e^{-\theta}} \quad (15)$$

Sigmoid functions have the domain of all real numbers, with a return value between 0 and 1, it increases monotonically and continuously everywhere. For values less than -10, the function's value is almost equal to zero. For values greater than 10, the function's values are almost equal to one.

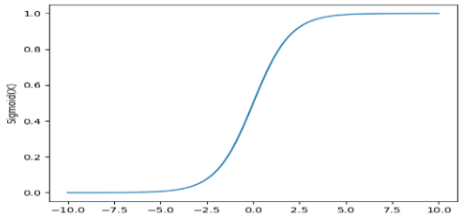


Figure 5. The sigmoid function

From (1), (2), (3) and (13), (14) we have the formula to update the velocity as follows:

$$v_i^{t+1} = \frac{1}{1+e^{\theta}} \times v_i^t + C_1 \times rand_1 \times \left(p_{best_i}^t - x_i^t\right) + C_2 \times rand_2 \times \left(p_{lbest_i}^t - x_i^t\right) \quad (16)$$

$p_{lbest_i}^t$ is calculated based on the best experience of the i^{th} particle and its "social" neighbors. The procedure for determining $p_{lbest_i}^t$may include a component that takes into account the current position X_i. In this paper, we propose a new algorithm to calculate $p_{lbest_i}^t$, that is implemented by a local neighborhood descent search method (LNDS). The goal of this method is to find a new vector position X_i', situated in the neighborhood of the vector X_i and having a better fitness value.

LNDS starts from current position X_i and makes changes in m iterations (m is the number of tasks). At each iteration, only one position #d, d = 1,…,m is changed to reduce the fitness function. Hence, a sequence of adjacent position vectors is generated as in the following scheme:

$$X_i \to X_i^1 \to X_i^2 \to \cdots \ldots X_i^m \to X_i' \quad (17)$$

function LNDS(X_i)

(1) Begin
(2) for d =1 to m do
(3) for k = 1 to n do
(4) $R_i^{old} \leftarrow X_i^d$
(5) flag ← CheckResource(R_k,Re,Sk)
(6) if flag Then
(7) Begin
(8) $X_i^d \leftarrow R_k$
(9) val← EvalFitness(X_i)
(10) if (val < $fitness_i$) then
(11) $fitness_i$← val
//restore to its initial value
(12) else$X_i^d \leftarrow R_i^{old}$

```
(13)End
(14)return Xi, fitnessi
 (15)End
```

The function CheckResource (Re_k, Re, Sk) returns true, if Re_k satisfies the skill type and skill level constraints for task J_d. The constraints are depicted in Figure 3. If the $fitness_i$ is not improved, the value in position #d,X_i^d is restored to its initial value. When the fitness value decreases, the complemented value at position #d is kept, and the LNDS will try to diminish this value once more by complementing the next resource Re_{k+1}. The algorithm continues with the next positions X_{i+1}, if the current position X_i is a local optimum, then the LNDS will not improve the value $fitness_i$.

The proposed dynamic adaptive local particle swarm optimization (AdaL-PSO**)** algorithm can be summarized as follows:

AdaL-PSO Algorithm

```
(1)Begin
(2) Load and Valid iMopse dataset
(3) t← 0
(4) Size ← size of the population
(5) for i =1 to Size
     vi← random vector; xi← random vector
(6) end for
(7) f(t)←calculate fitness, makespan
(8) fitness.Add(f(t))
(9) while(Stop criterion)
(10) calculateθ according to(14)
(11) for i = 1 to Size
(12)   Calculate P_best_i
(13)   P_lbest_i ← LNDS(x_i)
(14)   update velocityvi according to formula(16)
(15)   update vector positionxi according to formula(2)
(16) end for
(16) Update P_gbest
(15) gbest← Makespan(P_gbest)
(16)
(16) t← t+1
(17)end while
(18)return makespan
 (19)End
```

Where:

(i) Stop criterion: the loop *while - end while* would be stopped if the difference between *makespan* of some continuous generations is smaller than a threshold.

(ii) Size: number of solutions in the population

(iii) N: number of tasks

As shown in line 10, the values of inertia weight are dynamically changing and are determined based on the fitness function and line 13, the local best position of the ith particle is calculated by the LNDS function.

At the t-th iteration, the position vector of particle i-th is updated based on the best position of this particle up to time t and the local best of this particle, the p_{lbest_i} is calculated by the local neighborhood descent search method. This method only explores different neighborhoods randomly, it also often runs faster than previous local search algorithms and gives good quality solutions.

We see, each subpopulation will converge to a different optimal value in the search space due to the presence of local attractive features. Thus, niches from the overall population don't have any knowledge of domain problem. However, subpopulation has only few particles, the search capabilities of this model are quite poor, especially in the large search spaces and with many extreme points. To overcome this disadvantage, we propesed a new method to improve the search process in an local best model, which is called local neighborhood descent search (LNDS).

The inertial weights in the proposed algorithm are updated according to the dynamic adaptive method based on the sigmoid function, using formula (15). The value of the inertial weights is updated according to the value of the objective function. This method allows timely adjustment of particle speed, gets out of a locally extrema solution and improves convergence speed.

5. EXPERIMENT AND ANALYSIS

5.1. Simulation datasets

We use two datasets: benchmark dataset iMOPSE and real dataset TNG [35] to evaluate the performance of the proposed algorithm. The above two datasets are also used for GreedyDO and GA algorithms, the experimental results of the proposed algorithm are compared with the experimental results of the two algorithms GreedyDO and GA.

Table 3. iMOPSE dataset.

Dataset instance	Tasks	Resources	Precedence Relations	Skills
Q1	100	5	48	9
Q2	100	5	64	15
Q3	100	5	64	9
Q4	100	10	64	9
Q5	100	10	65	15
Q6	100	20	65	15
Q7	100	20	65	9
Q8	200	10	84	9
Q9	200	10	85	15
Q10	200	20	55	9
Q11	200	20	97	15
Q12	200	20	97	9
Q13	200	40	45	9
Q14	200	40	90	9
Q15	200	40	91	15

The benchmark iMOPSE is a dataset for investigating optimization algorithm, which consists of 15 groups and is presented in Table 3.

The TNG dataset, which is a digitized data set from the industrial sewing production of the TNG company [38], is presented in Table 4.

Table 4. TNG dataset.

Dataset instance	Number of tasks	Number of resources	Precedence Relations	Skill Levels	Project Time
TNG_1	71	37	1026	6	409
TNG_2	71	39	1026	6	325
TNG_3	71	41	1026	6	296
TNG_4	71	45	1026	6	392
TNG_5	137	37	1894	6	1174
TNG_6	137	39	1894	6	1052
TNG_7	137	41	1894	6	871
TNG_8	137	45	1894	6	996

5.2. Parameters and system settings

The simulation is conducted using: Simulation environment: Matlab ver. 2014

Simulation tool: the performance of previous algorithms such as GA is fairly evaluated using GARunner [13], the simulation tool provided by the author of those algorithms themself.

(i) Input data: 15 iMOPSE instances and TNG that are described in Table 3 and Table 4

(ii) Number of solutions in the population N_p: 100

(iii) Number of generations N_g: 40,000

(iv) Since all of the considered algorithms are metaheuristic algorithms, so they do not guarantee the best solution. To evaluate the algorithms, we run on each instance 30 times to get the average value, the standard deviation value, and the best value.

5.3. Performance analysis

5.3.1. Performance on iMOPSE dataset

Table 5. The makespan of the best solutions found by GreedyDO and GA.

Dataset instance	**GreedyDO**	**GA**
Q1	779	528
Q2	640	527
Q3	597	508
Q4	533	296
Q5	426	286
Q6	310	240
Q7	408	181
Q8	999	567
Q9	706	549
Q10	999	312
Q11	680	424

Q12	816	321
Q13	821	209
Q14	963	211
Q15	519	200

Table 6. The average, the best, and the standard deviation of makespan obtained from GA and AdaL-PSO.

Dataset instance	**GA**			**AdaL-PSO**		
	Avg	**Best**	**Std**	**Avg**	**Best**	**Std**
Q1	535	528	9.7	487	482	0.4
Q2	530	527	2.5	505	505	1.3
Q3	521	508	9.9	492	475	1.0
Q4	305	296	6.6	261	255	1.6
Q5	290	286	5.0	264	259	0.5
Q6	240	240	0.0	205	209	0.4
Q7	187	181	4.5	171	157	1.0
Q8	583	567	11.4	533	524	3.0
Q9	555	549	4.9	487	491	0.6
Q10	318	312	4.2	310	292	0.4
Q11	438	424	9.7	332	332	0.1
Q12	326	321	6.2	311	302	3.0
Q13	213	209	2.9	217	208	1.6
Q14	215	211	3.1	214	202	1.0
Q15	205	200	3.4	204	190	3.4

The experimental results are presented in Tables 5 and 6, where the makespan value is calculated in hour unit. The data in the above tables are the makespan values of the best solution found by GreedyDO, GA, and the proposed algorithm (AdaL-PSO). Table 5 shows that the makespans found by the GA algorithm are always better than the makespans found by the GreedyDO algorithm.

The comparison results between the average value, the best value and the standard deviation of makespan found by the GA algorithm and the proposed algorithm AdaL-PSO are presented in Table 6.

The average makespan time for all the four algorithms is computed, the results in Table 5 and Table 6 show that:

(i) Based on 15 instances of iMOPSE experimental dataset, the mean of makespan found by AdaL-PO, GA and Greedy algorithms is: 679.7 (hours); 357.2 (hours); and 325.5 (hours), respectively. We see that the solution of AdaL-PSO algorithm is better than GA by 8 % and better than Greedy by 52 %.

(ii) Based on the mean column in Table 5, we see that the solution found by AdaL-PSO is better than GA from 0.7 % to 24.8 % and better than Greedy from 17 % to 77.5 %.

(iii) In the proposed AdaL-PSO algorithm, we use adaptive inertia weight updated strategy based on the fitness value, which controls the local search and convergence to the global optimum solution. Therefore, the AdaL-PSO algorithm has higher convergence speed and stability. The data in column std of Table 5 demonstrates this problem. The average of the std (standard deviation) column of the AdaL-PSO algorithm is 1.3, while the average of the

std (standard deviation) column of the GA algorithm is 5.6, which proves that the stability of the AdaL-PSO algorithm is better than the GA.

5.3.2. Performance on TNG dataset

The makespan values of the best solution for GreedyDO, GA and AdaL-PSO algorithms are presented in Table 7. The duration of actual projects in TNG company is shown in the TNG column of Table 7.

Table 7. Comparison between the makespan of real TNG project, makespan of schedule founded by GreedyDO, GA, and AdaL-PSO.

Dataset instance	TNG	GreedyDO	GA	AdaL-PSO
TNG_1	409	236	201	163
TNG_2	325	243	198	167
TNG_3	296	258	212	165
TNG_4	392	248	176	165
TNG_5	1174	972	751	702
TNG_6	1052	963	791	710
TNG_7	871	834	810	723
TNG_8	996	906	720	671

From the experimental data in Table 7, we see that the schedule generated by the AdaL-PSO algorithm is always better than the schedule generated by the GA and Greedy algorithms. The data in Table 7 and Figure 5 also show that the schedule generated by the evolutionary algorithms is always smaller than the actual schedule at the TNG company.

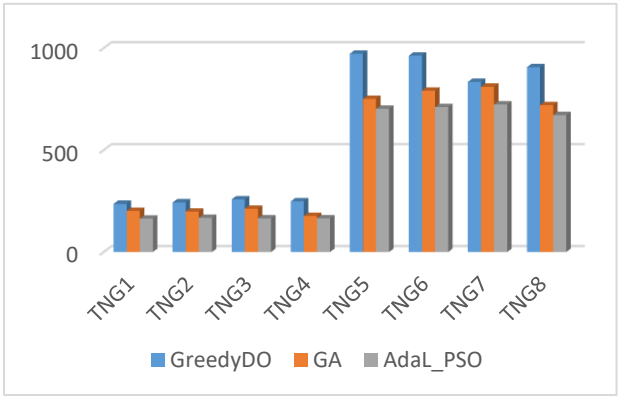


Figure 5. Comparison of makespans between AdaL-PSO, GA and factory TNG's solution.

Averaged over 8 instances of the TNG dataset, the makespan values generated by the GreedyDO, GA and AdaL-PSO algorithms are: 582.5 (hours); 482.375 (hours); and 433.25 (hours), respectively. Therefore, on average, the schedule generated by the AdaL-PSO algorithm is about 10 % smaller than the schedule generated by GA and about 25 % smaller than the schedule generated by the GreedyDO algorithm.

From this experimental result, it has been proved that the solution found by the proposed algorithm is always better than the previous algorithms. With the combination of neighbor

architecture and adaptive inertial weight update strategy, the AdaL-PSO algorithm always meets both the important properties of evolutionary algorithms that are fast convergence and averting getting trapped in local extrema.

The data in Table 7 also shows that the schedule generated by the AdaL-PSO algorithm is 15 % to 60 % smaller than the actual schedule at TNG company.

6. CONCLUSION AND FUTURE WORK

This paper defines and introduces the formulation of MS-RCPSP, which is one of the famous scheduling problems and is applied in many fields such as science, industry, and life. The mathematical model of the MS-RCPSP problem is also presented in section 3 of this paper, then we have proposed a new scheduling algorithm based on Particle Swarm Optimization method, called AdaL-PSO.

Resource-Constrained Project Scheduling Problems and Multi-skilled Resource-Constrained Project Scheduling Problem (RCPSP and. MS-RCPSP) have been essential topics of study over the last years. MS-RCPSP is suitable in projects with multi-skilled human resources or multi-functional machines. Based on Particle Swarm Optimization, we have devised a new metaheuristic algorithm that has the ability to outperform all the previous algorithms. We built a new function, which updates the inertial weight dynamically based on fitness value. This is the key component that creates the strength of the proposed algorithm.

To evaluate the efficiency of the proposed algorithm, we conducted many experiments based on simulated and real datasets. The experimental results are compared with the previous algorithms. The simulation result shows that our developed algorithm is more effective than existing algorithms. AdaL-PSO not only achieves better quality solutions but also converges to the global extremum faster than previous algorithms.

In the future, we wish to investigate how to improve the AdaL-PSO algorithm to solve bigger instances within a reasonable makespan.

CRediT authorship contribution statement. Author 1: Toan Phan Thanh is the principle investigator in this research and he is the proponent of the new algorithm. Author 2: Do Van Tuan helped with algorithm design and formalism and the write up of the manuscript.

Declaration of competing interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

1. AfsharN. B. - Multi-skilling in scheduling problems: A review on models, methods and applications, Computers & Industrial Engineering **151** (2021) 107004. https://doi.org/ 10.1016/j.cie.2020.107004.
2. Alirezaei, Mahsa, SeyedT. A. N., SeyedA. A. N. - A bi-objective hybrid optimization algorithm to reduce noise and data dimension in diabetes diagnosis using support vector machines, Expert Systems with Applications **127** (2019) 47-57.
3. Mejia, Oliver Polo, et al. - A new RCPSP variant to schedule research activities in a nuclear laboratory, 47th International Conference on Computers and Industrial Engineering (CIE47), 2017.

4. Blazewicz J., Lenstra J. K., Kan A. - Scheduling subject to resource constraints: Classification and complexity, Discrete Applied Mathematics **5** (1983) 11-24.
5. Myszkowski, Paweł B., Marek E. S., Krzysztof S. - A new benchmark dataset for multi-skill resource-constrained project scheduling problem, Federated Conference on Computer Science and Information Systems (FedCSIS), IEEE, 2015. DOI: 10.15439/2015F273.
6. Yuanyuan Z., Naixue X., Kim T. - Channel assignment and scheduling in multichannel wireless sensor networks, 2008 33rd IEEE Conference on Local Computer Networks (LCN), Montreal, Que. (2008) 512-513.doi: 10.1109/LCN.2008.4664215.
7. Cheng H., Naixue X., Yang L. T. - Distributed Access Scheduling Algorithms in Wireless Mesh Networks, 22nd International Conference on Advanced Information Networking and Applications (aina 2008), Okinawa. (2008) 509-516.DOI: 10.1109/AINA.2008.29.
8. Wei G., Vasilakos A. V., Naixue X. - Scheduling Parallel Cloud Computing Services: An Evolutional Game, 2009 First International Conference on Information Science and Engineering, Nanjing. (2009) 376-379.DOI: 10.1109/ICISE.2009.1046.
9. Nie W., Naixue X., Park J. H., Yeo S. - A Fair-Oriented Two-Level Scheduling Scheme for Downlink Traffic in WiMAX Network, 2010 2nd International Conference on Information Technology Convergence and Services, Cebu. (2010) 1-6.DOI: 10.1109/ITCS.2010.5581291.
10. Zheng W.,Naixue X., Ghani N., Min P. A., Vasilakos A. V., Liang Z. - Adaptive scheduling for wireless video transmission in high-speed networks, 2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Shanghai. (2011) 180-185.DOI: 10.1109/INFCOMW.2011.5928803
11. Dong P., Xie J., Tang W., Xiong N., Zhong H., Vasilakos A. V. - Performance Evaluation of Multipath TCP Scheduling Algorithms, IEEE Access.**7** (2019) 29818-29825.DOI: 10.1109/ACCESS.2019.2898110.
12. Klein R., Scheduling of Resource. - Constrained project, Springer Science Business Media NewYork, Kluwer Academic Publisher, ISBN 978-1-4613-7093-2, 2000.
13. Myszkowski P. B., Laszczyk M., Nikulin I., Skowronski E. - iMOPSE: a library for bicriteria optimization in Multi–Skill Resource–Constrained Project Scheduling Problem, Soft Computing. **23** (10) (2019) 3397-3410.
14. Cheng H., Xiong N., Huang X., Yang L. T. - An Efficient Scheduling Model for Broadcasting in Wireless Sensor Networks, 2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum, Cambridge, MA. (2013) 1417-1428.DOI: 10.1109/IPDPSW.2013.88
15. Lin B., Guo W., Xiong N., Chen G., Vasilakos A. V., Zhang H. - A Pretreatment Workflow Scheduling Approach for Big Data Applications in Multicloud Environments, IEEE Transactions on Network and Service Management **13** (3) (2016) 581-594.DOI: 10.1109/TNSM.2016.2554143.
16. Lin B., Guo W., Chen G., Xiong N., Li R. - Cost-Driven Scheduling for Deadline-Constrained Workflow on Multi-clouds, 2015 IEEE International Parallel and Distributed Processing Symposium Workshop, Hyderabad **15** (4) (2015) 1191-1198. DOI: 10.1109/IPDPSW.2015.56.

17. Tan L., Zhu Z, Ge F., Xiong N. - Utility Maximization Resource Allocation in Wireless Networks: Methods and Algorithms, IEEE Transactions on Systems, Man, and Cybernetics: Systems. **45** (7) (2015) 1018-1034.DOI: 10.1109/TSMC.2015.2392719.
18. Guo P., Cheng W., Wang Y. - A general variable neighborhood search for single-machine total tardiness scheduling problem with step-deteriorating jobs, J. Ind. Manag. Optim. **10** (4) (2014) 1071-1090.
19. Kavitha S., Venkumar P. - A vibrant crossbreed social spider optimization with genetic algorithm tactic for flexible job shop scheduling problem, Measurement and Control **53** (1-2) (2020).
20. Agrawal, Prakash A., Choudhary A., Arvinder K. - An Effective Regression Test Case Selection Using Hybrid Whale Optimization Algorithm, International Journal of Distributed Systems and Technologies (IJDST) **11 (**1) (2020) 53-67.
21. Guo W., Park J. H., Yang L. T., Vasilakos A. V., Xiong. N., Chen. G. - Design and Analysis of a MST-Based Topology Control Scheme with PSO for Wireless Sensor Networks, 2011 IEEE Asia-Pacific Services Computing Conference, Jeju Island. (2011) 360-367.DOI: 10.1109/APSCC.2011.20.
22. Zhuang X., Cheng H., Xiong N.,Yang L. T. - Channel Assignment in Multi-Radio Wireless Networks Based on PSO Algorithm, 2010 5th International Conference on Future Information Technology, Busan. (2010) 1-6.DOI: 10.1109/ FUTURETECH.2010.5482773.
23. Chen Z., Chyu C. - An Evolutionary Algorithm with Multi–Local Search for the Resource-Constrained Project Scheduling Problem, Intelligent Information Management **2** (3) (2010) 220-226. DOI: 10.4236/iim.2012.23026
24. Das P. P., Acharyya S. - Simulated Annealing Variants for Solving Resource Constrained Project Scheduling Problem: A Comparative Study, Proceedings of 14th International Conference on Computer and Information Technology (2011) 469-474. DOI: 10.1109/ICCITechn.2011.6164835
25. Hosseinian, Amir H., Baradaran V. - P-GWO and MOFA: two new algorithms for the MSRCPSP with the deterioration effect and financial constraints (case study of a gas treating company), Applied Intelligence **50** (2020) 2151-2176. DOI: 10.1007/s10489-020-01663-x
26. Hosseinian, Amir H., Baradaran V., Mahdi B. - Modeling of the time-dependent multi-skilled RCPSP considering learning effect. Journal of Modelling in Management **14** (3) (2019). DOI: 10.1108/JM2-07-2018-0098.
27. Nemati L. R., Hamed D. A., Hamid N. - Multi-mode resource constrained project scheduling and contractor selection: Mathematical formulation and metaheuristic algorithms, Applied Soft Computing **81** (2019) 105533. https://doi.org/ 10.1016/j.asoc.2019.105533
28. Younis M. T., Yang S. - Hybrid meta-heuristic algorithms for independent job scheduling in grid computing, Applied soft computing **72** (2018) 498-517. https://doi.org/10.1016/ j.asoc.2018.05.032
29. Tian Y., Xiong T., Liu Z., Mei Y, Wan L. - Multi-Objective multi-skill resource-constrained project scheduling problem with skill switches: Model and evolutionary

approaches, Computers& Industrial Engineering **167** (C) (2022). DOI:https://doi.org/10.1016/j.cie.2021.107897.

30. Shima J., Behrouz A. N., Seyed T. A. N. - Preemptive Multi-skilled Resource Investment Project Scheduling Problem; Mathematical Modelling and Solution Approaches, Computers & Chemical Engineering **9 6**(C) (2017) 55-68. DOI:10.1016/j.compchemeng.2016.11.001
31. Najafzad H., Hamed D. A., Reza N. L. - Multi-skill project scheduling problem under time-of-use electricity tariffs and shift differential payments, Energy **168** (2019) 619-636.
32. Kennedy J., Eberhart R. - Particle swarm optimization, Neural Networks, 1995. Proceedings, IEEE International Conference **4** (1995) 1942-1948.
33. Zavala A. E. M. - EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IIA Comparison, A Comparison Study of PSO Neighborhoods, Springer Verlag Berlin Heideberg, 251-295, ISBN 978-3-642-32725-4, 2013.
34. Shi Y., Eberhart R. - A modified particle swarm optimizer, IEEE World Congress on Computational Intelligence, Anchorage, AK, USA. (1998) 69-73. DOI:10.1109/ICEC.1998.699146.
35. TNG Investment and Trading Joint Stock Company, 434/1 Bac Kan street - Thai Nguyen city, Viet Nam, Website http://www.tng.vn