JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 38, 895-907 (2022)
DOI: 10.6688/JISE.202209_38(5).0001

An Optimized Modelling and Simulation on Task Scheduling for Multi-Processor System using Hybridized ACO-CVOA

ANNU PRIYA AND SUDIP KUMAR SAHANA+
Department of Computer Science Engineering
Birla Institute of Technology
Mesra, Ranchi, 835215 India
E-mail: annu.priya12@yhaoo.com; sudipsahana@bitmesra.ac.in

Task allocation on the multi-processor system distributes the task according to capacity of each processor that optimally selects the best. The optimal selection of processor leads to increase performance and this also impact the makespan. In task scheduling, most of the research work focused on the objective of managing the power consumption and time complexity due to improper selection of processors for the given task items. This paper mainly focusses on the modelling of the optimal task allocation using a novel hybridization method of Ant Colony Optimization (ACO) with Corona Virus Optimization Algorithm (CVOA). There are several other methods that estimate the weight value of processors and find the best match to the task by using the traditional distance estimation method or by using standard rule-based validation. The proposed algorithm searches the best selection of machines for the corresponding parameters and weight value iteratively and finally recognizes the capacity of it. The performance of proposed method is evaluated on the parameters of elapsed time, throughput and compared with the state-of-art methods.

Keywords: task allocation, hybrid optimization algorithm, ant colony optimization (ACO), corona virus optimization algorithm (CVOA), multiprocessor

1. INTRODUCTION

In a real-time application system, the processor in an embedded system needs to provide the proper result within the deadline. The major problem facing for task scheduling in embedded system is the power consumption due to the improper allocation of the processor. To improve the performance and to reduce the issues in resource allocation, the suitable scheduling model need to support for the optimal selection of task at each timeline process. The methods need to select the best processor by analyzing the weight value or the capacity of resources to do the task in particular time limit.

In case of parallel processing, the whole job is split into several blocks and given to the resource allocation system that validate the amount and find the best processor or resources to complete the task within the time limit specified for the model. Several scheduling techniques were focused on the allocation of task based on certain conditions and fixed rule-based process. In that, some were used the supervised machine learning technique to achieve better efficiency on the parameters such as satisfaction of deadline, minimization of total execution time, reduction of power consumption based on the amount of task running in a processor, *etc.* with fixed property validation.

In the proposed work, by using the optimization model, the best processor for the given task is identified and the scheduling process is performed for the appropriate resources in the embedded system. The limitations in commonly used optimization tech-

Received July 22, 2021; revised September 30, 2021; accepted November 26, 2021.
Communicated by Wei Kitt Wong.
+ Corresponding author.

niques are the random selection of the tasks for processor by considering only the parameter of weight of the task matching to the processor based on traditional distance estimation method. In the proposed work, a metaheuristic approach using hybrid of ACO and Corona Virus Optimization Algorithm (CVOA) is used to perform resource allocation for the given tasks.

The paper is organized in following sections: The related work survey and the observation about the methods in existing systems are discussed in Section 2. The proposed algorithm steps and its functionality of development are discussed in Section 3. Section 4 explains about the result and comparative study of proposed model for parameter analysis in the task scheduling process. And finally, conclusion is drawn in Section 5.

2. RELATED WORK

Several research works have been developed for the best task identification and optimal solution for the resource update in the multiprocessor scheduling process. A brief discussion on the related work is explained in this section.

In the task allocation model, there are several methods to identify the processors for corresponding tasks. Some of the methods perform task allocation both using the supervised classification and clustering technique or optimization technique. Out of them optimization techniques are much more popular. In [1], the authors proposed a threshold-based task allocation process on the Bayesian Nash equilibria. That performed a communication free task assignment for the swarm robotics. To solve the Multi-Robot coalition formation problem of robotics, the immigrant's based genetic algorithm (GA) was used for optimal task allocation [2]. The adaptive parameters of GA enhance the quality of the solutions during evaluations.

In [7], the authors proposed heuristic methods for task allocation and collision-free task planning in a multi-robot system. The optimal path was obtained by reducing the time taken by allocating tasks to each inspecting robot. For the Internet-of-Things-based application, an evolutionary-based task allocation model was proposed [3] to solve energy consumption. It controls the communications and processing between the processors in IoT objects. The Ant Colony Optimization (ACO) has more advantages than other optimization methods in various applications among several optimization models.

A novel strengthened pheromone update mechanism was designed for the ACO optimization to improve the pheromone on the edges in [4]. That reinitialized the pheromone matrix at each iteration and strengthened the convergence speed of the algorithm. In [5], a hybrid optimization algorithm of ANT and BEE colony optimization was proposed for feature selection and classification. The Artificial Bee Colony optimization is further improved by the Gene Recombination Operator (GRO) method [6] for the numerical function optimization technique. For the multi-label classification process, there are numerous functionality to find the relevancy between attributes. Many of them have been recently applied to solve different variants of the scheduling problem [14] in the context of multi-robot applications, real-time engineering optimization problems, ocean sampling problem, data mining, multi-agent systems. From the existing papers related to the various applications, the Covid Spread optimization can improve the searching performance and reduce the error rate with a sufficient iteration limit. If combined with the Ant Colony Optimization (ACO), this CS optimization improves the path identification and finds the best matching task for the resources. This type of hybrid combination achieved better performance than the other state-of-art methods in the task scheduling concept. The detailed description of the proposed work is in the following sections.

3. PROPOSED WORK

As per literature, Swarm and Evolutionary techniques are much more fruitful in complex task allocation problems as compared to the others and also, hybridization is one of the key approaches to improve the performance. We have considered ACO, a swarm-based approach due to the uniqueness of indirect communication and ability to tackle combinatorial optimization problems and Covid Virus searching concept which is very fruitful for complex search problems. Furthermore, the major drawback of retardation in convergence speed at the later stage of ACO is eliminated by the amalgamation of Corona Virus Optimization Algorithm (CVOA) which provides wider search space areas in less iterations. In actual scheduling arrangement each processor runs the proposed ACO-COVA algorithm separately to generate a possible scheduling sequence. The overall flow diagram of proposed system is shown in Fig. 1.

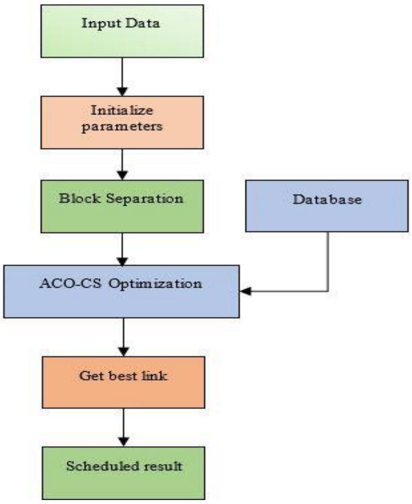

Fig. 1. Overall flow diagram of proposed work.

Since, the Covid virus spreading to the human being is very fast and makes high impact to them. the advantages of Covid search is used in the optimization process of ACO. This can optimally select the best machine / processor for the given jobs / tasks in multiprocessor system. The detailed description about the ACO, CVOA and proposed algorithm of hybrid ACO-CVOA optimization is described in the following subsections.

3.1 Ant Colony Optimization (ACO)

The traditional concept of Ant Colony Optimization (ACO) is to find the best path using the concentration of pheromone, a chemical substance, lay on their path. In this model, the ants are initially split into different paths as a trials and deposit pheromone on its corresponding route. Based on the probability, the ant's movement are chosen, and pheromone is updated on path. The possible movement p_{ij}^h of ant 'h' from position i to position j can be represent as

$$p_{ij}^h \begin{cases} \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{l\in N_i^h}\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}, & \text{if } j \in N_i^h \\ 0, & \text{else} \end{cases}. \tag{1}$$

Where, N_i^h – Number of coordinates position for 'h' ant

η_{ij} – Heuristic Information (*i.e.*) $\eta_{ij} = 1/d_{ij}$.

d_{ij} – Distance between 'i' and 'j' position.

'α' and 'β' are the relative parameters of pheromone 'τ' and 'η'.

'l' – Length of position changes for an ant movement.

The rule to update the pheromone for each updated position of ant can be represent by

$$\tau_{ij} = ((1-p)\tau_{ij}) + \sum_{h=1}^{m}\Delta\tau_{ij}^h. \tag{2}$$

Where $\rho \in [0, 1]$ – Evaporation rate of pheromone.

The pheromone update for the best movement of ant can be identify by the total length of pheromone deposited in the best path at each iteration count. This can be represented as

$$\Delta\tau_{ij}^{bs} = \frac{1}{L_{bs}}. \tag{3}$$

By considering this, the Eq. (2) can be updated as

$$\tau_{ij} = ((1-\rho)\tau_{ij}) + \Delta\tau_{ij}^{bs}. \tag{4}$$

This was also validated by the limit of $\tau_{\min}$ to $\tau_{\max}$ range of interval. This is to reduce the loss of energy due to more amount of pheromone should not be deposit in one particular position. The position of ant movement can be consider based on the update of pheromone best selection with heuristic selection. According to that, the position 'j' can be evaluated by the Eq. (5).

$$j = \arg\max_{l\in N_i^h}\{\tau_{ij}\eta_{ij}^{\beta}\} \tag{5}$$

From this, the randomized movements of the ants were regularized by the updated position and it defines the global and local pheromone that are can be represent as in the Eqs. (6) and (7).

The global pheromone,

$$\tau_{ij} = ((1-\rho)\tau_{ij}) + \rho\Delta\tau_{ij}^{bs}. \tag{6}$$

The local Pheromone,

$$\tau_{ij} = ((1-\varphi)\tau_{ij}) + \varphi\tau_0. \tag{7}$$

Where τ_0 – initial value of the pheromone 'τ'
φ – Decay coefficient of pheromone particles.

From these steps, the traditional ACO algorithm selects best path of movement for the ant which indicates the best selection of attributes from the overall dataset. The advantages of ACO over other optimization algorithm is that it reaches the best path at minimum number of iteration count which may reduce the time complexity of system. This also achieved better convergence that results in reduction of cost value at each iteration and also reduce the error rate in validating the best match in data points.

3.2 Corona Virus Optimization Algorithm (CVOA)

Corona Virus Optimization Algorithm (CVOA) [8] make selection of best one from the infected populations. The best individual from the newly affected population is estimated for each iteration and update the recovered population for searching new series. Algorithm 1 shows the steps followed in CVOA optimization algorithm.

Algorithm 1: CVOA Algorithm

Input: Input data matrix, $\{D\}$
Output: Best selection of individuals, F_s
1: Initialize infected population as D.
2: Initialize newly affected population set and recovered population as a set and initialize time as $t = 0$.
3: Initialize infected patient, and best individuals as zero list.
4: Initialize pandemic duration as maximum iteration, 'Iter_Max'.
5: **While** (t < Iter_Max) && (length(D) > 0), **do**
6: Estimate dead count from the overall infected population by

$$Dead = D_i \text{ if } (R > P_{DIE})$$

Where, R – Random value,
P_{DIE} – Die probability constant
7: **For** $j \in D$:
8: Estimate the infected population 'aux' with respect to dead list. This can be represented as in Eq. (8).
9: **If** (aux $\neq$ {}), **then**
10: newinfectedPopulation = aux
11: **End if**
12: **End For**
13: Estimate current best individual, I_{CB} using Eq. (10).
14: **If** (fitness(I_{CB}) > bestIndividual), **then**
15: bestIndividual, $F_s(t) = I_{CB}$
16: **End if**
17: recoveredPop = D
18: Infected Population, D = newinfectedPopulation
19: $t = t + 1$
20: **End while**

The optimization algorithm selects the best attributes by estimating the individuals that are stable from the affected population data. Initial population is randomly initialized. The initial population consists of infected individual. Each infected individual has a probability of dying, according to the COVID-19 death rate. This rate is set to ~5% by the scientific community [15]. The infected population is updated in each iteration to identify the individuals at each time interval. In that, the affected population 'aux' for the dead population list is represented as in Eq. (8). The newly affected population are estimated by using 4 different conditions which is based on the 2 different state namely: spreader rate (SR) and traveler rate (TR).

$$aux = \begin{cases} f(Y_{SO}), \text{ if } (R1 < P_T) \text{ and } (R2 < P_{SS}) \\ f(Y_{SSO}), \text{ if } (R1 < P_T) \text{ and } (R2 \geq P_{SS}) \\ f(Y_{ST}), \text{ if } (R1 \geq P_T) \text{ and } (R2 < P_{SS}) \\ f(Y_{SST}), \text{ if } (R1 \geq P_T) \text{ and } (R2 \geq P_{SS}) \end{cases} \tag{8}$$

Where, R1 & R2 are the random values. P_T and P_{SS} are the probability of Travel and Super Spreader respectively. $f(Y_{SO})$, $f(Y_{SSO})$, $f(Y_{ST})$, and $f(Y_{SST})$ are the functions that identifies new infected population in different state for the input of Spreader rate, Ordinary rate and Super Spreader rate. So, finally the state function is represented in Eq. (9),

$$f(Y_{State}) = \begin{cases} i, & \text{if } Rule1 \\ i, & \text{if } Rule2. \\ \text{Recovery}, & \text{else} \end{cases} \tag{9}$$

Where,

$Rule1 = (i \notin \alpha)$ and $(i \notin \beta)$ and $(R4 > P_{ISO})$
$Rule2 = (i \notin \alpha)$ and $(i \in \beta)$ and $(R3 > P_{ReInfect})$

'i' $\in$ infected people based on Spreader Rate and Traveler Rate.
R3 & R4 are random values.
'α' and 'β' dead and recovered population respectively.

P_{ISO} – Probability of Isolation.
$P_{ReInfect}$ – Probability of reinfected population after recovered in previous.

Also, the current best individuals 'I_{CB}' from the validation of affected and recovered population can be estimate by using the Eq. (10).

$$I_{CB} = i, \text{ if } (fit(i) > bestFit) \tag{10}$$

Where i – Infected population.

3.3 Proposed Hybrid ACO-CVOA

In the proposed optimization algorithm, the ACO particles are used as the moving objects based on the parameters estimation of Corona Virus Optimization Algorithm

(CVOA) for each iteration. For task allocation process, the optimization evaluates the task weight matrix and the processor availabilities and its capability to complete the work in scheduled time. This also consider the inter-link range of tasks that depends on the output of one application to another.

According to the Corona Virus Optimization Algorithm (CVOA), the individual particles are estimated from the infected population for updating result at each iteration. The dead rate and the recovery are the other additional parameters that are considered to find the best selection of individual particles. In the hybrid concept of optimization, the probability estimation for the calculating infected population was considered as the decay probability of ACO algorithm. The main contribution of CVOA in ACO is to evaluate the best position and strength of path by the updating process of new population and the fitness value of best individual of the iteration time. Here, tasks are assigned to the resources such that the total execution time (makespan) of the schedule can be minimized, while satisfying given precedence constraint between the activities and resource constraint. This will reach the convergence of optimization at minimum iteration count and reduce the cost value compare to traditional ACO algorithm. Algorithm 2 describes the step-by-step process of proposed ACO-CVOA optimization algorithm.

Algorithm 2: ACO-CVOA Algorithm

Input: Input data matrix, $\{D\}$
Output: Best selection of attributes, F_s
Initialize particles 'p' for data 'D'.
Initialize pheromone, τ_0.
Initialize random path coordinated as zero list.
Initialize maximum iteration, 'Iter_Max'.
Initialize pheromone energetic parameters such as 'α' and 'β'
While (t < Iter_Max) && (length(D) > 0), **do**
    Place ants at random coordinate position.
    Estimate the possible movement of ant by Eq. (1)
    Estimate the strength of pheromone for the updated position by using Eq. (11).
    Find the best rule to estimate the best match of path which is nearest to theparticles using Eq. (2).
    Update the selected path and position using Eq. (5).
    Estimate the fitness value for the current position using Eq. (13).
    If current_position > previous_position, **then**
        Update position coordinates
        $F_S(t) = i$
    End if
End while

The objective function of the proposed algorithm ACO-CVOA is to find the schedule such that the makespan can be minimized, while satisfying given precedence constraint between the activities and resource constraint. Makespan can be defined as the overall time taken by the resources to perform of all tasks. The objective function is denoted as in Eq. (11) follows:

$$F(x) = Min\{\sum_{i}^{n} makspan\} \tag{11}$$

Where, i = number of jobs.

For the proposed algorithm ACO-CVOA, the estimation of path can be updated as in Eq. (12) from the Eq. (8) which can be written as

$$aux = \begin{cases} f(Y_{WD}), \text{ if } (R5 < P_M) \text{ and } (\forall P_W) \\ f(Y_{MD}), \text{ if } (R6 \geq P_M) \text{ and } (\forall P_W) \end{cases}. \tag{12}$$

Where, R5 & R6 are the random values. P_M and P_W are the probability of movement and weight of particles respectively. $f(Y_{WD})$, and $f(Y_{MD})$ are the functions that identifies new path identification in different state for the input of pheromone decay rate and the distance of movement factor for an ant from one coordinate to another position. This function can be represented as Eq. (13).

$$f(Y_{State}) = \begin{cases} i, & \text{if } Rule1 \\ i, & \text{if } Rule2. \\ 0, & \text{else} \end{cases} \tag{13}$$

Where,

$Rule1 = (i \notin \alpha)$ and $(i \notin \beta)$ and $(R7 > P_{decay})$
$Rule2 = (i \notin \alpha)$ and $(i \in \beta)$ and $(R8 > P_{best})$

'i' $\in$ infected particles that are place in the random position and dropped the pheromone in the selected path.

R7 & R8 are random values.

'α' and 'β' pheromone energetic factor and heuristic factor respectively.

P_{decay} – Probability of decay rate of pheromone.

P_{best} – Probability of selected path compared with previous distance factor.

Also, the current best fitness for particles 'F_S' from the validation of selected path in the ant colony to reach the food can be estimate by using the Eq. (14).

$$F_S = i, \text{ if } (fit(i) > bestFit) \tag{14}$$

Where, i – updated particles of ant.

From this fitness value estimation, the selected path from F_S is considered as the best optimal task for the processor to execute it. The performance of proposed algorithm was discussed in the result analysis for the parameters of error rate and other parameters that represents the resource allocations.

4. RESULT ANALYSIS

This section analyses the performance of proposed work and compare the parameters to other existing methods. The testing and simulation are processed in the python scripts version of 3.7. The result of proposed work is validated for the RCPSP dataset [9] and IMOPSE dataset [10] which are publicly available. The dataset contains 4 different size of job items along with the task information and time sequences to frame the limit of execution. The main purpose of the library is to evaluate the optimal solution for single or multiple resource constrained scheduling process. These set of instances with different job size have been generated by the ProGen (Standard Project Generator). This is a benchmark dataset to evaluate the performance of algorithms for optimal scheduling process.

The performance evaluation is obtained by considering, the makespan of scheduled task of the given population of the proposed technique. This can also analyze by estimating the amount of resource allocation for the given task sizes. The Mean Absolute Percentage Error (MAPE) can be calculated by the Eq. (15).

$$MAPE = \frac{e(t)}{V(t)} \times 100 \tag{15}$$

Where, '$e(t)$' represents the error rate at the time instant 't' and '$v(t)$' be the actual value of it. In that, the error rate can be calculated by using the Eq. (16).

$$e(t) = v(t) - p(t) \tag{16}$$

Where, '$p(t)$' – predicted value at 't'.

This procedure is applied to benchmark RCPSP datasets to measure the makspan of each experiment dataset. The tasks which are not completed within the deadline or misalign with the task flow are treated as error. Fig. 2 shows the initial task model for the input of job size 30. In this figure, the network like structure shows the connectivity of task that must be done in the proper flow. The algorithm that schedule the job for processor need to follow the connecting information and it should not misalign the task from its network. In that, the 30 number of tasks have been labelled in the nodes and its weigh values are in the data matrix.

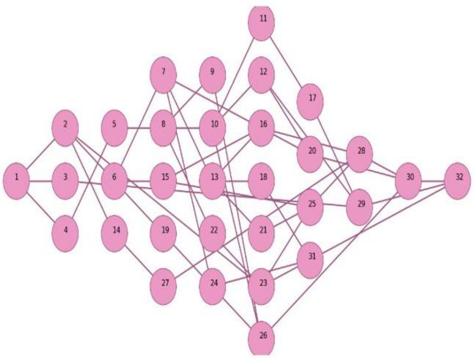

Fig. 2. Initialize map plot of job size 30.

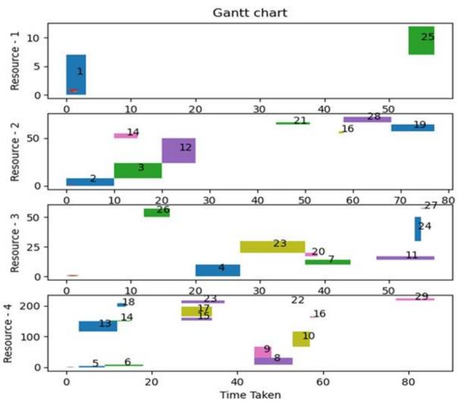

Fig. 3. Gantt chart for resource allocation.

Fig. 3 shows the scheduled task from the proposed optimization algorithm of ACO-CVOA. Each task has been differentiated by the color representation and label along with it. Here, there are 4 number of resources have been selected and mapped as the best processors to complete the task within the deadline. In this graph, the x-axis representing the time taken to execute the tasks and y-axis represents the resource utility to complete the task. Fig. 4 shows the bar chart to represent amount of resources that are utilized for each time instant. This was represented in the range of percentage that displays amount of resources active at that time instant and selects the optimal processors for the given task size. In that graph, the x-label represent the time period to complete the overall task in j30 datasheet of RCPSP dataset and y-label represent the rate of resources allocated at each time period in terms of percentage.

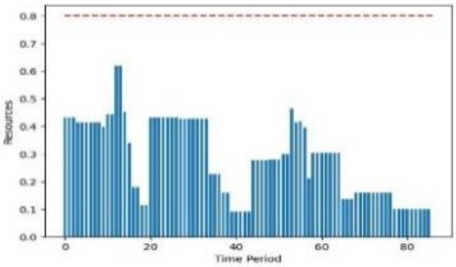

Fig. 4. Resource utilization graph.

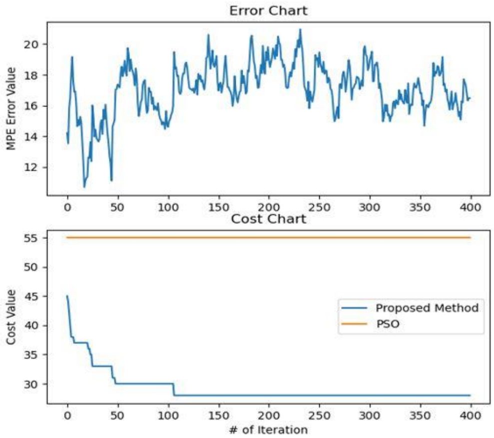

Fig. 5. Error chart.

The error rate and the comparison result of cost value for proposed ACO-CVOA and traditional PSO optimization algorithm for the j30 RCPSP dataset is shown in Fig. 5. In that graph, the first chart represents the error rate at each iteration count which represents that the error value is falling in the range of approximately 10 to 25.

The second chart displays the cost value of ACO-CVOA and PSO algorithm (as PSO is coming to be best among the rest of the considered algorithms in terms of Makespan value). The proposed method achieved convergence much eariler than the PSO optimization. The ACO-CVOA has further tested in the MS-RCPSP D36 IMOPSE DATASET for 400 iterations and compared with the existing systems of GreedyDO [11], HAntCO [12], GA [13], CSM [10] and PSO methods, the comparison table is presented in Table 1.

Table 1. Makespan comparison of ACO-CVOA with [10].

Dataset	GreedyDO	HAntCO	GA	CSM	PSO	ACO-CVOA
100_5_22_15	630	504	516	488	484	**475**
100_10_26_15	370	266	292	247	247	**248**
100_10_47_9	549	297	296	268	260	**246**
100_20_46_15	394	194	206	188	184	**177**
100_20_65_9	408	180	179	174	170	**163**
200_10_128_15	780	522	580	477	473	**460**
200_10_50_15	763	529	586	500	496	**490**
200_20_54_15	488	336	376	329	325	**335**
200_20_55_9	999	313	313	304	300	**276**
200_40_91_15	519	207	211	197	194	**190**

Fig. 6 shows the graphical representation of makespan obtained by proposed algorithm ACO-CVOA with other existing algorithms. It is found that ACO-CVOA is superior as compare to GreedyDO, HAntCO, GA, CSM and PSO. It is found that ACO-CVOA is superior as compare to GreedyDO, HAntCO, GA, CSM and PSO. The proposed ACO-CVOA, assuming local update of ACO takes p unit time and global update requires q unit time for each job, the complexity comes out to be $(p + q) \times n$ for n number of jobs to be scheduled. Considering m number of processors, the complexity of ACO-CVOA finally comes out to be $O((p + q) \times n) = O(n)$ which is also found preserved from the simulation results as well.

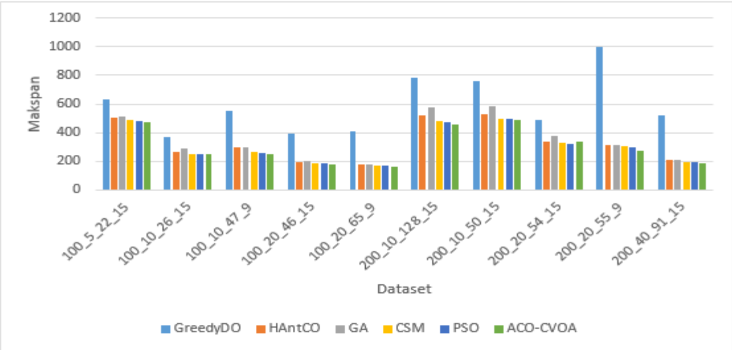

Fig. 6. Comparison ACO-CVOA with other techniques.

5. CONCLUSION AND FUTURE ENHANCEMENT

In this paper, a novel optimization algorithm is proposed for the application of task allocation in multi-processor model. Here, the ACO is integrated with the new optimizer Corona Virus Optimization Algorithm (CVOA) to enhance the search speed of optimal solution in ACO and to increase the performance of ACO in task allocation model. This hybridization process is capable to reduce the error rate and reached the convergence at minimum number of iteration count. The result analysis shows a better performance of the proposed ACO-CVOA optimization algorithm than other state-of-art methods of task scheduling. The comparison chart also represents the enhancement of proposed algorithm over traditional PSO optimization method by the parameter of cost graph for each iteration period. The resource allocation reached an efficient level of rage which reduced maximum percentage of ~30% from overall utilization.

REFERENCES

1. A. Kanakia, B. Touri, and N. Correll, “Modeling multi-robot task allocation with limited information as global game,” *Swarm Intelligence*, Vol. 10, 2016, pp. 147-160.
2. K. Jose and D. K. Pratihar, “Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods,” *Robotics and Autonomous Systems*, Vol. 80, 2016, pp. 34-42.

3. N. Hooshangi and A. A. Alesheikh, "Agent-based task allocation under uncertainties in disaster environments: An approach to interval uncertainty," *International Journal of Disaster Risk Reduction*, Vol. 24, 2017, pp. 160-171.
4. D.-H. Lee, "Resource-based task allocation for multi-robot systems," *Robotics and Autonomous Systems*, Vol. 103, 2018, pp. 151-161.
5. S. Sasikala, S. A. Balamurugan, and S. Geetha, "A novel adaptive feature selector for supervised classification," *Information Processing Letters*, Vol. 117, 2017, pp. 25-34.
6. P. Shunmugapriya and S. Kanmani, "A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid)," *Swarm and Evolutionary Computation*, Vol. 36, 2017, pp. 27-36.
7. H. Lim, J. Lee, and D.-W. Kim, "Optimization approach for feature selection in multi-label classification," *Pattern Recognition Letters*, Vol. 89, 2017, pp. 25-30.
8. F. Martínez-Álvarez, G. Asencio-Cortés, J. F. Torres, *et al.*, "Coronavirus optimization algorithm: A bioinspired metaheuristic based on the COVID-19 propagation model," *Big Data*, Vol. 8, 2020, pp. 308-322.
9. H. Chen, G. Ding, J. Zhang, and S. Qin, "Research on priority rules for the stochastic resource constrained multi-project scheduling problem with new project arrival," *Computers & Industrial Engineering*, Vol. 137, 2019, p. 106060.
10. H. Dang Quoc, L. Nguyen The, C. Nguyen Doan, and T. Phan Thanh, "New cuckoo search algorithm for the resource constrained project scheduling problem," in *Proceedings of International Conference on Computing and Communication Technologies*, 2020, pp. 1-3.
11. P. B. Myszkowski, Ł. P. Olech, M. Laszczyk, and M. E. Skowroński, "Hybrid differential evolution and greedy algorithm (DEGR) for solving multi-skill resource-constrained project scheduling problem," *Applied Soft Computing*, Vol. 62, 2018, pp. 1-14.
12. P. B. Myszkowski, M. E. Skowroński, Ł. P. Olech, and K. Oślizło, "Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem," *Soft Computing*, Vol. 19, 2014, pp. 3599-3619.
13. P. B. Myszkowski, M. Laszczyk, I. Nikulin, and M. Skowroński, "iMOPSE: a library for bicriteria optimization in multi-skill resource-constrained project scheduling problem," *Soft Computing*, Vol. 23, 2018, pp. 3397-3410.
14. A. Priya and S. K. Sahana, "Multiprocessor scheduling based on evolutionary technique for solving permutation flow shop problem," *IEEE Access*, Vol. 8, 2020, pp. 53151-53161.
15. M. Kenji and G. Chowell. "Estimating risk for death from coronavirus disease, China, January-February 2020," *Emerging Infectious Diseases*, Vol. 26, 2020, pp. 1251-1256.

Annu Priya received the B.E degree in Computer Science Engineering from Bangalore Institute of Technology, India, in 2014 and the M.E. degree in Software Engineering in 2016 from Birla Institute of Technology (Ranchi, India). She completed her Ph.D. from the Computer Science and Engineering, B.I.T Mesra, Ranchi, 2021. She is currently working as a faculty in Computer Science and Engineering, NIT Srinagar. Her research interests include soft computing, image processing and high-performance computing.

Sudip Kumar Sahana received the B.E (Computer Technology) from Nagpur University in 2001, and the M.Tech. (Computer Science) in 2006 from the B.I.T (Mesra), Ranchi, India where he has done his Ph.D. (Engineering) in 2013. His major field of study is in Computer Science. He is currently working as Assistant Professor in the Department of Computer Science and Engineering, B.I.T., Mesra, Ranchi, India. His research and teaching interests include soft computing, and computational intelligence. He has authored numerous articles, research papers and books in the field of Computer Science and assigned as editorial team member and reviewer for several reputed journals. He is a lifetime member of Indian Society for Technical Education (ISTE), India and fellow of IETE, India.