



Available online at <http://scik.org>

J. Math. Comput. Sci. 11 (2021), No. 1, 893-913

<https://doi.org/10.28919/jmcs/5218>

ISSN: 1927-5307

DYNAMIC TASK SCHEDULING USING NATURE INSPIRED ALGORITHMS

SASMITA KUMARI NAYAK^{1,*}, CHANDRA SEKHAR PANDA²

¹Department of Computer Science and Engineering, Centurion University of Technology and Management, Odisha,
India

²Department of Computer Science and Application, Sambalpur University, Sambalpur, Odisha, India

Copyright © 2021 the author(s). This is an open access article distributed under the Creative Commons Attribution License; which permits unrestricted use; distribution; and reproduction in any medium; provided the original work is properly cited.

Abstract: In this article; introduces a dynamic task scheduling in distributed system. Many techniques have been proposed for dynamic task scheduling problem. In a distributed system; Nature inspired algorithms have been drawn up for scheduling the heterogeneous tasks on heterogeneous processors dynamically. It utilizes different nature inspired algorithms to minimize and maximize the makespan and average utilization of processors respectively. It deals with the dynamic task scheduling problem. This paper is demonstrated with three phases. In first phase; introduces the dynamic task scheduling problem with the computation of objectives. In the next phase; explaining about the Nature Inspired algorithms applied to this problem. All proposed Nature Inspired algorithms are introduced as a multi-objective optimization algorithm. In the last phase; the experimental results compared with the varied Nature Inspired algorithms to get the better performance in dynamic task scheduling problem. We have accomplished more effective and good outcomes by analyzing all the techniques over a varied scenario with scheduling of 52 tasks on 29 heterogeneous processors.

Keywords: dynamic task scheduling; nature inspired algorithm; genetic algorithm; bacteria foraging optimization;

* Corresponding author

E-mail address: nayaksasmita484@gmail.com

Received November 16, 2020

genetic based bacteria foraging optimization; water cycle algorithm; krill herd algorithm; symbiotic organisms search.

2010 AMS Subject Classification: 68W40.

1. INTRODUCTION

Nowadays, dynamic task scheduling is the most concerned research area due to the fact that day-by-day growing speeds in the execution of a workload. Distributed computing is a supporting method to satisfy the increasing computational needs of academic researches. We accept that task scheduling is a more significant of these issues due to improper scheduling of tasks cannot operate their capacities of a distributed system as well as can offset the gains from parallelization because of under-usage of processors or outrageous communication cost. Generally, dynamic task scheduling is an NP-hard problem [1-4], [7], [9], [22], [24]. It is the scheduling of heterogeneous tasks onto heterogeneous resources. Heterogeneous environments as well as dynamic nature of problems are the major issues for executing, analysing and designing the phases of task scheduling algorithms.

Scheduling problem is another focal errand in high-level synthesis. Scheduling systems have come out to be very complicated and most efficient within the previous couple of years. The structure of the scheduling systems largely depends on an optimum method to the scheduling model. Scheduling is utilized to create an allotment of tasks to processors for a particular amount of time to optimize the objective function. There are two cases of the scheduling are there in literature, viz. static and dynamic. Static scheduling needs background knowledge of the tasks to be scheduled to find its execution time. Dynamic task scheduling does not require any background knowledge of the tasks to be scheduled.

The word 'Optimization' implies to the study of issues during which one seems maximize or minimize the objective function, by consistently selecting the real values or whole number from among the associate allowed set.

During distributed system computation, scheduling of a group of tasks is either dependent or

independent. In dynamic environment, resource availability and load on resources is amended from session to session. Therefore, in multiprocessor system, the scheduling resources are a complex task. Particularly, this is an exigent issue in the environment of heterogeneous computing in which the capability of the resources varies [5].

A wide range of scheduling strategies is suggested by considering some factors concurrently. Such strategies are categorised into various categories, like centralized vs. distributed scheduling, static vs. dynamic scheduling, and local vs. global scheduling [6], [8], [10]. The main focus of this article is dynamic task scheduling. In this case, whole task or part of the task is operated at runtime. Most of the existing dynamic scheduling strategies are proficient; however they might not provide good schedules. The quality of such strategies has been examined widely however the immediate impact of scheduling intricacy on the total execution time and its trade-off with the complexity of the resulting schedules has not been processed.

From the beginning of the research in this area, several methodologies are produced for the result of dynamic task scheduling. From them, few are based on heuristic methodology and a few explore meta-heuristics including neighbourhood search, nature inspired as well as evolutionary methodologies. A few of them followed the hybrid techniques. Most of the meta-heuristics outperformed traditional heuristic-based algorithms at the cost of computing effort and extra time. Hence, meta-heuristics like Particle Swarm Optimization (PSO), Tabu Search (TS), Simulated Annealing (SA), Genetic Algorithm (GA) etc. have been taken by researchers to obtain high quality schedules. Here, we have discussed, analyzed and focused mainly the various Nature Inspired optimization algorithms to get the efficient algorithm for maximum processor utilization with minimum execution time like GA, Bacteria Foraging Optimization (BFO), Hybrid Genetic Based Bacteria Foraging Optimization (GBF) [2], [4], [7], [9], [22], [24], Water Cycle Algorithm (WCA) [4], [7], Krill Herd Algorithm (KHA) [9], and Symbiotic Organism Search Algorithm (SOSA) [22] and its applications in the realm of dynamic task scheduling with experimental findings.

The remainder of the article is sorted out as takes after, Sect – 2 depict the related work of

dynamic task scheduling problem. The concise presentation of dynamic task scheduling problem is done in Sect-3. The working procedure of all proposed Nature Inspired Optimization Algorithms are illustrated in Sect-4. Sect-5 discusses the simulation of all proposed algorithms. The conclusion is summarised in Sect-6. Finally the future work is projected in Sect-7.

2. RELATED WORK

There are numerous research results that help the appropriateness of Nature Inspired algorithms for minimizing the makespan in dynamic task scheduling problem. Several researchers in computer science have compared several methods in the optimization problem. A few of them are outlined here in Table 1.

Table 1 enlightens the fine points of various heuristics and Nature Inspired methods as per their development year as well as their corresponding developers or authors.

Table 1. Nature Inspired Algorithms as per development year

Algorithms	Development Year	Researchers	Objectives	Remarks
Bacterial Foraging Optimization Algorithm (BFOA) [11]	2002	Prof. K.M. Passino	High performance optimizer algorithm.	Used in various applications such as image processing, network scheduling, electric load forecast.
Genetic based Bacteria Foraging algorithms (GBF) [2]	2012	Nayak, S.K., Padhy, S.K., Panigrahi, S.P.	To minimize the total execution time of dynamic task scheduling in grid computing.	A hybrid algorithm which uses both GA and BFO algorithms.
KH algorithm [12]	2012	Gandomi, Alavi	Used in various optimization problem.	Derivative information is not necessary. Simple and easy to implement.
Genetic Algorithm[13]	1994	E.S.H.Hou; N.Ansari; Hong Ren	To minimizing the schedule length.	Robust stochastic search algorithms for various optimization problems.
Genetic Algorithm [14]	1999	Bohler, M., Moore, F. and Pan, Y.	Minimizing the schedule length for a general task graph on a multiprocessor system.	Easily adaptable to a variety of task graphs.

DYNAMIC TASK SCHEDULING USING NATURE INSPIRED ALGORITHMS

Genetic Algorithm [15]	2006	Jelodar, M.S., Fakhraie, S.N. et al.	Minimizing the total execution time.	Duplication of tasks was allowed.
Genetic Algorithm (GA) [16]	2011	Engin, O., Gulsad, C., A. Yilmaz, M.K.,	Minimizing the makespan.	Uses a new mutation operator called Neighborhood Based Mutation (NBM)
Water cycle algorithm [17]	2012	Eskandar H, Sadollah A, Bahreininejad A, Hamdi M	To optimize engineering problems and constrained functions.	Derived from nature on the basis of observation of water.
Symbiotic Organism Search Algorithm (SOSA) [18]	2014	Cheng, Prayogo	The SOSA works on the cooperative behavior and the association of organisms present in nature.	A simple and powerful meta-heuristic algorithm.
Symbiotic organism search (SOS) [27]	2015	Ruskartina Eki, Vincent F. Yu, Santosa Budi, A. A. N. Perwira Redi	To solve capacitated vehicle routing problem (CVRP) with good solution as a preliminary testing.	Solved discrete problems.

3. DYNAMIC TASK SCHEDULING

The objectives of dynamic task scheduling problem are the minimization and maximization of total execution time and average processor utilization respectively. This article takes the views of the allocation of tasks to the varied heterogeneous processors with the associated circumstance. This problem contains a group of tasks (B) and processors (A), which accomplished on diverse processors experiences distinctive execution time [2], [4], [7], [9], [22], [24]. A task can make usage of processors from its execution processor. Minimum execution times with maximum processor utilization by distribution of tasks to the processors are the main goals. This article discusses the proposed meta-heuristics methods to solve the dynamic task scheduling. A descriptive example has been demonstrated here, containing seven tasks and four processors as shown in Table 2 [2], [4], [7], [9], [22], [24]. The column and row shows the tasks and processors respectively. The pair $[A_4, B_1] = 1$ represents allocating task B_1 to processor A_4 . The pair $[A_2, B_2]$

=0 represents not allocating task B_2 to processor A_2 .

Table 2. Population representation of task assignment using Nature Inspired Optimization

	Algorithms [22]						
	B ₁	B ₂	B ₃	B ₄	B ₅	B ₆	B ₇
A ₁	1	1	0	1	0	0	1
A ₂	0	0	0	1	1	0	1
A ₃	0	0	0	0	1	1	0
A ₄	1	0	0	1	1	1	0

This problem is simulated with the help of some meta-heuristic global optimal algorithms, so called Nature Inspired Optimization algorithm. In next section, all the proposed Nature Inspired Optimization algorithms have discussed. The objective is formulated to compute the total execution time and maximizing the processor utilization. Here, the objective function is used to compute the total execution time, *Makespan* as shown in Equation (1). Equation (2) gets the fitness function that computes the goodness of the schedule [2], [4], [7], [9], [19], [22], [24].

$$Makespan = \max \left(comp_time(A_i, B_j) \right) \quad (1)$$

$$fit_fun_i = (1/Makespan) \times \max(utilization) \quad (2)$$

The computation of average utilization is done on the particular execution of the processor. Equation (3) is utilised to get the utilization of the individual processor is given by [2], [4], [7], [9], [19], [22], [24],

$$utilization(A_i) = Finish_time(A_i)/Makespan \quad (3)$$

The division of total processor utilization and no. of processors (n) is the process of evaluating the average processor utilization. Exactly when the average processor utilization is upgraded to optimum value, at that point maintain an avoidance of the processors being idle for a while. The *Objective fun*, you may find using Equation (4). It estimates the average of the *fit_fun* while allocating the tasks to the processors [2], [4], [7], [9], [19], [22], [24].

$$Objective\ fun = \min \left\{ \frac{\sum_{i=1}^n fit_fun_i}{n} \right\} \quad (4)$$

The goal is to get the minimum *Objective fun* discussed in Equation (4). The value clearly indicates the optimum schedule along with the balance in the processor utilization.

4. NATURE INSPIRED ALGORITHMS

Nature Inspired algorithms are based on inspiration of nature. These algorithms follow the process of living things and mimic the behaviours of living things to achieve effective systems in engineering discipline [26]. Nature is a chief motivation to introduce new meta-heuristic approach and therefore, the nature-inspired algorithms are established for creating systems and resolving issues [20]. These algorithms could be classified according to the inspiration from biology and natural science. To define the kind of Nature Inspired algorithms, we have considered the most frequently used term meta-heuristic algorithms. The chief classifications of the nature-inspired meta-heuristic algorithms are the Biologically-inspired algorithms. The efficacy of the bio-inspired algorithm is their substantive resources to mimic the most effective characteristics of nature. Especially, these are derived from the “selection of the fittest” in biological systems, which created by natural selection process over numerous years. Some Nature Inspired optimization algorithms are discussed below.

A. Genetic Algorithm (GA)

Holland provided the Genetic Algorithm (GA) as a heuristic algorithm on the basis of “Survival of the fittest” [21]. It was found out a suitable tool for optimization and search problem. It holds a populace of possible solutions and these solutions are called as chromosomes. The chromosome selection has done by estimating the fitness function. The working process of dynamic task scheduling using GA is shown in figure 1. It is implemented to make the comparison of its performance with other Nature Inspired optimization algorithms [2], [4], [7], [9], [22], [24].

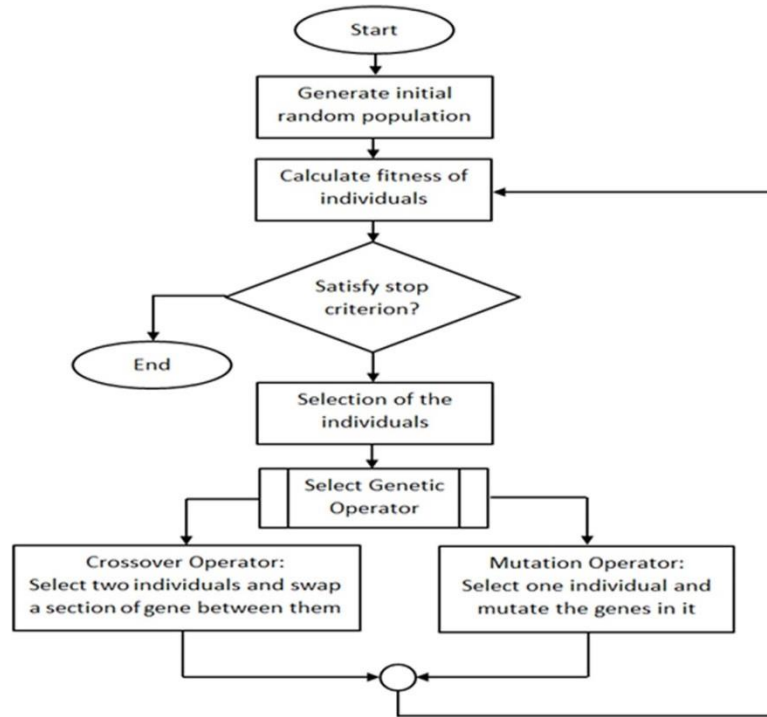


Figure 1: Dynamic Task scheduling using GA [23].

B. Bacterial Forging Optimization (BFO)

Bacterial Foraging Optimization (BFO) Algorithm [11], a nature inspired optimization algorithm, is suggested by Kevin Passino (2002). The main thrust of this algorithm is the group foraging approach of a swarm of E.coli bacteria in multi-optimal function optimization. This algorithm is introduced to produce approximate solutions to impossible or extremely difficult numerical issues [11]. It follows a probabilistic approach. The simulation process is built on the reproductive and the food seeking operation of the bacteria.

Bacteria looks for nutrients is a way to maximize energy acquired per unit time. All bacteria are communicated with each other through the signals. A bacterium makes foraging selections after consideration of two previous factors. The procedure of moving the bacteria to search the food is referred to as chemotaxis. The basic idea is imitating the chemotactic movement of virtual bacteria in the problem search space. The working process of dynamic task scheduling using BFO algorithm [2], [4], [7], [9], [22], [24], is explained with the following four steps as shown in figure 2.

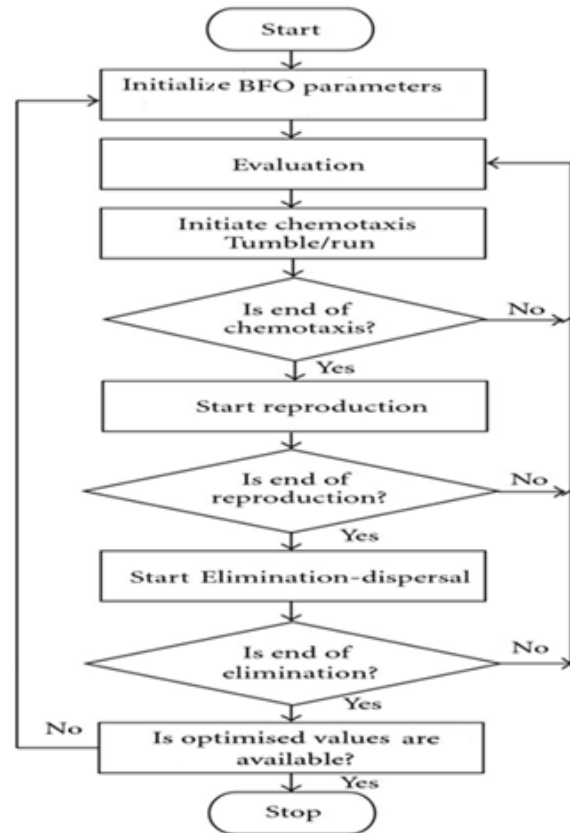


Figure 2: Dynamic Task scheduling using BFO

- i) Chemotaxis
- ii) Swarming
- iii) Reproduction and
- iv) Eliminational-Dispersal

C. Genetic Based Bacterial Foraging (GBF) Algorithm

The Genetic Based Bacteria Foraging (GBF) algorithms, is introduced in [2], [4], [7], [9], [22], [24], for scheduling the tasks dynamically. This algorithm has the benefits of both the GA and BFO algorithms. The GA can discover feasible solutions and evade premature convergence. The BFO algorithm fine-tunes in the search space and finds better solutions. In the meantime, heuristics are combined with the GA as a local search to improve the search ability [2].

D. Water Cycle Algorithm (WCA)

This is a Nature Inspired algorithm introduced in [2], [4], [7], [9], [17], [22], [24]. This algorithm is performed on the basis of how the rivers and streams flow down towards the ocean

and revert. The starting point of water is the top of mountain, which flows down in the form of rivers, streams etc. and ended in the ocean. All rivers, streams gather water from the rain and other streams on their way downhill. The water of lakes and rivers is vaporized once plants discharge water as the process of transpires. At that moment, clouds are produced once the vaporized water is transported in the atmosphere. These clouds gather in the colder atmosphere and make the rain to release the water back, which creates new streams as well as rivers. This process is called as water cycle process as shown in figure 3 [2], [4], [7], [9], [22], [24]. The steps of water cycle process are as follows:

- i) Transpiration
- ii) Evaporation
- iii) Condensation
- iv) Precipitation
- v) Percolation

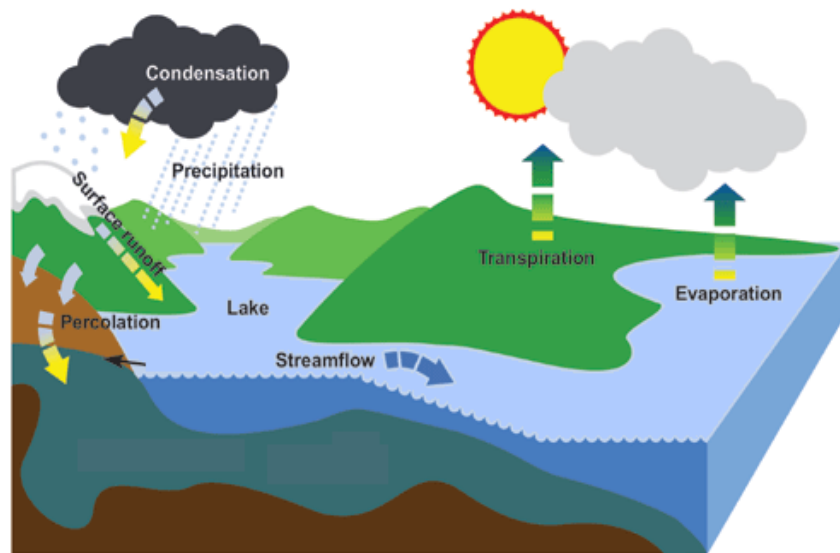


Figure 3: The Water cycle process [7], [9].

The working process of dynamic task scheduling using WCA is shown in figure 4.

E. Krill Herd Algorithm (KHA)

The Krill Herd (KH) [2], [4], [7], [9], [12], algorithm is a meta-heuristic algorithm, based on the bio-based Swarm Intelligence (SI) algorithm. This algorithm is simulated on the searching and grouping behavioural of Krill Swarms. The time-dependent position of each krill individual

is equivalent to the subsequent three stages:

- i) Movement induced by other krill individuals
- ii) Foraging Activity
- iii) Random Diffusion

For each krill individual, the objective function is the distance from the food and highest density of the swarm. This function is estimated over all above mentioned stages and the best krill. The best position is set over the iterations till the optimization criteria are reached. A number of times, this is simulated in terms of problem dimension, population size, and number of iterations per run.

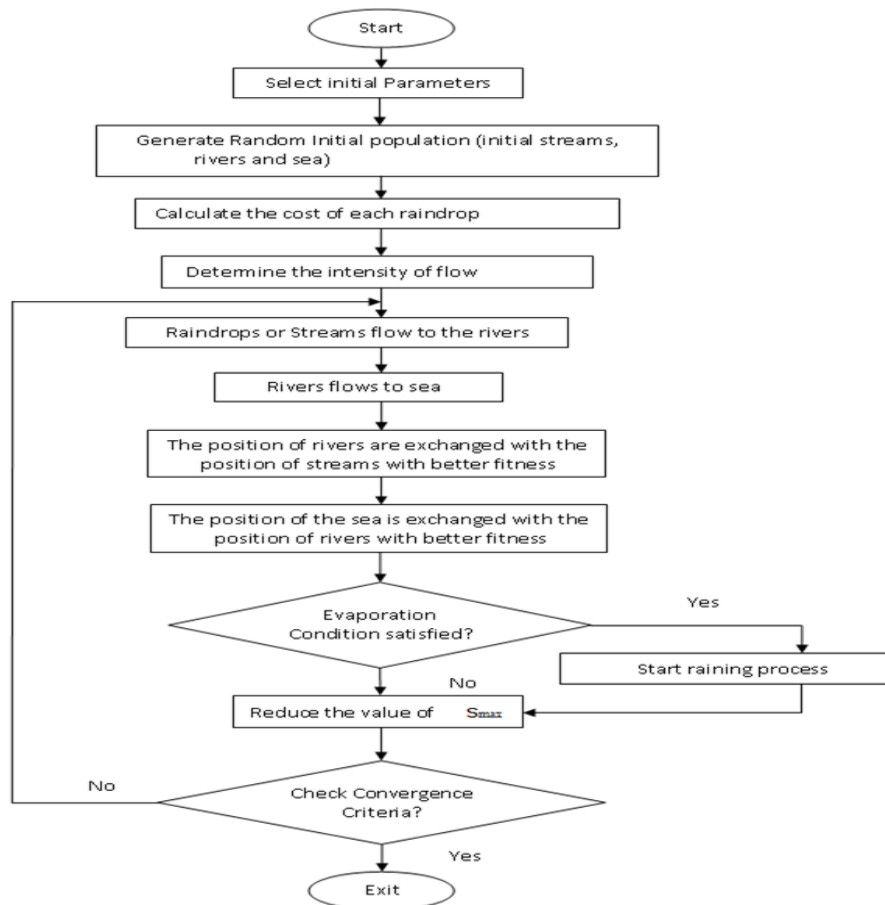


Figure 4: Dynamic Task scheduling using WCA.

The working process of dynamic task scheduling using KHA [2], [4], [7], [9], is shown in figure 5.

F. Symbiotic Organism Search Algorithm (SOSA)

Symbiotic organism search algorithm (SOSA) has highlighted in [18], [22], [26]. It is based

on the interactive behaviour by organisms for survival in an ecosystem. For their survival in ecosystem, organisms create relationships between symbioses. These relationships such as mutualism, commensalism, and parasitism are utilized for simulating the different types of symbiotic association of ecosystem. An ecosystem shows the details of each stage and the relationships between symbioses of any group of organisms as shown in figure 6.

A pair of organisms is interacted with each other for their mutual benefit but no organism is harmed from their interaction. This stage or phase is called as mutualism. A typical example is bee's interaction with flowers. Honey is produced from the flower with the collection of nectar by the bees. This collection of nectar enables the transmission of pollen grains which aid pollination. Thus, organisms are mutually benefited from their relationship.

The next phase is the commensalism phase. In this stage, from the pair of organisms, one is benefited whereas the other one is neutral i.e. neither harmed nor benefited. An interaction among sharks and remora fish is an example of commensalism. For food, Remora fish rides on shark. During that time, shark neither benefited nor harmed from their relationship.

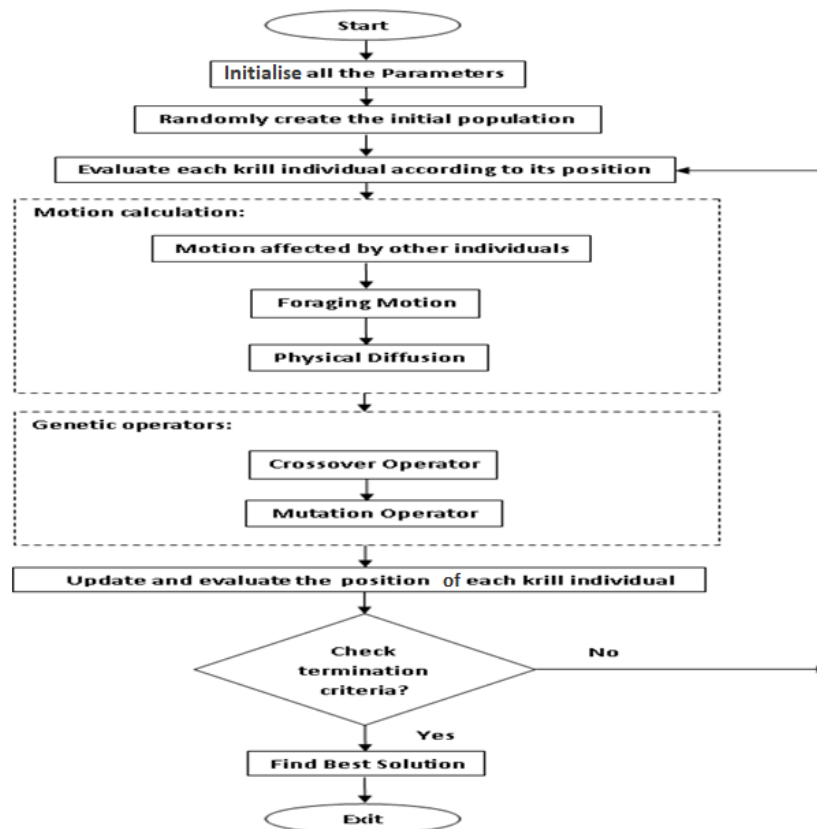


Figure 5: Dynamic Task scheduling using KHA.

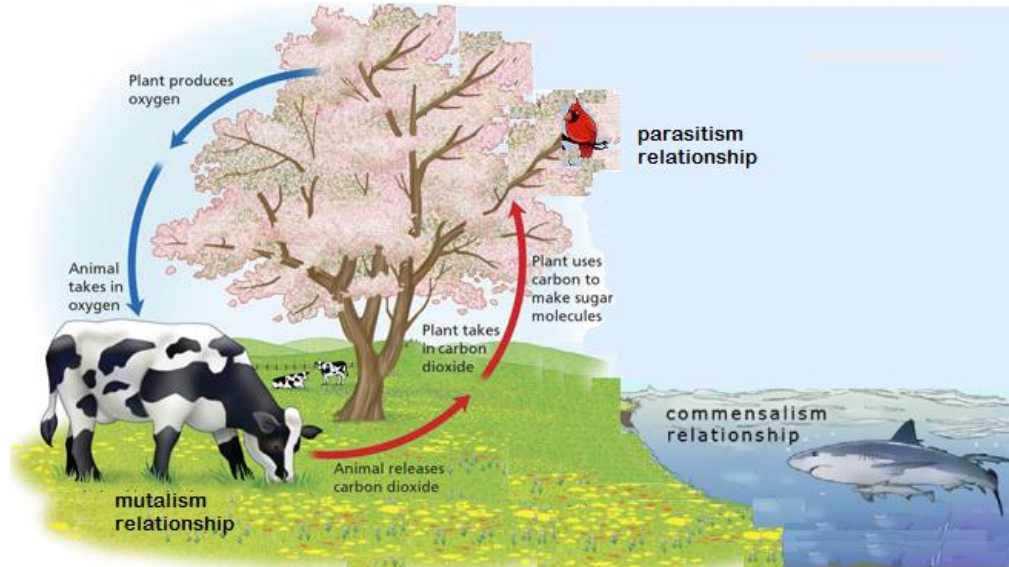


Figure 6: Symbiotic organisms live together in an ecosystem [22].

Similarly, in parasitism phase, from the pair of organisms, one is harmed whereas the other one is benefited. A typical example is human host interaction with anopheles mosquito. An anopheles mosquito transmits plasmodium parasite to human host which could cause the death of human host if his/her system cannot fight against the parasite. The working process of dynamic task scheduling using SOSA is shown in figure 7.

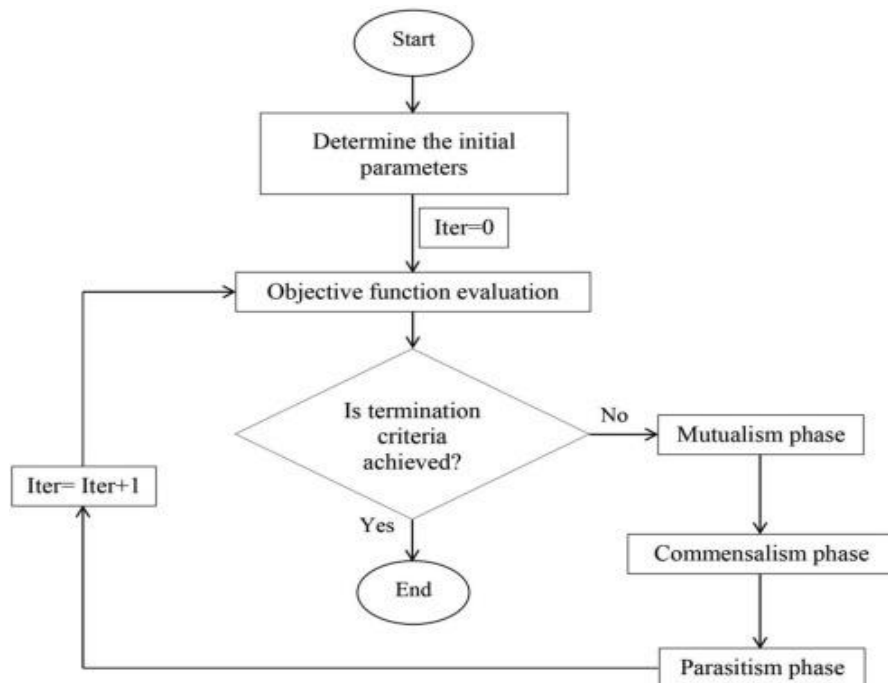


Figure 7: Dynamic Task scheduling using SOS [25].

5. SIMULATION

Through simulation, this article ascertains the impacts of tasks and processors on execution time of computation. The above six mentioned Nature Inspired algorithms are improved by their simulation results. Comparisons were examined the performance of all algorithms, which is clearly represented with the help of tabular and graphical representations of results.

Here, the proposed algorithms are simulated using MATLAB R2014a with different number of processors and tasks with the purpose of maximizing and minimizing the processor utilization and execution time respectively. Because high execution time of tasks tends to lower the percentage of the processor utilization and vice versa.

For simulation, the parameters of all proposed Nature Inspired optimization algorithms are shown in table 3.

Consider an illustration of Table 2, the number of processor is 4 and the number of task is 7. Pick randomly number of organisms and assigned in an array of all nature inspired algorithms, with the assumed number of tasks and processors. The value will be 0, if a task is not allotted to processor else 1.

Table 3. Description of parameters used in Nature Inspired Algorithms

GA	BFO	GBF	WCA	KHA	SOSA
No. of iterations : 100	No. of iterations : 100	No. of iterations : 100	No. of iterations, Npop: 100	No. of iterations: 100	No. of iterations : 100
Population size: No. of Chromosome	Population size: No. of Bacteria	Population size: No. of Bacteria as chromosome	Population size: No. of raindrops	Population size: No. of Krill Individuals	Population size: No. of Symbiosis organisms or ecosize
Selection operator : Roulette wheel selection	No. of Dimensions: 2	Number of Dimensions: 2	Number of variables, nvars=10;	Maximum induced speed, Nmax: 0.01	Maximum number of function evaluation, maxFE : 5000
Cross-over operator : Two point cross-over	Number of chemotactic steps, Nc=100	Number of chemotactic steps, Nc=100		Maximum diffusion speed, Dmax:0.005	

DYNAMIC TASK SCHEDULING USING NATURE INSPIRED ALGORITHMS

Cross-over probability: 0.8	Number of reproduction steps, Nre=4	Number of reproduction steps, Nre=4		Foraging speed, Vf:0.02	
Mutation probability : 0.03	eliminated/dispersed Probability: 0.25	eliminated/dispersed Probability= 0.25		alpha_i_best=0.01;	
		Selection operator : Roulet wheel selection, Cross-over operator : Two point cross-over, Cross-over probability : 0.8, Mutation probability : 0.03		No.of Variables, Nvar = 10;	

Randomly initialize the organism or population for all Nature Inspired algorithms and repeat the process for either No. of iterations or Maximum fitness evaluation times. In each iteration, find the best (smallest value of fitness function) fitness value. For every cycle, each algorithm has the capacity to get the global minima.

All functions i.e. objective functions, utilization and fitness function are continued with a maximum number of function evaluations or 100 number of iteration. Following 3 distinct cases are considered here. Such as:

Case 1: Comparing the execution time of all proposed Nature inspired Algorithms.

Case 2: Comparing the Processor Utilization of all proposed Nature inspired Algorithms.

Case 3: Execution time vs. Average processor utilization.

For all the above cases are explained in next subsections with different number of processor and task values in increasing order. Both average processor utilization and total execution time (makespan) and is computed for the following different number of processors and tasks.

No. of Processor = 20, 22, 24, 29

No. of Task = 40, 44, 47, 52

No. of Iteration = 100

A. Comparing the execution time of all proposed Nature inspired Algorithms

In this case, we have computed the minimum execution time (Makespan) i.e. principal objective of dynamic task scheduling and KHA gives minimum makespan as compared with others such as GA, BFO, GBF, WCA and SOSA as shown in Table 4.

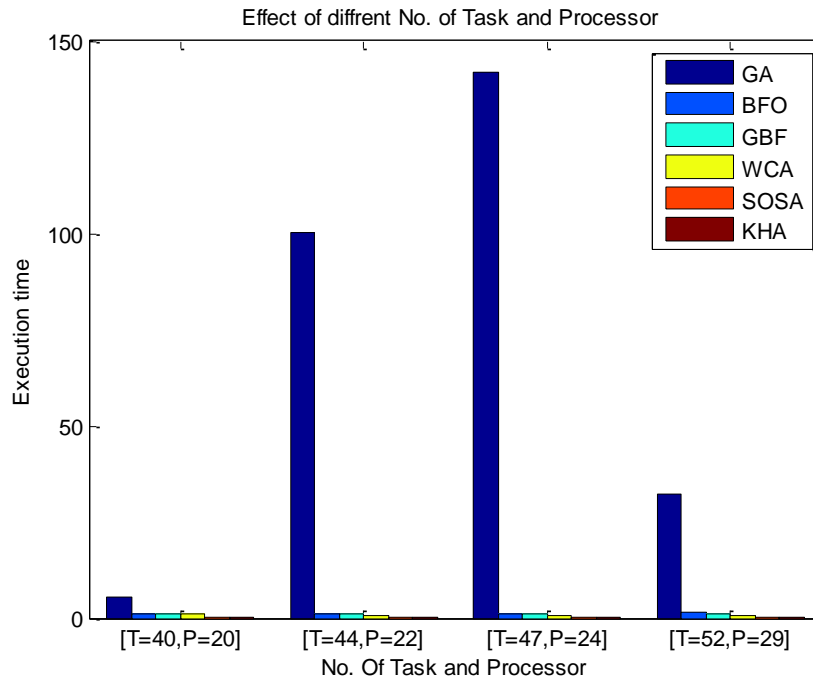


Figure 8: Performance of Execution Time using proposed Nature inspired Algorithms

By comparing all Nature Inspired algorithms, KHA provides better result for makespan with different no. of processors and tasks is shown in Figure 8. The bold value represents the best values as shown in Table 4.

Table 4. Comparison of Execution Time with different number of tasks and processors

No. of Task	No. of Processor	GA	BFO	GBF	WCA	SOSA	KHA
40	20	5.3878	1.0566	1.0089	1.0555	0.4176	0.0060
44	22	100.1667	1.2576	1.0055	0.5366	0.2751	0.0052
47	24	141.8000	1.0000	1.0013	0.6101	0.1585	0.0032
52	29	32.1111	1.5495	1.0015	0.5976	0.0260	0.0028

B. Comparing the Processor Utilization of all proposed Nature inspired Algorithms

In this case the processor utilization of dynamic task scheduling using Nature Inspired Algorithm is compared among with each other, such as, GA, BFO, GBF, WCA, SOSA and KHA as shown in Figure 9. Here, from the graphical visualization, we found that GBF has not utilizing the processor well as compared with the algorithms like GA, BFO, WCA, SOSA and KHA. However, KHA provides better result for average processor utilization by comparing with the other implemented algorithms like GA, BFO, GBF, WCA and SOSA as shown in Table 5. The bold values are represented as steadily increasing the utilization of processor with respect to the increasing the number of task and processors.

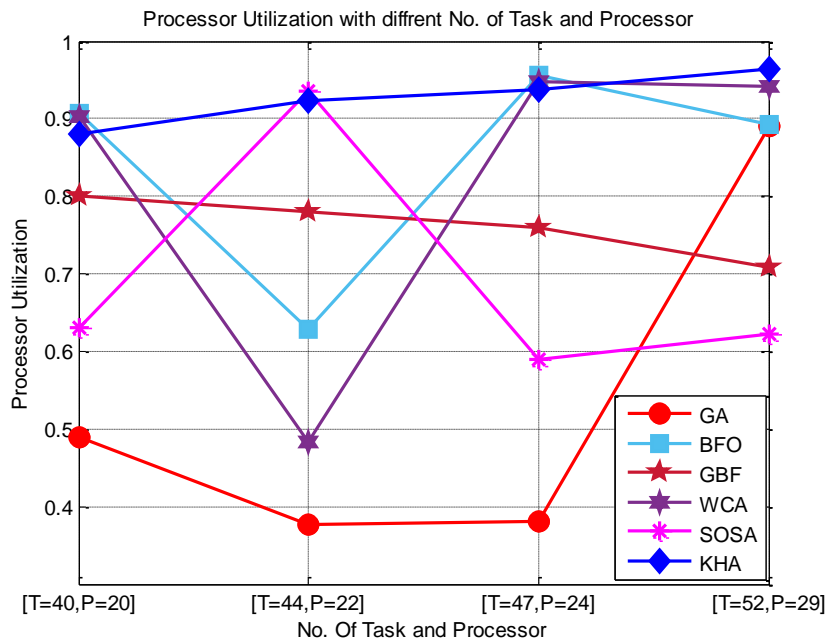


Figure 9: Performance of utilization of processors using proposed Nature inspired Algorithms

Table 5. Performance of Average Processor Utilization in percentage with 100 Iterations

No. of Task	No. of Processor	GA	BFO	GBF	WCA	SOSA	KHA
40	20	0.4888	0.9072	0.7998	0.9027	0.6299	0.8794
44	22	0.3770	0.6277	0.7796	0.4823	0.9358	0.9241
47	24	0.3806	0.9563	0.7596	0.9469	0.5895	0.9373
52	29	0.8912	0.8932	0.7082	0.9407	0.6234	0.9642

C. Execution time vs. Average processor utilization

Next we compare the makespan and average processor utilization as shown in Figure 10.

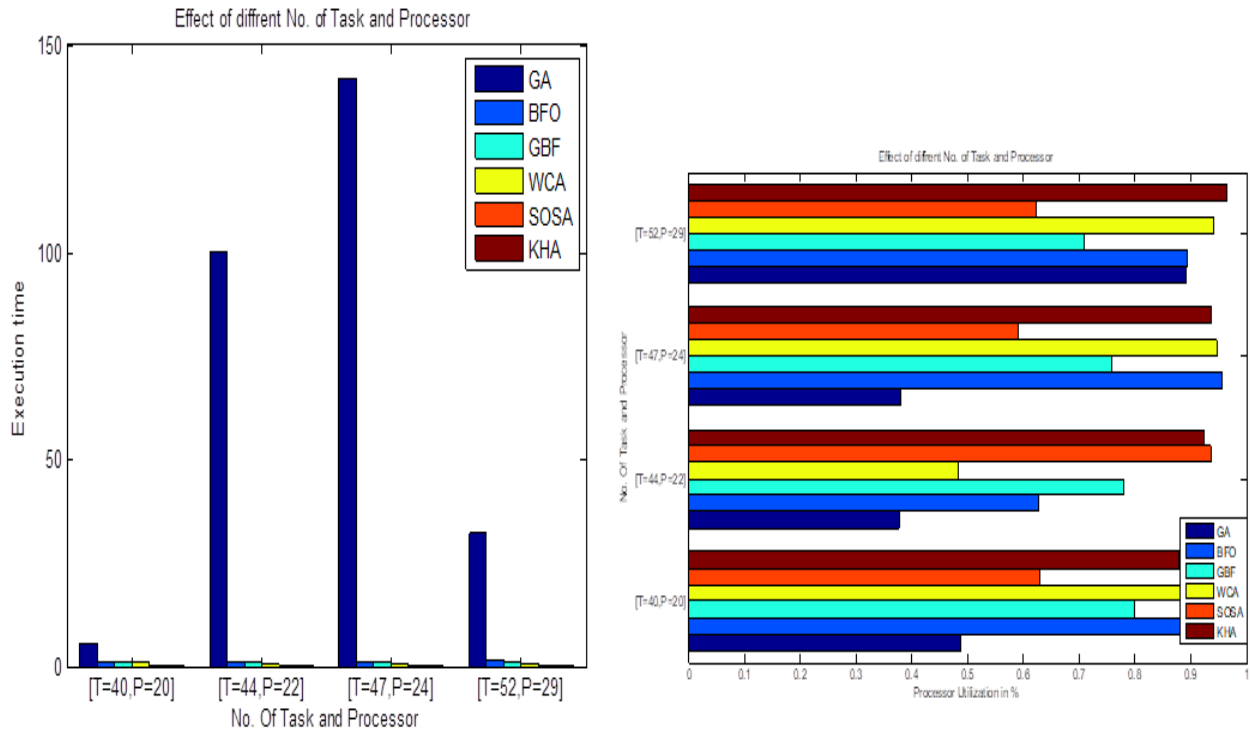


Figure 10: Execution time vs. Processor Utilization by using GA, BFO, GBF, WCA, SOSA and KHA

Here, the makespan and average processor utilization of all nature inspired optimization algorithm are compared with each other, such as, GA, BFO, GBF, WCA, SOSA and KHA. In both the cases, the results show that KHA algorithm performs better than other algorithms for all the different number of tasks and processors.

6. CONCLUSION

In this article, the proposed algorithm KHA is utilized for allocating different task to different processor in the dynamic task scheduling problems. Here, all nature inspired optimization methodologies have been implemented successfully to find the optimum values of dynamic task scheduling problem. The effectiveness of the proposed method is demonstrated on the test systems considered. From the simulation results i.e. after the graphical and experimental

consequences it can be concluded that the recommended KHA performed well for finding the optimum value of makespan, than the experimental results of other algorithms such as GA, BFO, GBF, WCA, and SOSA. In other words, the comparison results have shown that the KHA algorithm outperform than the existing algorithms.

7. FUTURE WORK

The future direction for our work is the implementation of new hybrid algorithm to solve the dynamic task scheduling problem with cloud environment.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

- [1] A.J. Page, T.J. Naughton, Dynamic Task Scheduling using Genetic Algorithms for Heterogeneous Distributed Computing, in: 19th IEEE International Parallel and Distributed Processing Symposium, IEEE, Denver, CO, USA, 2005.
- [2] S.K. Nayak, S.K. Padhy, S.P. Panigrahi, A novel algorithm for dynamic task scheduling, *Future Gener. Comput. Syst.* 28 (2012), 709–717.
- [3] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman and Company, San Francisco, (1979).
- [4] S.K. Nayak, S.K. Padhy, C.S. Panda, Efficient Dynamic task scheduling Using Water Cycle Algorithm. In: Pant M., Ray K., Sharma T., Rawat S., Bandyopadhyay A. (eds) *Soft Computing: Theories and Applications. Advances in Intelligent Systems and Computing*, vol 583. Springer, Singapore, (2018).
- [5] M. Maheswaran, S. Ali, H. J. Siegel, D. Hensgen, R. F. Freund, Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems, *J. Parallel Distrib. Comput.* 59 (1999), 107-131.
- [6] D.J. Lilja, Lau Ying Kit, B. Hamidzadeh, Dynamic task scheduling using online optimization, *IEEE Trans. Parallel Distrib. Syst.* 11 (2000), 1151–1163.

- [7] S.K. Nayak, C.S. Panda, S.K. Padhy, Efficient Dynamic task scheduling Using Water Cycle Algorithm. In: Ray K., Pant M., Bandyopadhyay A. (eds) Soft Computing Applications. Studies in Computational Intelligence, vol 761. Springer, Singapore. (2018).
- [8] T.L. Casavant, J.G. Kuhl, A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems, IEEE Trans. Softw. Eng. 14(2) (1988), 141-154.
- [9] S.K. Nayak, C.S. Panda, S.K. Padhy, Dynamic task scheduling using Krill Herd Algorithm (KHA). Int. J. Computer Sci. Eng. 6(6) (2018), 7-17.
- [10] M.G. Norman, P. Thanisch, Models of Machines and Computation for Mapping in Multicomputers, ACM Comput. Surv. 25(3) (1993), 263-302.
- [11] K.M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Syst. 22(3) (2002), 52-67.
- [12] A.H. Gandomi, A.H. Alavi, Krill herd: A new bio-inspired optimization algorithm, Commun. Nonlinear Sci. Numer. Simul. 17 (2012), 4831-4845.
- [13] E. Hou, N. Ansari, H. Ren. A genetic algorithm for dynamic task scheduling, IEEE Trans. Parallel Distrib. Syst. 5(2) (1994), 113–120.
- [14] M. Bohler, F. Moore, Y. Pan, Improved Multiprocessor Task Scheduling Using Genetic Algorithms, Proceedings of Twelfth International Artificial Intelligence Research Society Conference, Orlando, Florida, USA, 140-146, 1999.
- [15] M.S. Jelodar, S.N. Fakhraie, F. Montazeri, S.M. Fakhraie, M.N. Ahmadabadi, A Representation for Genetic-Algorithm-Based Multiprocessor Task Scheduling, in: 2006 IEEE International Conference on Evolutionary Computation, IEEE, Vancouver, BC, Canada, 2006: pp. 340–347.
- [16] O. Engin, G. Ceran, M.K. Yilmaz, An efficient genetic algorithm for hybrid flow shop scheduling with multiprocessor task problems, Appl. Soft Comput. 11 (2011), 3056–3065.
- [17] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110–111 (2012), 151–166.
- [18] M.-Y. Cheng, D. Prayogo, Symbiotic Organisms Search: A new metaheuristic optimization algorithm, Comput.

Struct. 139 (2014), 98–112.

- [19] S.N. Sivanandam, P. Visalakshi, Dynamic task scheduling with load balancing using parallel orthogonal particle swarm optimization, *Int. J. Bio-Inspired Comput.* 1(4) (2009), 276-286.
- [20] H. Zang, S. Zhang, K. Hapeshi, A Review of Nature-Inspired Algorithms, *J. Bionic. Eng.* 7 (2010), S232–S237.
- [21] J.H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA, USA, 1975.
- [22] S.K. Nayak, C.S. Panda, Dynamic task scheduling with Optimum Execution Time and Processor Utilization Based on the SOSA. *Int. J. Recent Technol. Eng.* 8(2) (2019), 5463-5471.
- [23] M.N. Ab Wahab, S. Nefti-Meziani, A. Atyabi, A Comprehensive Review of Swarm Optimization Algorithms, *PLoS ONE.* 10 (2015), e0122827.
- [24] S.K. Nayak, C.S. Panda, S.K. Padhy, Dynamic Task Scheduling Problem Based on Grey Wolf Optimization Algorithm, in: *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, IEEE, Gangtok, India, 2019: pp. 1–5.
- [25] S. Akbarifard, F. Radmanesh, Predicting sea wave height using Symbiotic Organisms Search (SOS) algorithm. *Ocean Eng.* 167 (2018), 348-356.
- [26] M. Abdullahi, M.A. Ngadi, S.I. Dishing, S.M. Abdulhamid, M.J. Usman, A survey of symbiotic organisms search algorithms and applications, *Neural Comput. Appl.* 32 (2020), 547–566.
- [27] R. Eki, V.F. Yu, S. Budi, A.A.N.P. Redi, Symbiotic Organism Search (SOS) for Solving the Capacitated Vehicle Routing Problem, *Int. J. Ind. Manuf. Eng.* 9 (2015), 873-877.