A novel algorithm for real time task scheduling in multiprocessor environment

Joel Josephson & R. Ramesh

Cluster Computing
The Journal of Networks, Software Tools and Applications

ISSN 1386-7857

Cluster Comput
DOI 10.1007/s10586-018-2083-5

Cluster Computing
https://doi.org/10.1007/s10586-018-2083-5

A novel algorithm for real time task scheduling in multiprocessor environment

Joel Josephson[1] · R. Ramesh[1]

Received: 9 November 2017 / Revised: 19 January 2018 / Accepted: 7 February 2018

Abstract
The objective of the study is to establish task scheduling process by examining the various real times scheduling algorithm. Subsequently, the research attempted to propose a new algorithm for task scheduling in a multiprocessor environment. In addition, the study planned to implement the new algorithm for the security issues, hardware and software implementation. For developing real-time scheduling, TORSCHE toolbox is used. A novel algorithm was developed using features of particle swarm optimization, Cuckoo search, and fuzzy concepts. The findings showed that the proposed algorithm executes a maximum number of the process at a minimum time.

Keywords Real-time scheduling · Task scheduling · PSO · Cuckoo search · Fuzzy

1 Introduction

MTS or multiprocessor task scheduling refers to the scheduling of task scheduling within a multiprocessor environment. This is widely used in an array of computer applications right from scientific issues to engineering related ones. A single task is further divided into many small tasks during a parallel computation and is later assigned towards multiprocessor environment to execute simultaneously which would help in quick task execution and completion. The major aim of such task scheduling is to lower the overall length of scheduling of application program which is given. Tasks and the link between them are generally represented using the weighted DAG or directed acyclic graph for the task scheduling purpose. Task scheduling is considered to be an NP-complete issue in terms of few special cases. Task scheduling generally contains three principle components [1]: (i) Performance of Processor; (ii) Tasks mapping with the processors; and (iii) Task's order execution. There are two types of tasks scheduling such as (i) Dynamic tasks scheduling, and (ii) Static task scheduling [2]. During the static task scheduling algorithm, the execution time of the tasks the data about the communication time, and precedence constraint are revealed during the compilation time. Hence, this is also known as a deterministic scheduling algorithm. And during the dynamic task scheduling algorithm, on the other hand, the data about the tasks are revealed only at the time of execution. Hence, this is also known as non-deterministic algorithm [3].

2 Literature review

There are few authors who have conducted their research towards scheduling algorithms they are as follows. The study conducted by Kaur and Kaur [4], showed that the study about different scheduling algorithms and parallel scheduling strategies was performed. The scheduling algorithms are studied in a detailed fashion. It offers better efficiency and lower task completion time when compared to the rest of scheduling techniques. This study also offers a complete list that contains performance metrics that states which algorithms is ideal for comparison. Multiprocessor task scheduling is applicable to an array of issues where there are numerous tasks which can be performed on various processors. The tasks that this study considers are actually non-preemptive. The multiprocessor scheduling is explainable using DAG model. The scheduling that is task

✉ Joel Josephson

[1] Electrical and Electronics Engineering, College of Engineering Guindy, Anna University, Chennai 600025, India

Published online: 03 March 2018

duplication based offers better efficiency and has lesser completion time when compared to scheduling techniques. For performance and its improvement, various researchers have used many heuristics. It is further extendable to dynamic heterogeneous systems for those applications held real-time.

Gujarati et al. [5] presented the first analysis of multiprocessor scheduling in this study using the arbitrary processor affinities in the perspective of real time. It also showed that the fixed priority scheduling at the job-level using the arbitrary processor affinities is generalized rather than globalized, clustered, fixed priority scheduling at the job-level which are combined.

Rajak and Dixit [6] presented primary objective of list task scheduling algorithms was to bring down the overall execution time. DAG or directed acyclic graph represents the task scheduling in a multiprocessor environment.

According to Boveiri [7], the aim is to lower finishing time by mapping the tasks towards the elements of the processor in a fashion which preserves the precedence constraints. This issue is said to be NP tough in a general way and few ones in a restricted way. So there is the need to use both heuristic and meta-heuristic approaches for solving the issue in a logical fashion. LA or Learning automata are the abstract models for interacting with the stochastic environment; that reforms on its own with the environment feedback. Even though learning automaton is said to be a simple component, a group of it might exhibit complicated behavior even if cooperating with each other, and can hinder the solutions related to learning algorithm. In this study, the representation of an ingenious graph-like learning shows that learning automaton would help in solving multiprocessor task scheduling issue in a collective fashion. Many different experiment sets based on real-world task graphs are performed, and archived results look highly promising when compared with the older methods and genetic algorithm. In LA-MTS, the proposed approach every task in the task graph has the representation of learning automaton. Hence, the graph that depicts learning automata is generated based on the task that is given. The paper also shows that the latest BNP or bounded number of processors approach for scheduling multiprocessor task-graph on the learning automata graph was further introduced. According to LA-MTS, the proposed approach of every task in the task graph has the representation of learning automation. The first four graphs made an evaluation of LA-MTS in comparison with the traditional BNP heuristics. In a single graph, LA-MTS delivers the best performance, and it was equal to GA in the 2nd graph.

Sharma and Kaur [8], in order to solve NP-hard problem with the help of earlier strategies that follows reasonable time period. With time, many heuristic procedures presented and comprehended it. Hence, heuristic methods like genetic algorithms were considered to be very appropriate methods to task schedule inside the multiprocessor system. This study presented GA to represent static task scheduling within multiprocessor systems that were presented based on the tasks' priority and its execution was based on the task height in the graphs and the rest of parameters and later scheduling was also performed. The proposed method was also simulated and later was compared to basic genetic algorithm. GA generations number and scheduled time period were considered as directly proportional through the calculations up to computation of 1000 generations. Our observation also suggested that the number of tasks allotted plus the speed up revealed diverse result and didn't follow better assumptions and the values obtained with speed up ranged from 2.5 to 4.5. The experimental simulations were applied to an algorithm with different task graphs and generations number and by performing a comparison of Basic GA, which reveals that proposed algorithm has low schedule time than the basic GA. So, the proposed algorithm displays better behavior and is efficient enough to perform as a better scheduling system in various applications.

From the literature review it is found that there are few authors who have conducted their studies based upon the scheduling such as Kaur and Kaur [4], showed that the study about different scheduling algorithms and parallel scheduling strategies, however, the study has failed to extend to dynamic heterogeneous systems for those applications held real-time. Gujarati et al. [5]. presented the first analysis of multiprocessor scheduling in this study using the arbitrary processor affinities in the perspective of real time,however; the study didn't concentrate on the dominance of APA-based JLFP scheduling and partitioned JLFP scheduling. Rajak and Dixit [6] presented primary objective of list task scheduling algorithms was to bring down the overall execution time. However, they fail to implement their study in list scheduling. Sharma and Kaur [8] aimed to solve NP-Hard Problem with the help of earlier strategies that follows reasonable time period. However, the entire above-discussed algorithm fails to implement areal-time system. Hence in the present study, we aim to develop an algorithm in order to control the water tank level.

3 TORSCHE tool box

TORSCHE (Time Optimization of Resources, Scheduling) Scheduling Toolbox for Matlab is a freely (GNU GPL) available toolbox developed at the Czech Technical University in Prague. The toolbox is designed for researches in operational research or industrial engineering and for undergraduate courses. The current version of the toolbox covers the following areas: scheduling on monoprocessor/dedicated processors/parallel processors, open shop/flow shop/job shop scheduling, cyclic scheduling, and real-time scheduling. TORSCHE Scheduling Toolbox is written in the Matlab object-oriented programming language (backward compatible with Matlab environment version 2007) [9], and it is used in the Matlab environment as a toolbox. The toolbox includes two complementary parts. The first one is intended for solving problems from scheduling theory. Problems from this area or their parts can, very often, be reformulated to another problem which can be directly solved by a graph algorithm. For this purpose, the second part of the toolbox is dedicated to graph theory algorithms. (Fig. 1)

4 Real-time scheduling

When the system surpasses the specific time frame for producing better results than it would because undesirable results or fatal result, thereby the system failure will be considered as occurred. The task that is performed Real time would occur while few events are triggered and get repeated in random fashion repeatedly while holding some timing constraints. Such timing constraints would be taken into the account during the scheduling of task real time as shown in Fig. 2.

Even though there are n (n > 2) processors, the study author focuses only on three processors and eight unique tasks. Every task has distinct processing time as well as releasing time. LPT or longest processing time first and list scheduling algorithm that is performed with high processing time to very low processing time. The Gantt chart given above portrays that the processor 1 finished the t4 task with the longest processing time first. Similarly, processor 2 and processor 3 finish t2 and t5 that also take the longest processing time first. In addition, processor 3 also completed task t1, which takes smallest processing time. Lastly, processor 1 completed task 3, after tasks t4 and t5 was completed in Fig. 3.

SPT is used widely used in interior gateway routing protocols such as OSPF and IS–IS involves the computation of a shortest path tree (SPT). In commercial routers, the calculation of an SPT is done from scratch following changes in the link states of the network. As there may coexist multiple SPTs in a network with a set of given link states, such recompilation of an entire SPT not only is inefficient but also causes frequent unnecessary changes in the topology of an existing SPT and creates routing instability. This involves the measurement of the resistance to penetration of a sampling spoon under dynamic loading. The resistance is empirically correlated with some of the engineering properties of soil such as density index, consistency, angle of internal friction, bearing capacity etc. It becomes useful for general exploration of erratic soil profile for finding depth to bed rock or hard stratum and to have an approximate indication of the strength and other properties of soil, particularly the cohesion less soil, from which it is difficult to obtain undisturbed samples.

The figure given above is created with the help of list scheduling algorithm with SPT or shortest processing time. The Gantt chart showed that it was processor 1 that completed all the three tasks t5, t2 and t1 and processor 2 finished t8, t6 and t2 tasks while the processor 3 finished t4 and t3 tasks. However, task 1 alone was not finished by any processor is displayed in Fig. 3.

The figure given above was generated with the help of list scheduling algorithm that has ECT or Earliest Completion Time first. The Gantt chart also showed that processor 1 finished tasks t4 and t1 and processor 2 finished t5,

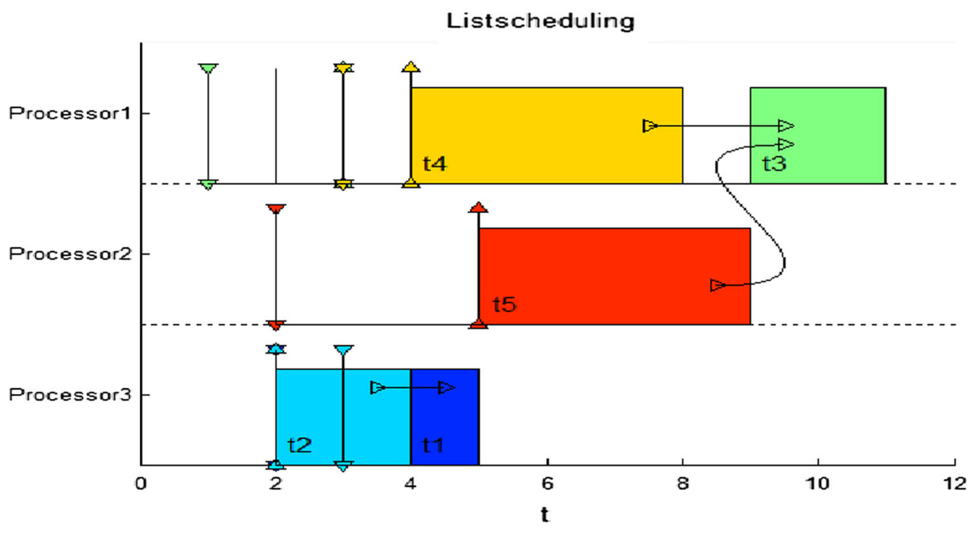


Fig. 1 LPT scheduling

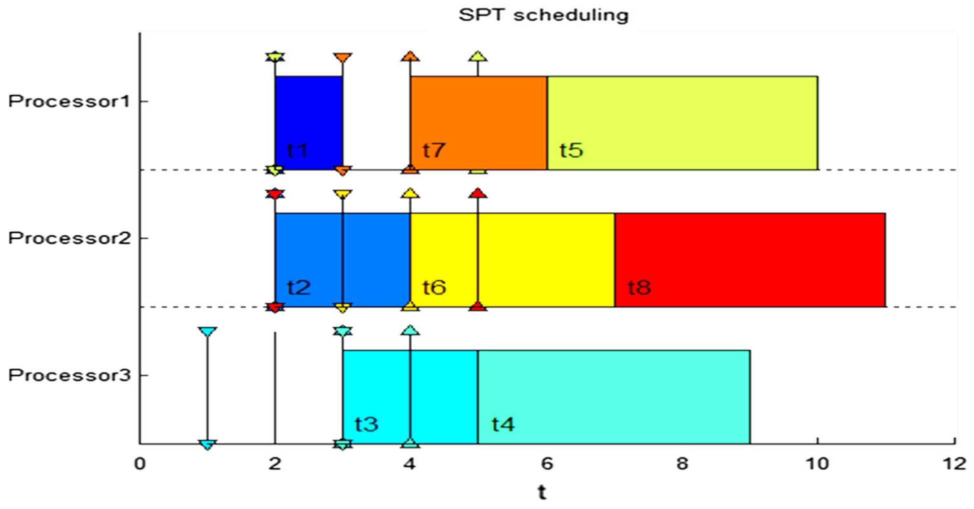


Fig. 2 SPT scheduling

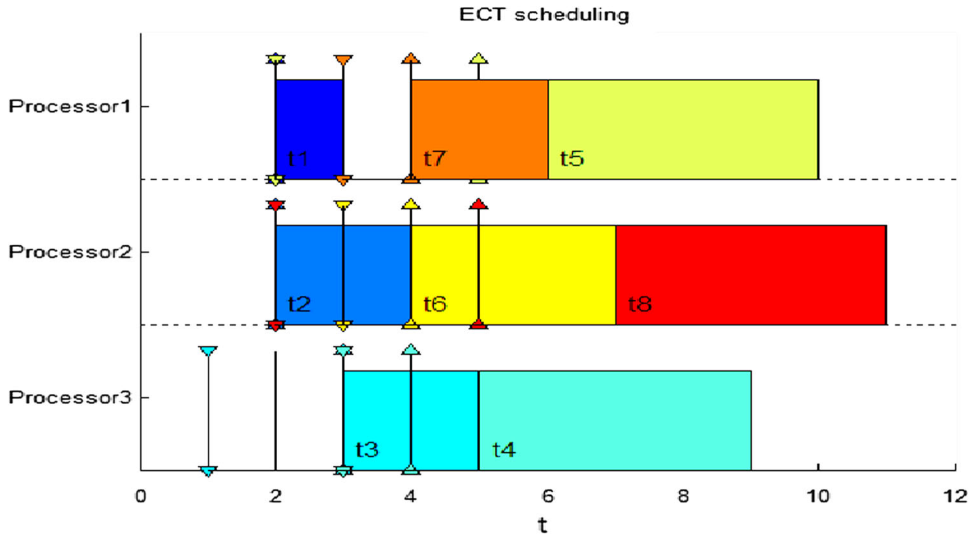


Fig. 3 ECT scheduling

t6 and t2 tasks while the processor 3 finished t7, t8 and t3 tasks in Fig. 4.

Many ECT methods arc applied newly in numerous industrial and medical processes. ECT is discovered in research from many parts for example how to rebuild accurate imageries of the thing under consideration, implantation of real-world recognition schemes and/or image viewing strategies. It has non-linear features and the number of the capacitances dimensions are a lesser compared to number of pixels for image to be renovated. So, the inverse problem is an under determined problem. Moreover, the sensitivity field of ECT is contained as not frequently distributed, and the reverse problem equation is in a seriously abnormal state. Therefore, image reconstruction algorithm is considered as a bottleneck for the development of an ECT, and a real time and high precise image reconstruction algorithm is required.

The Gantt chart given in Fig. 5 portrays that the 5 different tasks which are t5, t4, t3, t2, and t1 have the unit release time. Hu's algorithm is suitable for scheduling unit release time tasks using precedence constraints (in the tree). The tasks were completed one after one, on the basis of in-tree notation levels.

For EST sequences, some hundred clear bases are formed as of each sequencing read, and yet a full gene transcript may be several thousands of bases long. In widely obtainable records, EST length differs as of less than 20 to over 7000 base pairs, with an average length of 360 base pairs and standard deviation of 120 base pairs. Perceptibly, not all of these sequences are true single-read tags, but they are submitted and accepted as such, bringing extra complications to EST analysis. There is significant diversity in EST generation methods. Important thing is with random primers, and it gives in production of fragments deprived of direction, creating from different non-

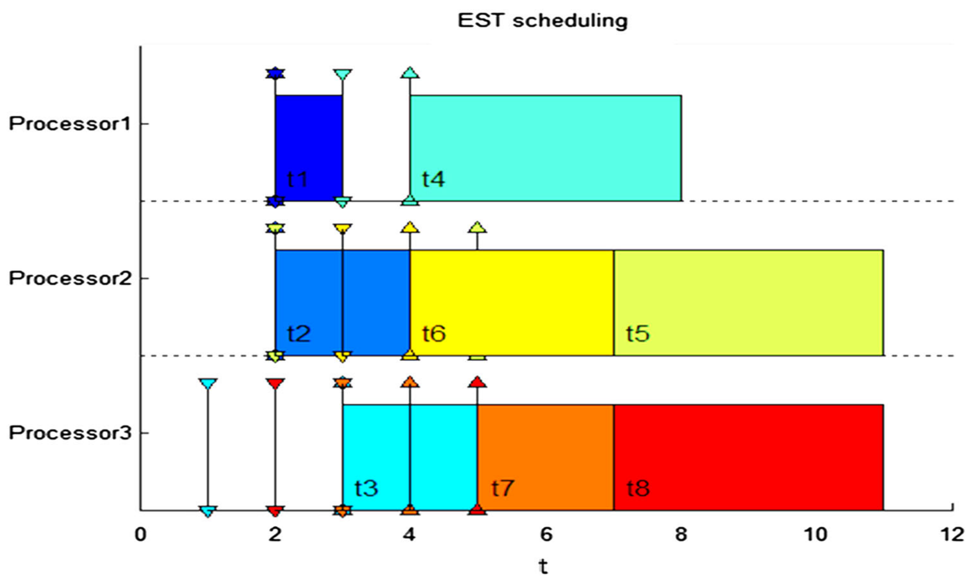


Fig. 4 EST scheduling

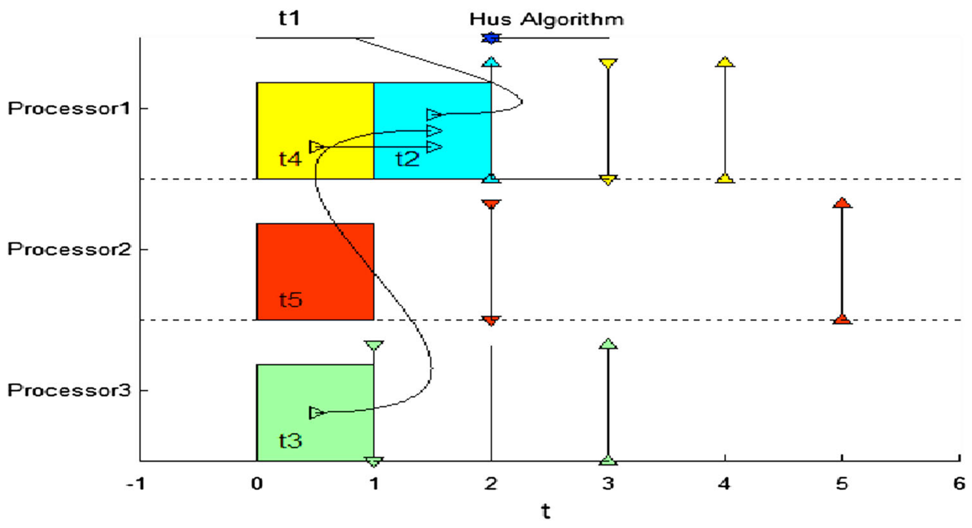


Fig. 5 Hus algorithm results

overlapping parts of the same mRNA. ESTs provide a "tag level" suggestion with an expressed gene sequence, trading quality and total sequence length for the high quantity of samples.

Based on the figure above, processor 1 finished both t4 and t2, which are the unit release time tasks. Also, processor 3 and 2 finished the t5 task and t3 task, the unit release time ones respectively. In the meantime, tasks t3, t4, and t5 were also completed.

Hu's algorithm permits to produce a task implementation schedule resultant in the least execution period for a given number of computers. In the generation of the optimal schedule of task execution, initial input data is a graph showing the sequence and the relationship of these tasks.

The arrow near the task t2 shows it was under process, after task t3 and t4′s completion. Also, it was tough to do task after completing task t2. Figure 6 shows the flow diagram of this process.

5 Existing and proposed algorithm

5.1 PSO

Adaptation of particle swarm has proved to be successful for optimizing an array of functions that are continuous [10, 11]. The algorithm that has the metaphor as the basis to facilitate social interaction, which makes a search on space by making adjustments to the trajectories of individual vectors, referred to "particles" as they are considered as the points that move within the multidimensional space [12–14]. The plot for PSO algorithm for water tank control is shown in Fig. 7.

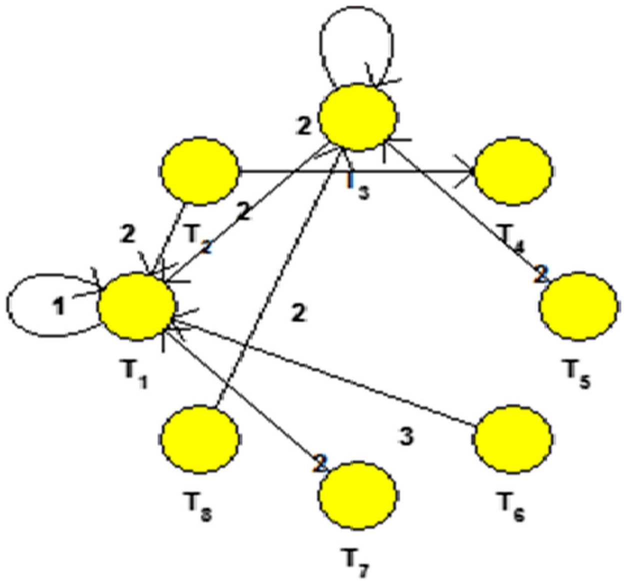


Fig. 6 Graphical representation

The diagram exhibited above shows the particle's velocity with many iterations. During the first iteration, the particle's velocity was too high up to 2.9. Then in the second and third iteration, particle's velocity was low by 2.5 and 2.0 respectively. The particle's velocities were gradually low when there is a rise in a number of iterations. The velocity was zero soon after the sixth iteration.

5.2 Cuckoo search algorithm

CS algorithm has the cuckoo breed's parasitic behavior as its basis that teams up with some fruit flies' and birds' follow Levy flight behavior. Few breeds of Cuckoo birds often lay the eggs inside the communal nests. When the host bird finds out that the eggs do not belong to it, the bird either tries to throw away the eggs or leave sits nest immediately to build a new one somewhere [15]. Fig. 8 shows the relation between cost and cuckoo iteration based on cuckoo search algorithm.

Figure 8 is given also ndicates that there prevails link between the duration of time and the tank's water level. When the time duration is high in value, then the tank's water level also rose. Hence, the value of the cost indicates the time duration and iteration notify the level of water.

5.3 Fuzzy logic

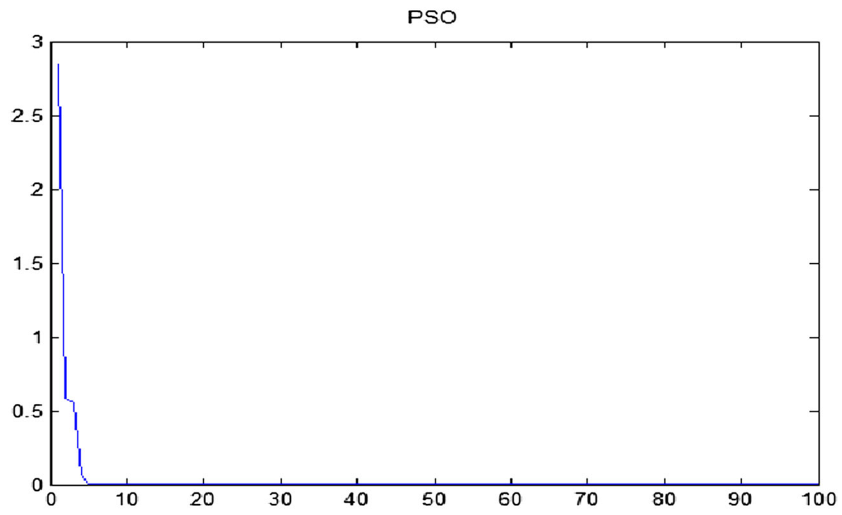
Fuzzy techniques are used in a successful way to control in various fields, Researchers and Engineers consider fuzzy logic algorithms at present for implementing in Embedded Systems, which performs intelligent functions. Fuzzy logic controller delivering better performance is the best one than the traditional controller, which is proved by various researchers [16] as shown in Fig. 9.

The figure is shown above also indicates the presence of tank's water level. It also showed that when time is high, the water level also raises. Later, it remains constant and then decreases.

5.4 Proposed algorithm

The proposed algorithm is said to be a blend of ACO, PSO, and FUZZY. This proposed algorithm can also be utilized to control the level of water level inside the tank. The Matlab 3D view for this process is shown in Fig. 10.

Fig. 7 PSO algorithm



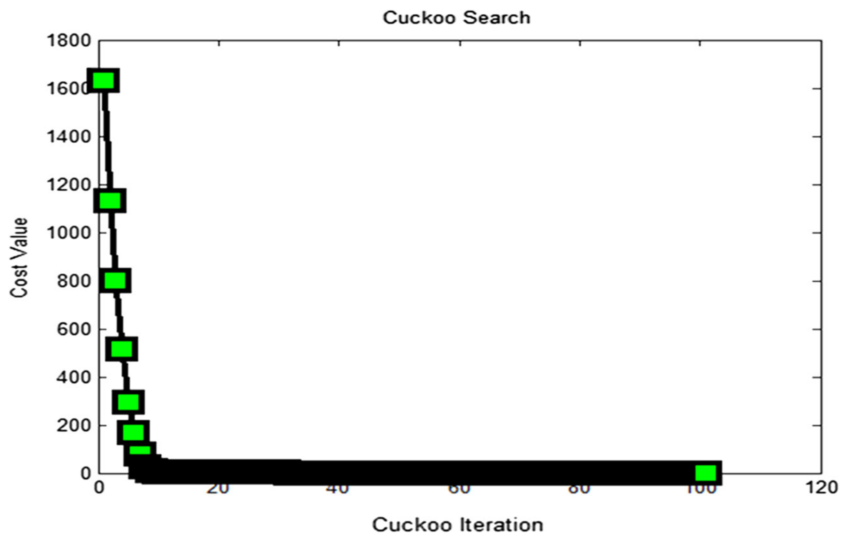


Fig. 8 Cuckoo search algorithm

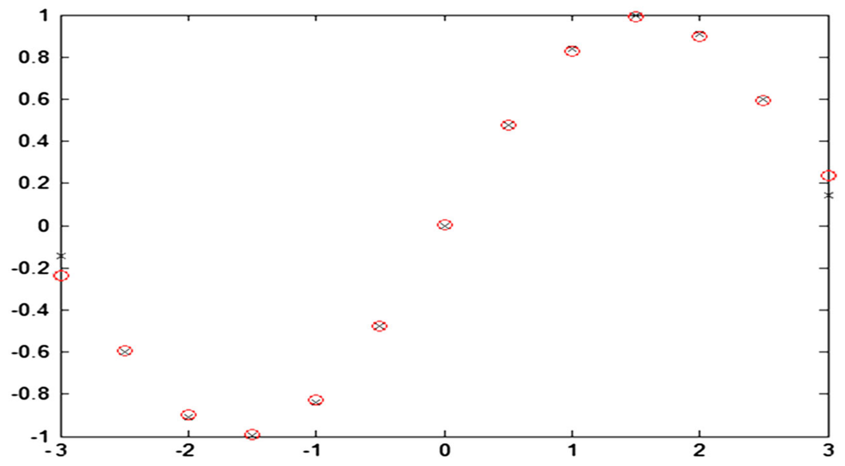


Fig. 9 Fuzzy logic

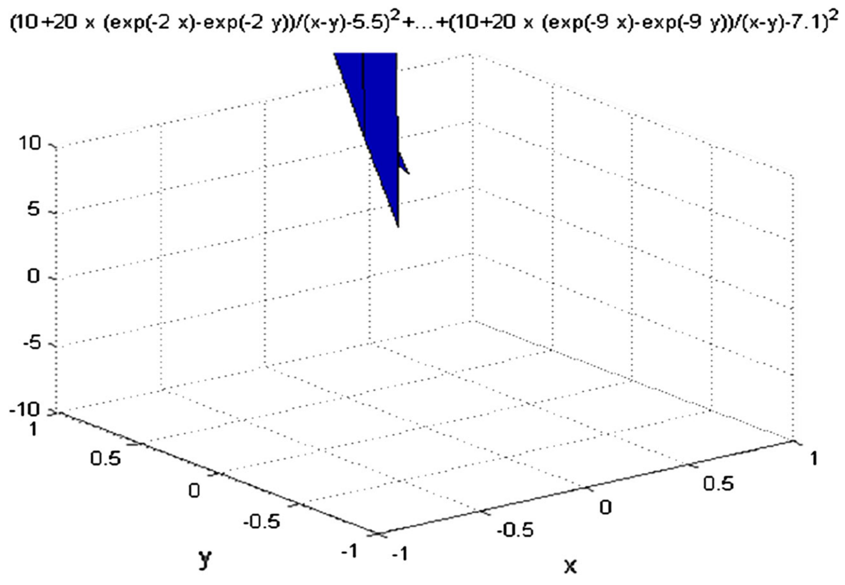


Fig. 10 Novel algorithm

```
Pseudo Code for Novel Algorithm:

Initialization: Initialize position vector and velocity vector of each particle define global arrays
for membership functions
Calculating pbest: Each particle in the pbest is assigned its best position value. If the particle
current fitness value is better than particles pbest, then replace pbest with current position
Calculating gbest: Select the particle's with best fitness value from all particle as gbest
Updating: Update each particle's position vector and velocity vector
    x= [0:0.02:1];
    y= [0:0.02:1];
    z=@(x,y)  (10+20.*x.*(exp(-2*x)-exp(-2*y))./(x-y)-5.5).^2  +(10+20*x.*(exp(-7*x)-exp(-
    7*y))./(x-y)-6).^2   +(10+20*x.*(exp(-9*x)-exp(-9*y))./(x-y)-7.1).^2   ;+(10+20*x.*(exp(-
    9*x)-exp(-9*y))./(x-y)-6.1).^2 +(10+20*x.*(exp(-14*x)-exp(-14*y))./(x-y)-7.2).^2;
    while (t Fj ), replace j by the new solution;
    end
    Calculating the fitness function
    Choose a nest among n (say, j) Randomly
    if (Fi > Fj ),
    replace j by the new result;
    end
    A fraction (pa) of worse nests isuncontrolled and new
    Ones are built;
    Keep the best solutions (or nests with quality results);
    Rank the solutions and discover the present best
    end while
     Post process results and visualization
    end
```

6 Water tank control

Water tank system is used to illustrate both advanced and traditional multivariable strategies. In Shingare and Joshi [17] quantitative feedback theory has been employed to the robust controller in order to regulate the level of liquid in two glass tanks. A multivariable laboratory process of four interconnected water tanks is considered in Vadigepalli et al. [18] for robust control and modeling of the tank level. In Vadigepalli et al. [18], flow and temperature of a water tank system are controlled employing a backstepping method, whereas a four-tank system laboratory experiment is designed in Rusli et al. [19] to illustrate the effects that time-varying dynamics can have on controllability. Also, a predictive fuzzy modeling technique is employed in Liutkeviius and Dainys [20] for analysis of a closed water tank. Furthermore, they applied Internal Mode Control–based robust tunable controller design technique to a water tank control system in Duan et al. [21].

7 Comparison of the proposed approach with existing method

In order, the compare the proposed methodology with the existing method the study has adopted three methods FCFS (First come first server), SJF (Shortest job first), RR (Round robin). In our present, the proposed methodology is been compared in terms of turnaround time, burst time, and waiting time.

7.1 FCFS

It is a non preemptive scheduling algorithm, in this procedures are reserved in a line which are performed one by one. If new procedure attains means then it will go in the end of the queue. Therefore, it is the final procedure in the queue. Here and now, if an innovative procedure reaches after this procedure, that innovative procedure develops the previous method in the queue. When a running procedure is choked the following procedure in the queue is run and the blocked process (when it is ready) is placed at the end of the queue. It is frequently used in batch systems. This

FCFS algorithm is very easy to implement. but it is non-preemptive, CPU usage can be wasted in some situations.

7.2 SJF

It is a non-preemptive, pre-emptive scheduling algorithm. Finest method to diminish waiting time. Easy to implement in Batch systems where needed CPU time is known in advance. Incredible to implement in interactive schemes where essential CPU time is not known. The computer should know in advance how much time process will take.

7.2.1 Advantage

1. Short processes are handled very quickly.
2. The system also requires very little overhead since it only makes a decision when a process completes or a new process is added.
3. When a new process is added the algorithm only needs to compare the currently executing process with the new process, ignoring all other processes currently waiting to execute.

7.2.2 Disadvantage

1. Like shortest job first, it has the potential for process starvation.
2. Long processes may be held off indefinitely if short processes are continually added.

7.3 RR

It is a preemptive scheduling algorithm. In this algorithm, each procedure is measured as equivalent. A time interval to run known as quantum is assigned to every process. A queue is preserved by quantum to each procedure. After a process has finished its quantum, it is placed at the end of the queue and a new process is run from the queue. Round Robin scheduling algorithm is used for personal computers and servers. The performance of the Round Robin Scheduling Algorithm relies on the size of the time quantum. At one extreme, if the time quantum is extremely large, cause less response time and it is similar to FCFS. If the time quantum is extremely small this causes too many context switches and lowers the CPU efficiency.

Figure 11 described the comparison of turn-around time with existing algorithm and proposed advanced PSO algorithm. It is found that in all the process advanced PSO outperforms the existing algorithms like FCFS, SJF, and RR.

In the Fig. 11, the value of FCFS in process 1 (P1) is 2.25%, for P2 is 3.20% and for P3 is 1.9%. The value of

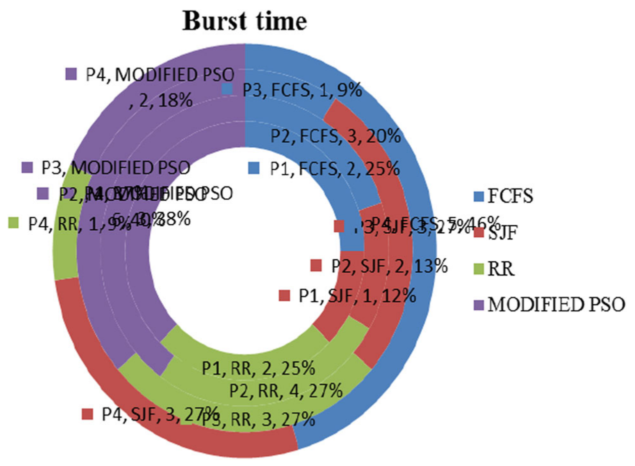


Fig. 11 Comparison of turnaround time

SJF for P1 is 1.12%, for P2 is 2.13% and for P3 is 2.16%, and P4 is 3.27%. The value of RR for P1 2.25%, for P2 is 4.27% and for P3 is 3.27%. Compared to these results proposed Modified PSO results gives better results as shown in above Fig. 11.

Figure 12 described the comparison of burst time with existing algorithm and proposed advanced PSO algorithm. It is found that in all the process advanced PSO outperforms the existing algorithms like FCFS, SJF, and RR.

Ian the above Fig. 12, The value of SJF for P2 is 1.5%, P3 is 3.10% and for P4 is 6.23%. The value of RR for P2 is 7.35% and for P3 is 9.31% and for P4 is 6.23%. Compared to these results proposed Modified PSO results gives better results as shown in above Fig. 12, like Modified PSO value for P1 is 2.100%, P2 is 10.50%, P3 is 12.42% and P4 is 8.31%.

Figure 13 described the comparison of burst time with existing algorithm and proposed advanced PSO algorithm.

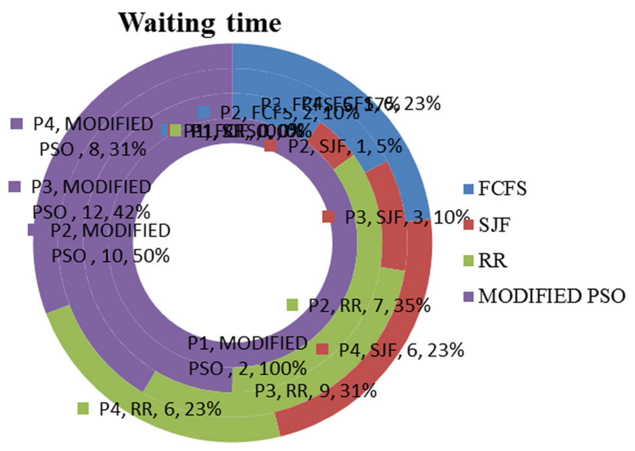


Fig. 12 Comparison of the burst time

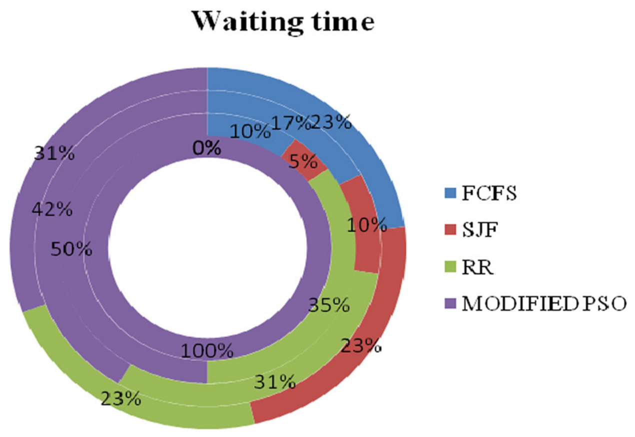


Fig. 13 Comparison of the waiting time

It is found that in all the process advanced PSO outperforms the existing algorithms like FCFS, SJF, and RR.

In this above figure, the value of FCFS in process for P2 is 10%, for P3 is 17% and for P4 is 23%. The value of SJF for P1 is 5%, for P2 is 10% and for P3 is 23%. The value of RR for P1 35%, for P2 is 31% and for P3 is 23%. Modified PSO value for P1 is 100%, P2 is 50%, P3 is 42% and P4 is 31%.

8 Conclusion

Based on the results attained, it is found that the proposed modified PSO outperforms the existing algorithms such as FCFS, SJF, and RR. The proposed modified PSO technique method is designed in such a way that it controls the water level in the tank system, and gives better results than existing techniques. It is also found that the waiting time, turnaround time and burst time is tending to be higher in proposed algorithm. In future, it can be extended to implement in any other real-time application. Priority driven preemptive scheduling is the one used in most time-sharing systems. In this we introduced novel algorithm in proposed technique. It is useful in many real-time applications such as; In non-real-time systems, the priority of a job changes depending on whether it is CPU-bound or I/O-bound. Static table-driven scheduling resources are required to meet the deadlines of safety–critical tasks must be reallocated so that they can be guaranteed a priority. Static table-driven approaches used for the analysis of static schedule ability. Dynamic planning-based approaches results of the feasibility analysis is a schedule or plan that is used to decide when a task can begin execution.

References

1. Singh, J., Singh, G.: Improved task scheduling on parallel system using genetic algorithm. Int. J. Comput. Appl. **39**, 17–22 (2012). https://doi.org/10.5120/4912-7449
2. Kwok, Y.-K., Ahmad, I.: Static scheduling algorithms for allocating directed task graphs to multiprocessors. ACM Comput. Surv. **31**, 406–471 (1999). https://doi.org/10.1145/344588.344618
3. Rajak, R., Katti, C.: Static task scheduling algorithm with minimum distance for multiprocessor system (STMD). Smart Comput. Rev. (2015). https://doi.org/10.6029/smartcr.2015.02.004
4. Kaur, R., Kaur, R.: Multiprocessor scheduling using task duplication based scheduling algorithms: a review paper. Int. J. Appl. Innov. Eng. Manag. **2**, 311–317 (2013)
5. Gujarati, A., Cerqueira, F., Brandenburg, B.B.: Multiprocessor real-Time Scheduling with Arbitrary Processor Affinities: From Practice to Theory. Germany (2014)
6. Rajak, N., Dixit, A.: Classification of list task scheduling algorithms: a short review paper. J. Ind. Intell. Inf. **2**, 320–323 (2014). https://doi.org/10.12720/jiii.2.4.320-323
7. Boveiri, H.R.: Multiprocessor task graph scheduling using a novel graph-like learning Automata. Int. J. Grid. Distrib. Comput. **8**, 41–54 (2015)
8. Sharma, A., Kaur, M.: An efficient task scheduling of multiprocessor using genetic algorithm based on task height. Int. J. Hybrid. Inf. Technol. **8**, 83–90 (2015)
9. Kutil, M., Sucha, P., Capek, R., Hanzalek, Z.: Optimization and scheduling toolbox. In: Leite EP (ed) Matlab - Modelling, Programming and Simulations, pp 239–276. (2010)
10. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, IEEE, pp 1942–1948. (1995)
11. Parsopoulos, K.E., Vrahatis, M.N.: Recent approaches to global optimization problems through particle swarm optimization. Nat. Comput. **1**, 235–306 (2002). https://doi.org/10.1023/A:1016568309421
12. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**, 58–73 (2002). https://doi.org/10.1109/4235.985692
13. Van Den, Bergh F., Engelbrecht, A.P.: A study of particle swarm optimization particle trajectories. Inf Sci (Ny) **176**, 937–971 (2006). https://doi.org/10.1016/j.ins.2005.02.003
14. Tripathi, P.K., Bandyopadhyay, S., Pal, S.K.: Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients. Inf. Sci. (Ny) **177**, 5033–5049 (2007). https://doi.org/10.1016/j.ins.2007.06.018
15. Kamat, S., Karegowda, A.G.: A brief survey on cuckoo search applications. International Conference on Advances in Computer & Communication Engineering (ACCE - 2014), pp. 7–14. Department of CSE & ISE, Vemana Institute of Technology (2014)
16. Nour, M., Ooi, J., Chan, K.Y.: Fuzzy logic control vs. conventional PID control of an inverted pendulum robot. In: 2007 International Conference on Intelligent and Advanced Systems, IEEE, pp 209–214. (2007)
17. Shingare, P., Joshi, M.A.: Modeling and robust control of level in hybrid tanks. Proceedings of the world academy of science, engineering and technology, pp. 279–283. The Pennsylvania State University, State College (2007)
18. Vadigepalli, R., Gatzke, E., Doyle, F.: Robust control of a multivariable experimental four-tank system. Ind. Eng. **40**, 1916–1927 (2001). https://doi.org/10.1021/ie000381p

19. Rusli, E., Ang, S., Braatz, R.D.: Quadruple tank process control experiment. J. Chem. Eng. Educ. **38**, 1–25 (2004)
20. Liutkeviius, R., Dainys, S.: Hybrid fuzzy model of a nonlinear plant. Inf. Technol. Control **34**, 51–56 (2005)
21. Duan, Y., Boulet, B., Michalska, H.: Application of IMC-based robust tunable controller design to water tank level regulation. Proceeding MIC'07 Proceedings of the 26th IASTED International Conference on Modelling, Identification, and Control, pp. 285–290. ACTA Press Anaheim, CA (2007)

Joel Josephson did his Bachelors of Technology in Electronics and Communication Engineering from Jawaharlal Nehru Technological University, Hyderabad. India. He did His Master of Technology in Embedded Systems from Jawaharlal Nehru Technological University, Hyderabad. India. His field of Interest is Embedded System Design; currently he is Pursuing his Research in Anna University Chennai.

R. Ramesh did his Bachelors in Engineering in Electrical and Electronics Engineering from University of Madras, and Masters in Engineering in Power Systems from Anna University. Currently he is working as Associate Professor in the Department of Electrical and Electronics Engineering in College of Engineering Guindy, Anna University Chennai.