

# An Optimal Scheduling Algorithm for Real Time Applications in Grid System

S.Baghavathi Priya<sup>1</sup>, T.Ravichandran<sup>2</sup>

<sup>1</sup> Research Scholar, Jawaharlal Nehru Technological University, Hyderabad, Andhra Pradesh, India.  
Associate Prof. /IT. Rajalakshmi Engineering College Chennai, Tamil Nadu India.

<sup>2</sup> Principal, Hindustan Institute of Technology, Coimbatore, Tamil Nadu, India.

## Abstract

The objective of the proposed work is to use an optimal scheduling algorithm for real-time application. A grid is considered to be an infrastructure that bonds and unifies globally remote and diverse resources in order to provide computing support for a wide range of applications. Real time applications in an industrialized technological infrastructure such as telecommunication systems, factories, defense systems, aircraft and space stations pose relatively rigid requirements on their performance. Aircraft scheduling represents the best example of real-time applications. The main focus of this work is to check the time taken for turn-around activities which comprises of taxi in, load/unload baggage, deboarding, water fueling, cleaning, catering, boarding, de-icing, take off processes, thus relating in the lowest flight delays and shortest waiting time. The optimal scheduling algorithm is used for aircraft take-offs. The penalties are associated with proper scheduling but delayed turn around activities, improper scheduling and early/late takeoffs.

**Keywords:** *Grid computing, Real-time systems, Task-Scheduling, Turn-around activities, Penalty.*

## 1. Introduction

The Grid computing refers to a new technology infrastructure paradigm which is based on the Web and the Internet. Grids provide scalable, secure, and reliable mechanisms for discovering and negotiating access to remote resources including clusters. Grid applications involve large amounts of data and/or computing and often require secure resource sharing across the organizational boundaries. Grid computing comprises of a combination of a decentralized architecture for resource management on one hand and a layered architecture of a specific hierarchy for the implementation of various services of the grid. In a grid environment, heterogeneous computer systems located over a large area or around the globe can be integrated and made to appear as a single computational resource, to be optimally or maximally utilized by the user community without any loss or wastage of time, investment or resources.

The consistent decrease in the cost of hardware has led to the employment of computers in many applications. As a consequence, the complexity of modern computer systems has increased. Proportionally, and more effort is needed to maintain the dependability of these computer systems. Systems in which the complexity exists in the dimension of time are called real-time systems. Examples of current real-time systems range from very simple micro controllers in embedded systems to highly sophisticated and complex systems such as air traffic control and avionics. A typical real-time system consists of a controlling system, a controlled system, and the environment. The environment in which a real-time system is to operate plays an important role in the design of the system. Many environments are well defined and deterministic. Many applications will be large, complex, distributed and dynamic; contain many types of timing constraints; need to operate in fault-prone highly nondeterministic environments, and evolve over a long system life time. Issues in real-time systems are resource management, architecture, software. Resource management issues deal with scheduling, resource reclaiming, fault tolerance and communication.

Task scheduling meeting task deadlines is of great importance in real-time systems, because failure to meet task deadlines may result in severe consequences possibly loss of human life. Scheduling of tasks involves the allocation of processes (including resources) and time to tasks in such a way that certain performance requirements are met. The scheduling algorithms have to satisfy not only the timing constraints of tasks, but also the resource constraints and/or precedence constraints among tasks. Similarly, providing predictable inter task communication is also of great significance in real-time systems, because unpredictable delays in the delivery messages can affect the completion time of tasks participating in the message communication. Classical task-scheduling theory typically employs metrics such as minimizing schedule length and the sum of completion times. In static scheduling, the performance requirement is that every task in the system

meets its deadline. In dynamic scheduling, however, the system may be unable to meet the deadlines of all tasks because it lacks knowledge about future task arrivals.

As the history of the field suggest there are many different variants of evolutionary algorithms. The common underlying idea behind all these techniques is the same: given a population of individuals the environmental pressure causes natural selection and this causes a rise in the fitness of the population. Given a quality function to be maximized, a set of candidate solutions should be randomly created. Based on this fitness, some of the better candidates are chosen to seed the next generation by applying recombination and/or mutation to them. Recombination is an operator applied to two or more selected candidates and results one or more new candidates. Mutation is applied to one candidate and results in one new candidate. Executing recombination and mutation leads to a set of new candidates that compete based on their fitness with the old ones for a place in the next generation. This process can be iterated until a candidate with sufficient quality is found or a previously set computational limit is reached.

The rest of the paper is organized as follows; Section 2 illustrates the related work on real-time applications and new approach. Section 3 describes proposed architecture for aircraft take-off system. Section 4 describes optimal scheduling algorithm for real-time applications. The experimental results are discussed in section 5. Section 6 concludes the paper and presents the future work.

## 2. Related work and Our New approach

We first review related work on real-time applications in computational Grid. Then, we use evolutionary algorithm for providing optimal solution to complex problem.

### 2.1 Related Previous Work

Grid Computing concepts are clearly explained in Grid and Cluster computing. Online scheduling approach was proposed for multiple mixed-parallel workflows in grid environments [1]. Resource management issues deal with scheduling, resource reclaiming, fault tolerance and communication, architecture issues involve processor architecture, network architecture and i/o architecture, software issues encompass specification and verification of real-time systems, programming languages and databases. These are the major issues in real-time systems [2]. Task scheduling, preemptive and non-preemptive scheduling, resource reclaiming, static scheduling algorithms and dynamic scheduling algorithms are dealt in deadline scheduling for real-time systems [3]. Resource

management in a dynamic real-time system involves both scheduling and reclaiming of resources are explained resource management in real-time systems and networks [4]. Airport logistics, which is a framework of resource management in the air transportation system and focuses on processes supporting turn-around [5]. The future traffic management is the use of planned four-dimensional trajectories (4DTs) for airborne and surface operations at busy airports were explained [6]. The optimization of airport operations is recognized as a challenge that aims at finding the best trade-off solution in order to maximize the airport capacity and minimize both pollution and noise [7]. Evolutionary algorithms (EAs) are stochastic optimization techniques based on the principles of natural evolution. Numerous applications of these techniques were explained [8]. Efficient scheduling of the aircraft for take-off can reduce the total separations and increase throughput. A runway controller is responsible for take-off scheduling. These concepts were explained [9].

### 2.2 Our New Approach

The scheduling of tasks in grid real-time systems has recently attracted many researchers. The demand for more and more complex real-time applications, which have high computational needs with timing constraint. The first objective is to check the performance of the turn-around activities by calculating the delay and waiting time for completing that particular activity. In most real-time applications, computational complexity is a challenging factor. Priority based scheduling algorithm is used to check the time taken for turn-around activities. Evolutionary algorithms can solve often complex problems. Evolutionary algorithms, such as shuffled frog-leaping optimization algorithm has been used to arrive at near-optimum solutions to complex and large-scale problems. The third objective is to report penalty by using objective function. This function has been used for providing penalty for sub-optimal solutions and invalid results.

## 3. Grid Scheduling

### 3.1 The Grid Scheduling Architecture

The Fig.1 shows the architecture and functionalities supported by various units of the grid scheduling system. One of the most complex airport processes is the turn-around process. The turnaround is the collective name for all those activities that affect an aircraft while it is on the ground. One process where most of the actors are involved is the turn-around process. The turn-around process starts when an aircraft touches down and is going on until the aircraft takes off again. Grid agent is to check the

performance of the turn-around activities by calculating the delay and waiting time for completing that particular activity. Global scheduler receives the information about the status of turn-around activity. Scheduled has been prepared according to the information received from Grid Information Service (GIS).GIS provides the information about the workstation. Local scheduler provides the information of the resources (runways) availability to the GIS. If there were no delay in completing the turn-around activity then prepared schedule will be executed. Otherwise the schedule should be revised. If there were no delay in completing the schedule then the aircraft will take-off as per schedule. Otherwise the scheduling delay will be informed to the GIS, in turn it will be reported to the scheduling agent to pay the penalty.

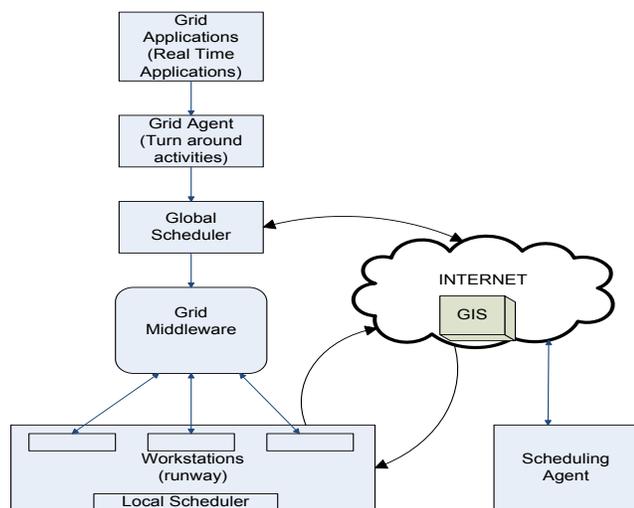


Fig. 1 The Grid Scheduling Architecture.

### 3.2 Priority Based Scheduling

In priority based scheduling algorithm, each process is assigned and the higher priority processes are scheduled before the lower priority processes. At any point of time, the process having the highest priority among all the ready processes is scheduled first. In case two processes are having the same priority, they are executed in the first come first serve order. The priority scheduling may be either preemptive or non-preemptive. The choice is made whenever a new process enters the ready queue while some process is executing. If the newly arrived process has the higher priority than the currently running process, the preemptive priority scheduling algorithm preempts the currently running process and allocates CPU to the new process. On the other hand, non-preemptive scheduling

algorithm allows the currently running process to complete its execution and the new process has to wait for the CPU. In the case of scheduling turn-around activities in air traffic management system, non-preemptive scheduling algorithm is used.

Consider four aircrafts A1, A2, A3, A4 with their required processing time (in milliseconds) and priorities as shown in the following table.

Table 1: Priority Scheduling

Aircrafts	A1	A2	A3	A4
Processing time(ms)	7	4	3	2
Priority	4	3	1	2

The aircrafts will be scheduled for completing turn-around activities as A3, A4, A2, and A1.

A3	A4	A2	A1
0	3	3	2
5	4	9	7
16			

Waiting time of A1 = 9 ms Turn around time for A1 = 16ms  
 Waiting time of A2 = 5 ms Turn around time for A2 = 9ms  
 Waiting time of A3 = 0 ms Turn around time for A3 = 3ms  
 Waiting time of A4 = 3 ms Turn around time for A4 = 5ms

### 4. Evolutionary Algorithm

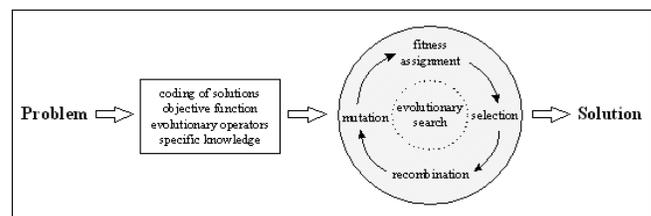


Fig. 2 Problem solution using evolutionary algorithm.

The Fig. 2 shows problem solution using evolutionary algorithm. This section presents the Shuffled Frog-leaping algorithm used in this paper for providing near-optimum solutions.

The Shuffled Frog-Leaping (SFL) algorithm is a memetic metaheuristic that is designed to seek a global optimal solution by performing a heuristic search. It is based on the evolution of memes carried by individuals and a global exchange of information among the population (Eusuff and Lansey 2003). In essence, it combines the benefits of the local search tool of the particle swarm optimization (Kennedy and Eberhart 1995), and the idea of mixing information from parallel local searches to move toward a global solution (Duan et al. 1993). The SFL algorithm has

been tested on several combinatorial problems and found to be efficient in finding global solutions (Eusuff and Lansey 2003).

The SFL algorithm involves a population of possible solutions defined by a set of frogs (i.e. solutions) that is partitioned into subsets referred to as memeplexes. The different memeplexes are considered as different cultures of frogs, each performing a local search. Within each memeplex, the individual frogs hold ideas, that can be influenced by the ideas of other frogs, and evolve through a process of memetic evolution. After a number of memetic evolution steps, ideas are passed among memeplexes in a shuffling process (Liong and Atiquzzaman 2004). The local search and the shuffling processes continue until convergence criteria are satisfied (Eusuff and Lansey 2003). The SFL algorithm is described by the corresponding flowchart in figure 3.

First, an initial population of 'P' frogs is created randomly. For S-dimensional problems, each frog i is represented by S variables as  $X_i = (x_{i1}, x_{i2}, \dots, x_{iS})$ . The frogs are sorted in a descending order according to their fitness. Then, the entire population is divided into m memeplexes, each containing n frogs (i.e.  $P = m \cdot n$ ). In this process, the first frog goes to the first memeplex, the second frog goes to the second memeplex, frog m goes to the mth memeplex, and frog m+1 goes to the first memeplex, and so on.

Within each memeplex (figure 1b), the frogs with the best and the worst fitness are identified as  $X_b$  and  $X_w$ , respectively. Also, the frog with the global best fitness is identified as  $X_g$ . Then, an evolution process is applied to improve only the frog with the worst fitness (i.e. not all frogs) in each cycle. Accordingly, the position of the frog with the worst fitness is adjusted as follows:

$$\text{Change in frog position } (D_i) = \text{rand}() * (X_b - X_w) \quad (1)$$

$$\text{New position } X_w = \text{Current position } X_w + D_i; \quad (2)$$

$$(D_{\max} \geq D_i \geq -D_{\max})$$

Where  $\text{rand}()$  is a random number between 0 and 1; and  $D_{\max}$  is the maximum allowed change in a frog's position. If this process produces a better frog (solution), it replaces the worst frog. Otherwise, the calculations in equations (1) and (2) are repeated with respect to the global best frog (i.e.  $X_g$  replaces  $X_b$ ). If no improvement becomes possible in this latter case, then a new solution is randomly generated to replace the worst frog with another frog having any arbitrary fitness. The calculations then continue for a specific number of evolutionary iterations within each memeplex (Eusuff and Lansey 2003). The main parameters of the SFL algorithm are: number of frogs P, number of

memeplexes, and number of evolutionary iterations for each memeplex before shuffling.

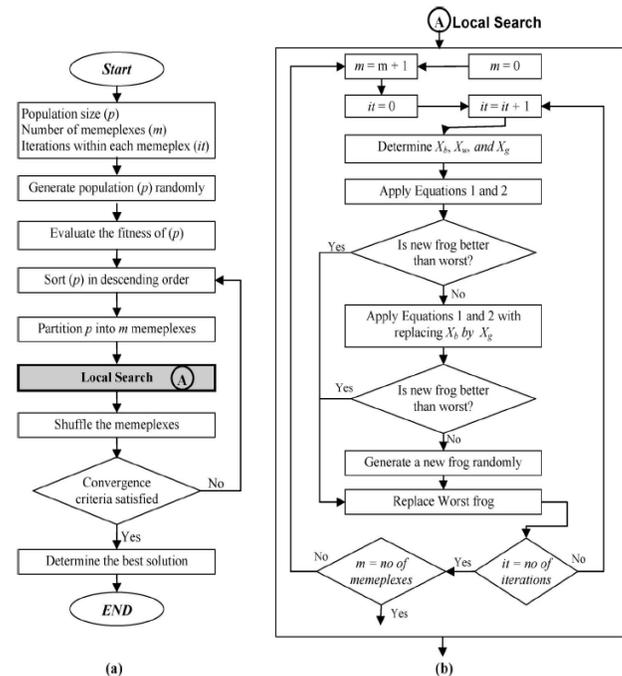


Fig 3. Flowchart of the shuffled frog-leaping algorithm

Penalties help to limit the turn-around activities delay and improper scheduling of an aircraft. Penalties considered in this paper include aircraft delay penalty ( $P_A$ ), turn-around activities delay penalty ( $P_{TA}$ ), improper schedule penalty ( $P_S$ ) of an aircraft. The trade-off situation occurs as the system schedules turn-around activities to maximize proper scheduling while doing fastest turn-around activities for aircraft. The objective function of this model is to minimize total system penalties ( $P_T$ ) while doing proper scheduling through optimal scheduling algorithm.

The objective function can be expressed as :

$$P_T = \alpha P_E + (1 - \alpha) P_S \quad (1)$$

Where

$P_E$  ----- Expected aircraft delay penalty and turn-around activities.

## 5. Results and Discussions

Gridsim Toolkit 4.0 which allows modeling and simulation entries in grid system.



Fig 4. Implementation of an algorithm

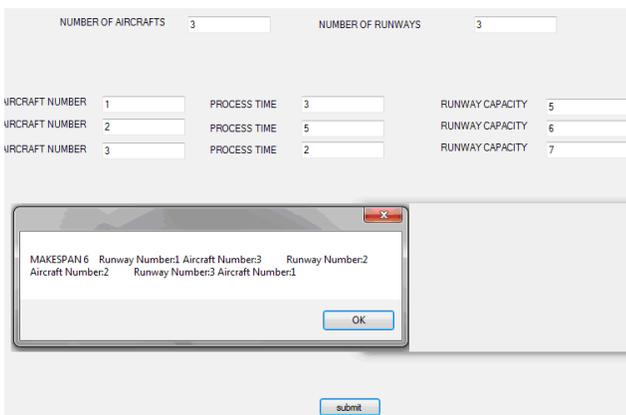


Fig 5. Initial order of assignment

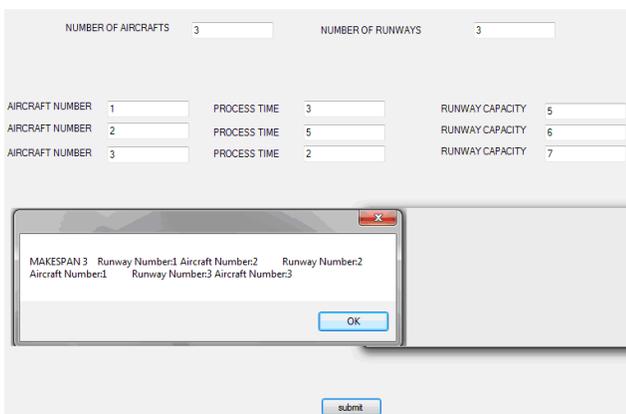


Fig 6. Final order of assignment

## 6. Conclusions and Future Work

In this paper, we have checked delay and waiting time of turn-around activity. Based on the calculation, scheduling of aircraft will be planned in suitable runway for achieving

minimum completion time. The shuffled frog-leaping algorithm was used and implemented for aircraft take-off. This algorithm gives the results of order of aircraft and its suitable runway in a optimized way. Penalty will be calculated and reported to a scheduling agent if any delay in turn-around activity or delay in scheduling. In the future, we plan to bring fault tolerant system in grid real-time applications.

## References

- [1] C.S.R.Prabhu, “Grid and cluster computing”, (Deputy Director General, National Informatics Centre, A.P, Hyderabad) Prentice Hall of India Private Limited 2008.
- [2] Jane W.S.Liu, “ Real-Time Systems, Pearson, 2000.
- [3] John A. Stankovic, Marco Spuri, Krithi Ramamritham, Giorgio C.Buttazzo, “Deadline Scheduling for Real-time systems, Kluwer Academic Publishers, 1998.
- [4] C. Siva Ram Murthy and G.Manimaran, “Resource management in real-time systems and networks”, PHI Learning Private Limited, 2009.
- [5] Anna Norin, Tabias Anderson Granberg, Peter Varbrand&Di yuan, “Integrating Optimization and Simulation to gain more efficient airport logistics”, Eighth USA/Europe Air Traffic Management Research and Development Seminar(ATM 2009).
- [6] Stephen Atkins, Christopher Brinton, Yoon Jung, “Implication of variability in Airport Surface operation on 4-D Trajectory Planning”.
- [7] S.Bagassi, D.Francia, F.Persiani, “Simulating airport operations in a synthetic environment”, International Conference on Innovative methods in product design. June 2011.
- [8] Enrique Alba and Carlos cotta, “ Evolutionary Algorithms”, February 19, 2004.
- [9] Jason A.D. Atkin, Edmund K.Burke, John S.Greenwood, Dale Reeson, “Hybrid Metaheuristics to Aid Runway Scheduling at London Heathrow Airport”, Transportation Science, Vol.41, No.1, February 2007, PP.90-106.

**S.Baghavathi Priya** received the B.E Degree in Computer Science and Engineering Sundranar University and M.Tech Degree from Dr.M.G.R Educational and Research Institute, Chennai, India. She is pursuing PhD in Faculty of Computer Science Engineering, Jawaharlal Nehru Technological University, Hyderabad, Andhra Pradesh, India. Presently she is working as an Associate Professor in the department of Information Technology, Rajalakshmi Engineering College, Chennai, India. Her research area includes Grid Computing and Distributed Computing. She has published around 10 papers in National and International conferences and journals.

**Dr.T.Ravichandran** received B.E degree from Bharathiar University, Tamilnadu, India and M.E degree from Madurai Kamaraj University, Tamilnadu, India in 1994 and 1997, respectively and PhD degree from the Periyar University, Salem, Tamilnadu, India, in 2007. He is currently the Principal of Hindustan Institute of Technology, Coimbatore, Tamilnadu, India. His research interests include Distributed Computing, Image Processing, Internet Computing and Security, Mobile Computing, Performance Evaluation and Fault Tolerant Computing. He has published more than 40 papers in National and International Conferences and Journals. He is the life member in professional society of IEEE, CSI and ISTE.