I.J. Modern Education and Computer Science, **2014, 10, 57-83**
Published Online October 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2014.10.08

A Survey on Scheduling Heuristics in Grid Computing Environment

[1]Manoj Kumar Mishra, [2]Yashwant Singh Patel, [3]Yajnaseni Rout, [4]G.B. Mund
[1234]School of Computer Engineering, KIIT University, [1]Innovadors Lab Pvt. Ltd., Odisha, India
Email: {[1]manojku.mishra05, [2]yashwant.patelasct, [3]routyajnaseni}@gmail.com, [4]mundgb@yahoo.com

Abstract—Job scheduling is one of the thrust research area in the discipline of Grid computing. Scheduling in the Grid environment is not only complicated but also known to be NP-Complete problem and that is all due to its unique characteristics. Thus, there are limited opportunities to find an optimal solution. In recent past, many eminent researchers reported a variety of Scheduling Heuristics that can have a substantial impact on the performance of the Grid systems. Unfortunately, it gives rise to difficulty in evaluating and keeping track of those solutions. Therefore, the motivation of this comprehensive study is to present firstly, an in-depth review of the topic under discussion mostly in the perspective of Grid Scheduling environment, and secondly, a proposal for a new state-of-the-art classification of the existing Scheduling Heuristics. All these Heuristics in each category have been further studied based on significant parameters frequently used in Scheduling Heuristics. The final part of this study includes a fair assessment of those mostly used dominating parameters. This report deals with the key concepts behind existing Scheduling Heuristics including Objectives, Types of Job Scheduling, Functionality of Grid, Nature of Grid, and the importance of the proposed classification.

Index Terms—Classification, Grid Computing, Grid Scheduling System, GIS (Grid Information Service), Heuristics.

I. Introduction

In recent years, the phenomenal advancement in computing speed, development in network technologies and demand for solving large scale problems have led us into the era of high performance distributed computing widely known as Grid computing. The concept of Grid was proposed by Foster and Kesselman [1, 2]. They introduced Grid as a collection of large number of heterogeneous, distributed, decentralized and dynamic resources. The resources may be distributed over multiple control domains and interconnected through high speed communication network. The purpose of Grid is to provide a high performance computing platform for executing large scale applications, which may contain a number of dependent or independent jobs. Appropriate scheduling of the applications onto the Grid resources plays an important role in achieving the goals of Grid computing. The rest of this paper is organized as shown in the Table I, given as follows:

Table 1: Organization of the Study

II. Preliminaries of Grid Computing

This section introduces some basic definitions and related terminologies generally used in Grid computing and scheduling environment [3, 4, 5]. The preliminaries are as follows:

i) *Application*: It consists of a number of jobs. Usually, a job may be single unit or a collection of tasks. Job and task can also be represented as a synonym to each other. Each job may have its own attributes like deadline, completion time, execution time etc. A job or a set of jobs can be assigned to a single resource or to a set of resources. Job can be of two types; either they are dependent or independent to each other.

ii) *Site:* A site better known to be a node is formed with a single or multiple numbers of resources.

iii) *Resource:* A resource can be referred as a machine, which contains individual or multiple numbers of processors to perform job execution. Resources can either be software such as program files, software programs etc. or hardware such as processors, computers, display devices, mobile devices or networks of supercomputers etc.

iv) *Optimisation Criterion:* Optimisation criterion is used to make better scheduling decisions while achieving the desired goals. It is represented in terms of objective functions of the scheduling problem.

v) *Quality of Service (QoS):* Quality of Service (QoS) is the degree of satisfaction of a user against the Grid services to achieve the desired level of performance.

III. Grid Scheduling

Scheduling is considered to be a critical part of any well defined system. It is a process that assigns jobs onto some specific available resources with respect to time [4, 6] and manages its execution by minimizing or maximizing certain objective function(s) specified by the user. Scheduling has been extensively studied with respect to traditional, single and distributed computing systems before Grid systems became apparent. But Grid concept has introduced certain additional requirements that make Grid Scheduling different than other form of distributed scheduling systems. It is really hard to study the performance of Grid scheduling strategies under classic performance models. The Conventional Scheduling Heuristics do not suit well in a dynamic and uncertain environment like Grid and also could not satisfy the requirements of the users. Therefore, some Unconventional Scheduling Heuristics are designed to satisfy the performance requirements of Grid users. The goal of Grid scheduling is to make an optimal mapping of tasks to computational resources and utilize them efficiently [7] with the objective(s) of minimizing makespan, flowtime, cost and maximizing load balancing across resources, resource utilization, throughput etc. [8, 9, 10]. Even if, the detailed information concerning to the status of jobs and resources is well-known then also optimal scheduling has been shown to be NP-complete [8, 11, 12, 13, 14, 15].

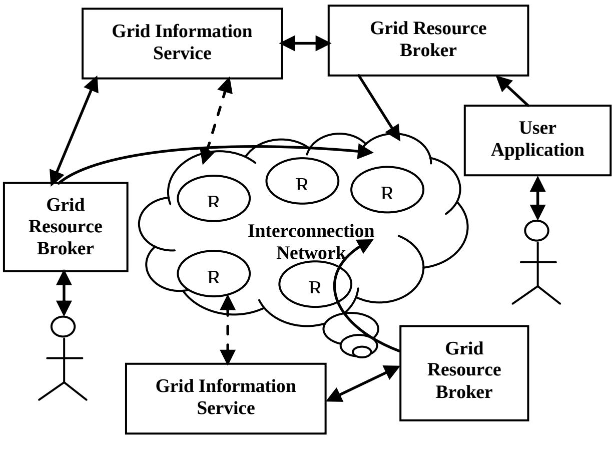


Fig 1: Basic Grid Scheduling Environment

Owing to NP-complete aspect of scheduling problem and uncertainty in Grid scenario, current researches attempt to find sub-optimal or approximation strategies for Grid scheduling problems [3]. In Grid, scheduler collects the state information of resources and selects suitable resources to give best possible schedule for execution of the applications. Grid scheduling mostly depends on two factors one is ordering and other is mapping. In the case of multiple applications, ordering is needed to arrange the pending applications by means of some specific strategy. In mapping, the purpose is to select a set of suitable resources and allocating them to various applications to perform optimal or near optimal scheduling [16]. A basic Grid scheduling environment is shown in Fig. 1.

IV. Major issues & its significance in grid scheduling

This section includes some issues such as Scheduling challenges, Framework, Logical architecture, Grid Middleware and Grid Tools with their significance in context of Grid scheduling.

A. Scheduling Challenges in Grid Platform:

Grid computing is not entirely different from traditional distributed systems. Because, it inherits some of its characteristics from distributed system as well as it includes some of its own unique characteristics that distinguishes it from other traditional distributed systems. Therefore, these special features make the design of Grid scheduling Heuristics more critical and challenging than other traditional distributed scheduling systems. The main characteristics of Grid are referred from [3, 5, 7, 16, 17, 18, 19, 20, 21, 22, 23], and discussed as follows:

i) *Heterogeneity:* Grid includes numerous types of heterogeneous resources that may be software and hardware. Hardware heterogeneity may be in the form of instruction set, computer architectures, number of processor, CPU speed, memory size and so on, and also networks may differ in their bandwidth and communication protocols. Similarly, software heterogeneity may be in the form of cluster management software, operating systems, file systems, and so on. For design and analysis of scheduling Heuristics in such a platform is a challenging task.

ii) *Site Autonomy:* Grid is a collection of multiple administrative sites. Usually each site is an autonomous entity and has its own scheduling policy independent of other sites. The challenging task in site autonomy is that in Grid all sites are not controlled and maintained by the same administrative domain. Unlike most traditional schedulers, the Grid scheduler has only limited control over the resources. Thus, site autonomy greatly influences the design of effective Grid scheduling.

iii) Scalability: Grid is known to be a large-scale distributed system which can be scalable from thousand to millions of nodes connected through networks. Scheduling and maintenance of such a large scale system in a controlled manner is a challenging task.

iv) Performance Dynamism and Adaptability: In traditional scheduling model, such as a cluster, the pool of resources is assumed to be stable. But Grid exhibits dynamic behaviour due to the frequent failures, uncertain availability and changing capability of its resources. Making scheduling decisions in such a highly dynamic Grid platform is a difficult task. Hence, Scheduling Heuristics designed for Grid should be feasible and adaptive to such dynamic behaviours.

v) Coordination: Grid users need assurances from the scheduler that they will receive expected, persistent and often desired levels of performance. So, resources must be coordinated by means of some mechanism to support the expected level of performance. Unlike most traditional schedulers, the Grid scheduler has limited control over all the resources due to its autonomy and performance dynamism nature. So, it is really a big challenge to obtain an effective scheduler for Grid system.

vi) Transparency: Grid provides an abstraction by giving a single system image or behaves as a single virtual computing node to maintain transparency to the users, without providing the details of underlying complexities and physical locations of the resources. Scheduling requires the development of tools that manage transparency is a challenging task.

vii) Reliability: In Grid, reliability means to find mechanisms to handle issues like heterogeneity in hardware and software, various policies of different organizations, use of combined computational power of Grid resources, continuously changing performance and availability of Grid resources and resource failures. It can be observed that overall performance fluctuation is resulted from inherent Grid characteristics. So, the interested researchers may try to make feasible scheduling Heuristics that should be adaptive to such Grid characteristics and diversity of user applications.

B. A Framework for Grid Scheduling Architecture:

A framework for scheduling is the beginning point that provides facilities, or specifies common requirements to build a complete and well defined architecture as shown in Fig. 2.

The framework includes a scheduling policy, which is responsible for defining the performance model of a Grid system. The role of scheduling policy is to take appropriate decisions on applications execution and resource utilization. Performance model must be able to properly represent both the application model and resource model for predicting the performance potential of a schedule. As the quality of performance prediction substantially influences the effectiveness of the scheduler. Grid users are concerned about application performance through the design of a proper application model. The resource model represents resource owners' concern about overall utilization of the whole system. Application model involves representation of each characteristic of scheduled application while resource model describes the characteristics of the underlying Grid resources.

C. A Logical Architecture of Job Scheduling Sub-System:

The development of an effective scheduling system is extremely necessary to utilize the full potential of Grid system [16]. So, a logical architecture is essential for the designing of an efficient scheduling system. Some logical architecture designs are discussed in [24, 25, 26, 27]. Thus, a generic logical architecture of job scheduling system is suggested, as shown in Fig. 3. A common scheduling architecture entails various functional components along with the defined course of interactions among them to realize the principles and flow of job scheduling process. This sub-section includes the discussion on a) Components of Job Scheduling System and their Interaction, b) The Functionality of Scheduling Components and c) The Common Job Scheduling Process in Grid Environment.

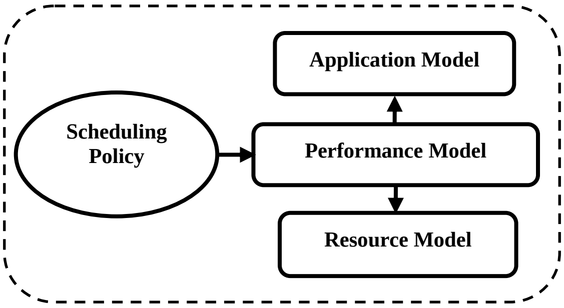


Fig 2: Framework for Grid Scheduling Architecture

a. Components of Job Scheduling System:

This comprises of the following components:

i) Grid Users, ii) Job Scheduler, iii) GIS (Grid Information Service), iv) Grid Resources, v) Communication Network.

Grid Users interact with the job scheduler through application portal. Job scheduler interacts with GIS with the help of high speed communication network such as Internet. GIS is responsible for providing information about various geographically distributed resources.

b. Functionality of Scheduling Components:

i) *Grid Users:* Grid Users submit application along with other specifications through the application portal to the job scheduler and receive the computational results returned by the scheduler. User applications are represented by a tuple specifying its requirements such as resource requirements, performance models, application types and optimization objectives. The specification is necessary for proper performance prediction. Such specifications can be provided by the users or can be extracted with the help of application profiling.

ii) *Job Scheduler:* Job scheduler contains some sub components such as scheduling Heuristic, Grid resource and network information table, dispatcher and output receiver. Each part is having its own functionality as follows:

Scheduling Heuristic: To find a better mapping between jobs and resources.

Grid Resource and Network Information Table: It contains the filtered information about Grid resources and network related information provided by GIS in response to the query.

Dispatcher: The mapping of jobs to the resources decided by the scheduling Heuristic is actually realized through dispatcher.

Output receiver: It is used as a temporary buffer to hold the computational results provided by the Grid resource.

iii) *Grid Information Service:* One role of GIS is to collect and predict the nature of resource and network state information for instance resource availability, CPU capacity, memory size, software availability, load on a site and network bandwidth within a given time period. The other role of GIS is to provide resource and network related information to the scheduler in response to the user queries. Some widely used GIS to solve Grid scheduling problems are Metacomputing Directory Service (MDS) [28], Network Weather Service (NWS) [29], Ganglia [30], Giggle [31] etc.

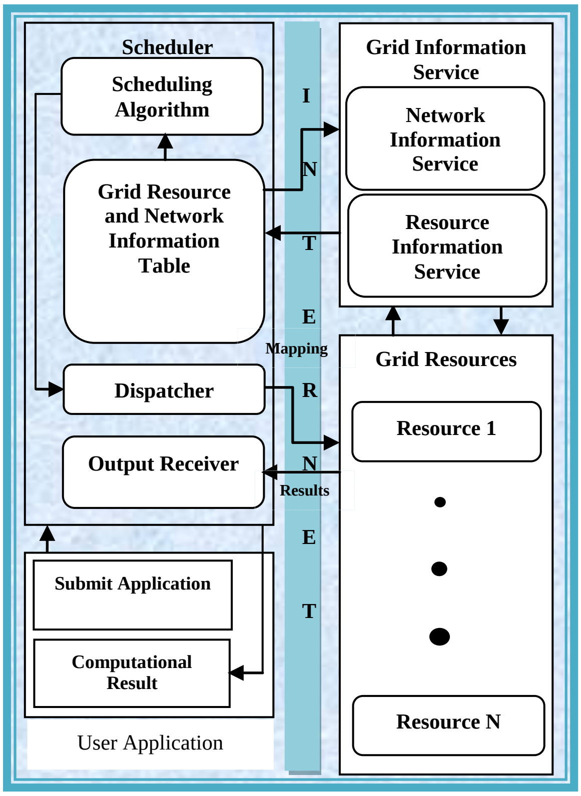


Fig 3: Logical Architecture of Job Scheduling Sub-System

iv) *Grid Resources:* In the Grid scheduling environment, resources are heterogeneous, dynamic and geographically distributed in nature. It can be hardware or software such as storage devices, computers and software applications. The function of resources is to receive and execute the assigned jobs and then provide the computational results to the Grid scheduler.

v) *Communication Network:* The communication is one important component in a Grid environment and is realized through high speed communication network such as Internet.

c. The Common Job Scheduling Process in GridEnvironment:

The phases of common job scheduling process are Resource Discovery, Resource Filtering, Resource Selection and Scheduling Policy [3, 5, 32, 33]. These phases are illustrated as given below:

i) *Resource Discovery:* The key role of Grid infrastructure is to support a resource discovery mechanism for an efficient management of Grid resources [34]. In this phase of resource discovery, up-to-date status information and geographical locations of authorized and reliable resources are gathered by GIS.

ii) *Resource Filtering:* This is the resource elimination phase i.e. those resources are filtered out which do not meet the application requirements specified by the user. Therefore, in the next phase of scheduling, only those resources will participate, which satisfies the application requirements.

iii) *Resource Selection:* This can be termed as the next phase of resource elimination, i.e. some more resources may be filtered out. The purpose of this phase is to determine a set of feasible resource(s) which better satisfies the user requirements. This selection phase requires the detailed dynamic status information for proper ranking of those resources, which most likely to satisfy the scheduling objectives.

iv) *Scheduling Policy:* It defines a proper strategy for mapping of jobs to the resources and finally, their submission. (i) Job Mapping: Here, jobs are allocated to the selected resources based on some scheduling strategy. (ii) Job Submission and Gathering of Computed results: This is the last step of common job scheduling process. In this phase, dispatcher submits the jobs to the respective resources and keeps on monitoring till its completion. The computational results are collected from the resource(s) through the output receiver.

D. Grid Middleware:

The Grid Middleware is an essential part of the Grid system which resides between Grid applications and networked resources. Grid middleware facilitates the management of Grid applications and provides the infrastructure where users can connect to the desired computing resources, without knowing its geographical location. It offers a collection of Core Services as well as some User level Services. Middleware also offers the means to access information about various Grid resources [20]. Thus, Grid middleware works like a backbone of the Grid scheduling environment. Globus [35], Legion [36] and UNICORE [37] are some examples of Grid middlewares. The middleware architecture is shown in Fig. 4.

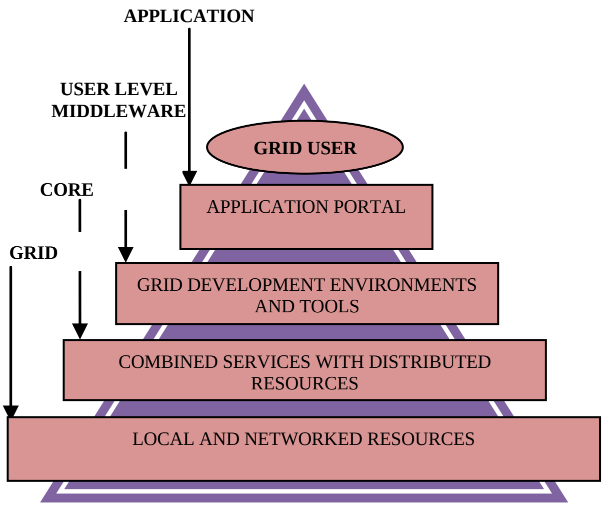

APPLICATION
USER LEVEL MIDDLEWARE
CORE
GRID
GRID USER
APPLICATION PORTAL
GRID DEVELOPMENT ENVIRONMENTS AND TOOLS
COMBINED SERVICES WITH DISTRIBUTED RESOURCES
LOCAL AND NETWORKED RESOURCES

Fig 4: Architecture of Grid Middleware

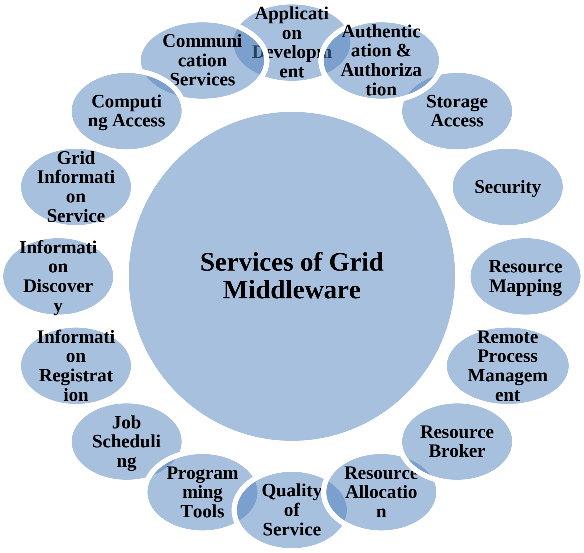


Fig 5: Services of Grid Middleware

Services of Grid Middleware:

The Grid middleware provides various types of services [20] designed in response to several challenges originating from the inherent characteristics of the Grid system [38]. These services create a seamless computing platform for the Grid users. Services of Grid Middleware are shown in Fig. 5.

E. Grid Tools:

This section discusses the tools in the aspect of Grid scheduling rather than overall services provided by Grid middleware [17]. Grid tools are classified into two parts: (i) Scheduling tools and (ii) Simulation tools, described as follows:

(i) Scheduling Tools:

Scheduling of applications on global resources is one of the major User-level services provided by Grid middleware. Some of the tools that have been developed to perform Grid scheduling includes Condor-G [39], GRaDS [40], Legion [41] etc. The tools used for scheduling is shown in Table II.

(ii)Simulation Tools:

In any research work, the design and development of a system involves constant evaluation of essential procedures for correcting the accuracy and efficiency of the proposed system before it is actually realized for use in real time environment. This also reduces the overall cost of developing and deploying the actual system. The tools used to simulate Grid environment for job scheduling is described in Table III.

V. STATE-OF-THE-ART CLASSIFICATION OF SCHEDULING HEURISTICS

The main objective of Grid computing is to make the user available transparent and efficient access to geographically distributed and remote resources that may be under multiple administrative domains. Resource access and job mapping are two important aspects of Grid scheduling. To enhance the functional capability of the Grid system, a suitable scheduling strategy is required for coordinating the access and mapping of jobs to different resources. The purpose of scheduling Heuristic is to make the best mapping, i.e. allocating jobs to the resources in the most appropriate way [54, 55]. Heuristics may not try to optimize but rather find an approximate and effective solution using a combination of quantitative and qualitative approach. To achieve an optimal solution in a dynamic environment like Grid is computationally infeasible and also known to be NP-Complete. For this reason, suboptimal Heuristics are the better options to be applied to solve scheduling problems in Grid environment [56, 57, 58].

The fast emergence of Grid computing motivated many eminent researchers to offer a variety of solutions for dealing with the Grid scheduling problem. As a result many Grid scheduling Heuristics have been reported in recent past using Conventional and Unconventional methods.

Table 2: List of Scheduling Tools

Name of Scheduling Tools	Reference	Discussion
Condor-G [39]	http://www.cs.wisc.edu/condor	To compute intensive jobs, It works as a specialized workload managing system. It manages a cluster formed by dedicated computing nodes.
GRaDS [40]	http://www.hipersoft.rice.edu/grads/	It is a Grid Application Development Software (Grads). It provides launching, scheduling and monitor performance of tightly coupled grid applications.
Legion [36]	http://legion.virginia.edu	It is a software project for object-based and meta-systems.
NetSolve/Grid Solve [41]	http://icl.cs.utk.edu/netsolve	It is a RPC based system used to perform computations with heterogeneous and geographically distributed resources connected through network.
Nimrod-G [42]	http://messagelab.monash.edu.au/NimrodG	It uses heterogeneous resources in the environment of grid. It is also used to execute applications, such as parameter sweep and automated modeling.
Sun Grid Engine [43]	http://web.njit.edu/all_topics/HPC/sge.html	It is used for dispatching, scheduling and for accepting parallel jobs in high performance computing environment like clusters. It is also used for scheduling of allocated distributed resources like memory, processors etc.
GAT [44]	http://www.Gridlab.org/grms	GAT (Grid Application Toolkit) provides a simple and unified programming interface for grid infrastructure.
Globus [35]	http://www.globus.org	It is an open-source toolkit used for creation of computer networks.
Unicore [37]	http://www.unicore.eu	Unicore (Uniform Interface to Computing Resources) is an ingredient of the European Middleware Initiative. It provides a Uniform Interface for computing resources.
EGI-InSPIRE [45]	http://www.egi.eu/about/egi-inspire//	It is designed to perform computation and join the new Distributed Computing Infrastructures (DCIs) like desktop grids, cloud, networks of supercomputing with in the area of European Research.
Cactus [46]	http://cactuscode.org	It provides an open source environment for solving problems and performs parallel computation with heterogeneous architectures.
Alchemi [47]	http://www.cloudbus.org/~alchemi/	It provides a simple framework for grid construction (desktop grid) and software development without sacrificing flexibility, scalability and reliability.

Table 3: List of Simulation Tools

Name of Grid Tool	Reference	Description
SimGrid [48]	http://simgrid.gforge.inria.fr/	It performs simulation with distributed as well as parallel applications in heterogeneous environment of distributed computing and also in computational grid.
GridSim [49]	http://www.buyya.com/gridsim/	It is a java based discreet event grid simulation toolkit used for simulation and modelling in the systems of parallel and distributed computing. It can be used for modelling of applications, users, resources, resource broker and evaluation of scheduling algorithms.
OptorSim [50]	http://www.gridpp.ac.uk/demos/optorsimapplet/	It is used to test dynamic approaches for replication and optimal scheduling. It also supports peer-to-peer messaging system.
Bricks [51]	http://ninf.is.titech.ac.jp/bricks/	The tool is based on java and is used to simulate scheduling heuristics, schedule programming modules, processing schemes made for servers and networks of high performance systems.
GridLab-D [52]	http://www.gridlabd.org/	It provides a flexible environment for scheduling with third-party tools for analysis and management of data.
Condor [53]	http://research.cs.wisc.edu/htcondor/	It supports High Throughput Computing (HTC) for huge collections of heterogeneous and distributed computing resources.

Unfortunately, this development has led to difficulty in evaluating and keeping track of those solutions, which again creates the confusion in identifying the exact methodologies applied to solve scheduling problems. Thus, overall scheduling Heuristics must be classified into some or other specialized areas so as to distinguish them in turn by their characteristics and applied methodologies.

A. Recent Works:

In recent years, a few classifications have been made in the field of Grid scheduling Heuristics. In 2013, Bharti Arora [59] et al. give a classification that mainly focuses on the characteristics, merits and demerits of scheduling Heuristics. Heuristics approaches in that paper are classified into Local based, Population based and Hybrid approaches. In 2012, the authors Ashish Chandak [60] et al. present a classification of task scheduling Heuristic in Grid computing environment that mainly focuses on the characteristics of scheduling strategies. This classification categorizes scheduling strategies into three different types named as Economic, Iterative and Other Heuristic. In 2008, Dalibor Klusacek [4], in his thesis presented a taxonomy of general Grid scheduling strategies. The taxonomy mainly comprises of two parts, Local Grid scheduling and Global Grid scheduling, based on the nature of jobs. In 2000, A. Abraham [61] et al. presented a broad survey on Economic models to manage resources in Grid computing environment. Their main focus of categorization is on the concept of “to sell goods and services in a market place of Grid”. Their classification involves Economic models such as Commodity Market Model, Posted Price Models, Auction model etc. In 1988, Casavant [62] et al. reported the taxonomy for scheduling approaches for parallel and distributed computing systems including Grid scheduling algorithms. The author has identified two types of scheduling algorithms namely Local scheduling algorithms and Global scheduling algorithms. The main focus of this paper is to investigate and identify important Objectives, Type of Job Scheduling, Functionality of Grid, Nature of Grid and QoS constraints used by most scheduling Heuristics and also report a comprehensive state-of-the-art classification of various scheduling Heuristics.

B. Job Scheduling in Grid Computing:

The Classification: After going through various existing scheduling Heuristics in Grid computing environment, it is observed that the available solutions to scheduling problem can be broadly classified into Conventional and Unconventional Scheduling Heuristics. A summarized classification is shown in Fig. 28. It is also observed that earlier solutions to Grid scheduling problem is not only inspired from Conventional strategies designed for traditional systems but also there exists many solutions that use Unconventional approaches. Fig. 6 depicts the initial classification framework.

Conventional approach is further classified into Standard Conventional, Modified Conventional and QoS based Conventional. On the other hand, Unconventional is classified into Economic and Soft Computing based Heuristics. Each sub-classification is further sub-divided into some sub-categories as given in the following sections. Each scheduling Heuristics under a sub-classification is studied from different points of view on the basis of Authors & Year, Heuristic, Objectives, Type of Job, Functionality of Grid, Nature of Grid and QoS constraints.

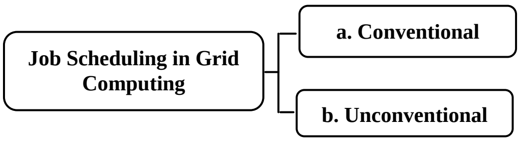


Fig 6: Classification of Scheduling Heuristics in Grid Computing

a. Conventional Scheduling Heuristics:

The concept of scheduling is a well studied subject matter related to traditional parallel and distributed computing. In Grid computing, the well-accepted traditional scheduling schemes are used in myriad ways. Generally, the matrices used in Conventional Heuristics, under the assumptions of an underlying traditional system are not suitable in a real Grid environment. However, the research and developments on scheduling problems for traditional systems still provide the stepping-stones to solve the new generation scheduling problems [63], such as Grid. The Conventional Scheduling Heuristics are also known as traditional or classical scheduling Heuristics. Usually, Conventional approaches are based on Optimization Techniques. These techniques are slow and lead to global convergence for small problems. The Conventional Scheduling Heuristics are divided into three categories as illustrated in Fig. 7. One is based on standard strategies, very well known as classical scheduling Heuristics such as FCFS [64], DSJF [65], HTSA[66], FT Min-Min[67] etc. Second is Modified Conventional, which is based on some modification done in standard scheduling approaches such as modified FCFS [68], modified MCT [69] etc. And the last one is QoS based conventional, which is Standard Conventional Heuristics along with Quality of Service as required by the user. QoS is the measure of the user satisfaction for the Grid services [70]. QoS is specified by a set of conditions and may be different for different users to run their applications successfully. Hence, different users may have different QoS requirements. It is expressed by means of qualitative measures like completion time, latency, cost, bandwidth, processing speed, throughput and reliability etc. Triangle module operator based QoS [71] and User Preference Driven Approach [72] etc. are some of the examples.

i) *Standard Conventional Scheduling Heuristics:* Standard Conventional Heuristics are of two types, one is Single objective and other is Multiobjective.

Single Objective Standard Conventional: It is the collection of those Conventional Scheduling Heuristics which are having one objective function, a flow diagram of which is shown in Fig.11. These are some earlier developed Heuristics for traditional systems, as listed in Table IV.

Multi Objective Standard Conventional: It contains those Heuristics which are having more than one objective functions, as depicted in Fig.12. In these Heuristics multiple objectives or criteria's are considered. These Heuristics are shown in Table V.

ii) *Modified Conventional Scheduling Heuristics:* It consists of those Heuristics which are modified version of some Standard Conventional Scheduling Heuristics, as given in Table VI. Its working is represented in Fig.13 with the help of a flow diagram.

iii) *QoS based Conventional Scheduling Heuristics:* These are Standard and Modified Conventional Scheduling Heuristics that use one or more QoS criteria's as shown in Fig.14 and depicted through a flow diagram. These Heuristics are gathered into the Table VII.

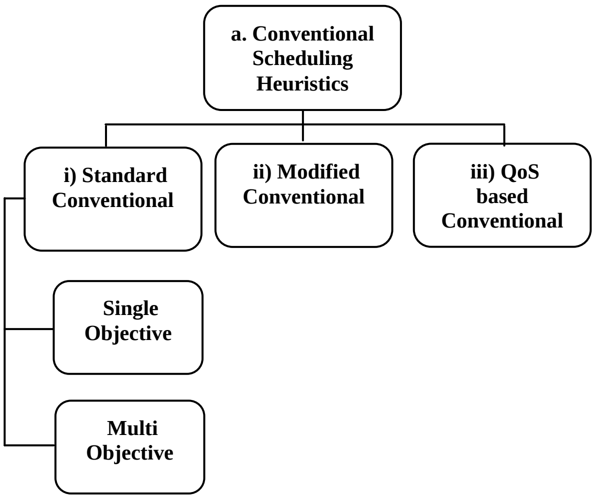

a. Conventional Scheduling Heuristics
i) Standard Conventional
ii) Modified Conventional
iii) QoS based Conventional
Single Objective
Multi Objective

Fig 7: Classification of Conventional Scheduling Heuristics

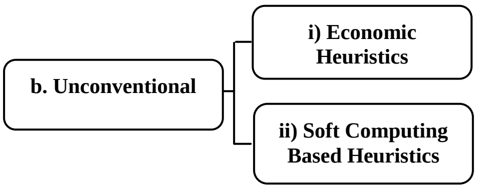


Fig 8: Classification of Unconventional Scheduling Heuristics

b. Unconventional Scheduling Heuristics:

Conventional Heuristics have limited scope and provide a poor scheduling choice to be used in Grid computing environment. Unconventional techniques commonly use some methods such as economic or approximation methods. These methods provide fast and frugal way of decision making without giving any guarantee for optimal solutions. They are broadly classified as Economic Heuristic and Soft Computing based Heuristic, as represented in Fig. 8.

i) Economic Scheduling Heuristics:

If inspected carefully, it can be observed that Grid computing is analogous to the real-world market because both are competitive, dynamic and decentralized systems, where both consumers and producers create a win-win environment to fulfil their respective objectives. Usually, a market consists of three basic components like producers, consumers and commodities. Similarly, Grid consists of resource owners, users and various computing resources. Based on this, various types of models for Economic Heuristics are introduced in order to optimize the resource management and scheduling problems in Grid computing environment [73]. Models for Economic Heuristics are not only meant for profit but also useful in designing new scheduling strategies with traditional objectives. Generally, Economic models support independent job scheduling, but more sophisticated economic models might be required for dependent job scheduling problems [3].

In the market based Grid, Time (minimizing completion time, makespan etc.), Price, or both Time and Price and QoS parameters are the instruments which creates a competitive market and heavily affects the decision making process of consumers and providers. Based on these instruments Economic models can further be classified as Time based, Price based Both Time and Price based and QoS based heuristics, as depicted in Fig. 9. The details of which are given in the Tables VIII, IX, X, XI respectively. The process involved in Economic Heuristics is also shown in the form of Flow diagrams in Fig. 15, 16, 17, 18 respectively.

ii) Soft Computing Based Scheduling Heuristics:

Conventional optimization Heuristics have a limited use and fail to provide appropriate solutions in a high-dimensional search space. In these problems, the search space grows exponentially with the problem size and therefore exhaustive search is not a practical solution. Hence, to solve these types of optimization problems, there have been some extensive uses of Unconventional approaches like Soft Computing approaches that make a little or no assumptions for a problem and can search for candidate solutions in a very large search space [86]. Soft computing based Heuristics are faster than Conventional Heuristics. By considering some general or specific characteristics, like computational effort, hybridization of Heuristics and QoS attributes, these are broadly classified into Generic Heuristics, Metaheuristic, Combined Heuristics and QoS based Heuristics as depicted in Fig. 10.

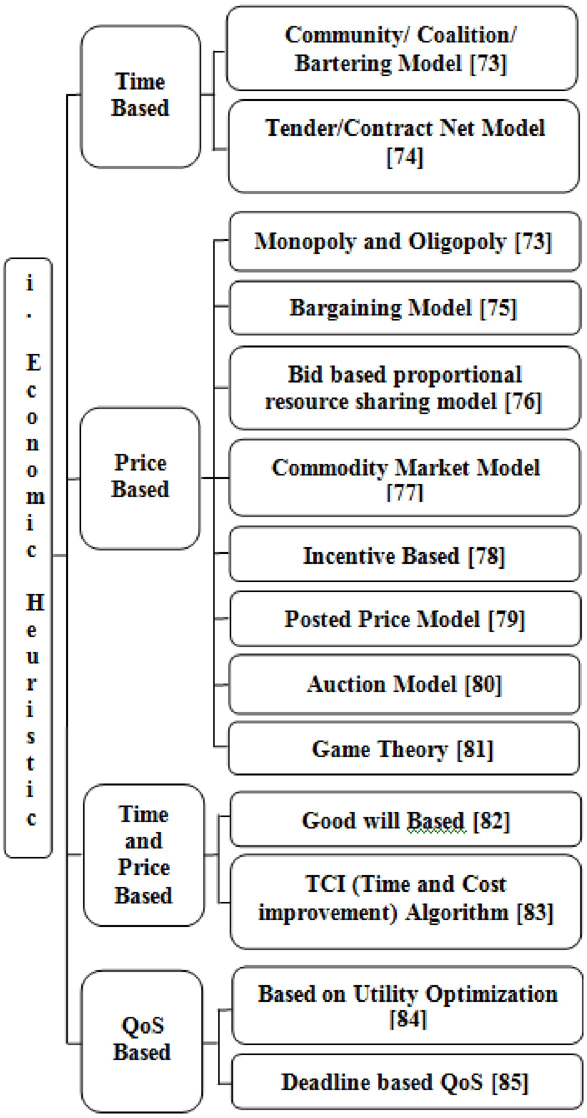

i. Economic Heuristic
Time Based
Community/ Coalition/ Bartering Model [73]
Tender/Contract Net Model [74]
Price Based
Monopoly and Oligopoly [73]
Bargaining Model [75]
Bid based proportional resource sharing model [76]
Commodity Market Model [77]
Incentive Based [78]
Posted Price Model [79]
Auction Model [80]
Game Theory [81]
Time and Price Based
Good will Based [82]
TCI (Time and Cost improvement) Algorithm [83]
QoS Based
Based on Utility Optimization [84]
Deadline based QoS [85]

Fig 9: Classification of Economic Heuristics

Generic heuristic: It is based on some Generic or specific characteristics of Soft Computing Heuristics like Fuzzy models, Neural network models etc. Some Generic Heuristic methods are Triangular pyramid scheduling [87], Fuzzy logic model [88], Neural network approach [89] and Compensation based approach [90] etc. Generic Heuristics are listed in Table XII and it process is shown in Fig.19 through a flow diagram.

Meta heuristic: It is another branch of Soft Computing based approach and derived from two Greek words that are “meta” means “higher level” and “heuristics” is from heuriskein means “to find” [91]. According to Laporte and Osman Metaheuristics is defined as “An iterative process, which intelligently combines different concepts of exploring and exploiting the search space, and uses learning strategies to structure the collected information to find near-optimal solutions in efficient manner”[92].

Metaheuristics have been successfully used to solve scheduling problems in Grid environment. Depending on number of candidate solutions, inspiration from some successful characteristic of nature and strategy of searching, Metaheuristics can be splitted into three different sub branches such as Population based, Nature inspired and Trajectory based Heuristics respectively.

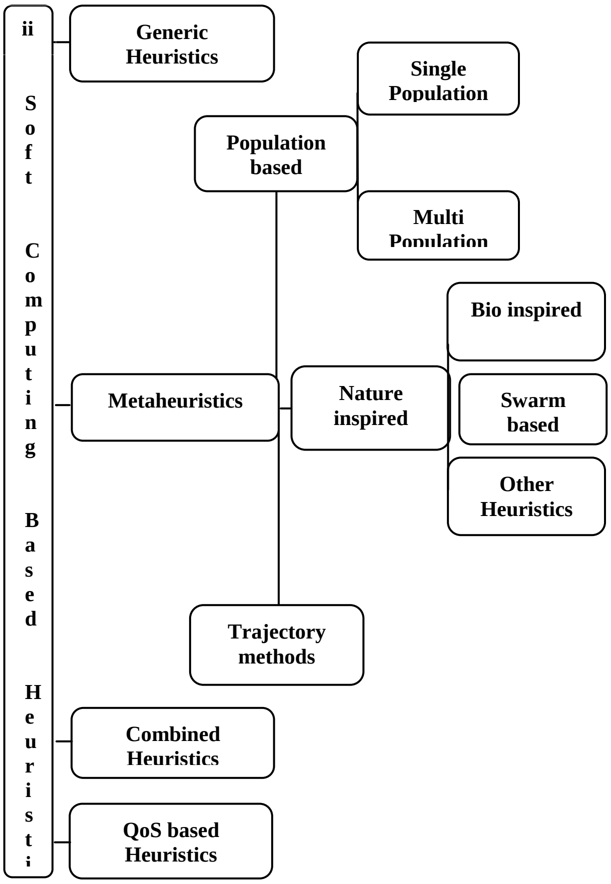


Fig 10: Classification of Soft Computing Heuristics

Population based Heuristics: It belongs to a large family of techniques to efficiently solve combinatorial optimization problems. It optimizes problem by taking population of individuals i.e. at a time it deals with more than one solution. It performs searching with multiple numbers of initial points. Again Population based Heuristics is of Single population and Multi population based as provided in Table XIII and Table XIV respectively. Single population deals with an individual population like Genetic [93] and Memetic [94] algorithms. Its process is shown in Fig. 20 through the flow diagram while in Multi-population, multiple individuals are optimized in parallel like Hierarchical Genetic Strategy [95] and Hierarchical Parallel Genetic Algorithm etc. [96] [97]. Its working diagram is represented in form of flow diagram in Fig. 21. Genetic algorithms are inspired from biological evolution processes and natural genetics. It is widely used for evolutionary systems and uses 3 basic operators i.e. selection, mutation and crossover. Based on the objective functions it can be of two kinds Single and Multi objective Genetic Algorithms as discussed in Table XV. and Table XVI. respectively.

Nature Inspired Heuristics: The work by Iztok Fister Jr. et al. [98] has presented a review on Nature-Inspired Algorithms for optimization problems. These are inspired from characteristics of biological system, swarm intelligence, physics, chemistry and music based systems. Based on their characteristics Nature Inspired Heuristics are of three types: Bio Inspired Heuristics, Swarm based Heuristics and Other Heuristics. Bio Inspired Heuristics are derived from some successful characteristics of biological system such as Biogeography based [99], Queen Bee evolution [100] etc. Its common process is shown in Fig. 22. through a flow diagram. Similarly, Swarm based Heuristics are more popular and concerned with self organizing and collective behaviour of multiple interacting agents like Artificial Bee Colony [101], Cuckoo Search [102] etc. Its process can be shown through a flow diagram as given in Fig. 23. Apart from Bio-inspired and Swarm based heuristics there are some Other Heuristics which are neither Bio-based nor Swarm based they are simply inspired from physics, chemistry and music based systems like Gravitational search [103], Simulated annealing [104] and Harmony search [105] etc. Its process is represented in Fig. 24. Nature inspired heuristics are recorded in Table XVII, XVIII, XIX respectively.

Trajectory Methods: It belongs to Metaheuristics and based on the perspective of local search. It uses a single agent or one solution at a time, which will trace out a trajectory in a search space as the iterations continue [86]. These methods are also known as popular search methods and its process is depicted in Fig. 25. Some trajectory methods are Tabu Search [106], Hill Climbing [107] etc. as documented in Table XX.

Combined Heuristic: As the name suggests, it is a combination of more than one Heuristics strategy mentioned above. In some cases, it is shown to be efficient for several problems by outperforming any such strategies that uses only one scheduling strategy [108]. Combined Heuristics are also known to be hybrid Heuristics [109] and its working is shown through a flow diagram given in Fig.26. Some combined Heuristics are Genetic Algorithm + Gravitational Search [110], Genetic algorithm + Gravity Algorithm [111], Genetic algorithm + Local Search [112] etc. as shown in Table XXI.

QoS based heuristic: In general, quality is a non-functional part of a Grid computing. It exhibits some attributes such as performance, security, reliability and cost etc. or any combination of these. In the classification of Soft Computing based Heuristics, QoS Heuristics are those which uses Soft Computing methods along with some QoS constraints for example ICA (Imperialist Competitive Algorithm) + QoS [113], Max-Min PSO [114], Local Search Based approach using SA [115] etc. as depicted in Table XXII. Its flow diagram is shown in Fig. 27.

VI. A FAIR ASSESSMENT OF GRID ENVIRONMENTAL AND PERFORMANCE PARAMETERS OF SCHEDULING HEURISTICS UNDER STUDY

This section introduces a fair evaluation of Grid environmental and performance parameters used in Heuristics that are included in the proposed classification. The various objective functions used in Heuristics are collected and the total frequency of each objective is recorded and analyzed. Similarly, an evaluation on QoS constraints is performed. The observations made for Grid environmental parameters used in Heuristics under each classification are given in Table XXIII. A summarized observation of frequently used Grid environmental parameters (all values are in percentage) such as Type of Job Scheduling, Functionality of Grid and Nature of Grid for each category of scheduling Heuristics is shown in Table XXIV. The overall observations made in Table XXIII, XXIV are shown in the form of pie charts as depicted in Fig. 29, 30, 31, 32, 33, 34, 35 and 36 respectively.

VII. CONCLUSIONS

The distinct characteristics of Grid Computing make the design of scheduling system more challenging than traditional distributed scheduling system. This is an effort made to look into various issues that stand in the way of Grid scheduling problem and reported a structured representation of various issues related to Grid scheduling. The survey mainly focuses on three parts, (i) the major issues influencing scheduling problem in Grid computing platform, and (ii) a state-of-the-art classification of scheduling Heuristics and (iii) An assessment of various parameters used in scheduling Heuristics. An effective and efficient scheduling system is essentially important to realize the desired performance of Grid. To make this happen, an appropriate process must be followed i.e. (i) Collecting the resource state information (Resource Discovery) (ii) Selecting suitable resources (Resource Filtering and Selection) and (iii) Scheduling Policy (Job allocation and Job submission). Each of the above steps contributes to achieve the desired performance level of Grid system. To show the summary of entire study, the following important parameters are identified and listed in the Table XXV.

The widely used Heuristics identified in this study are Soft Computing and Multiobjective Standard Conventional Scheduling Heuristics. The most frequently used objective functions are Minimize Makespan, Minimize Cost, Maximize Resource Utilization, Maximize Load balancing and Minimize Flowtime. In the case of QoS constraints Deadline, Budget and Reliability are mostly used constraints. The other frequently used factors identified in this study are the use of Independent jobs, Compute Intensive as functionality of Grid and Dynamic as Nature of Grid.

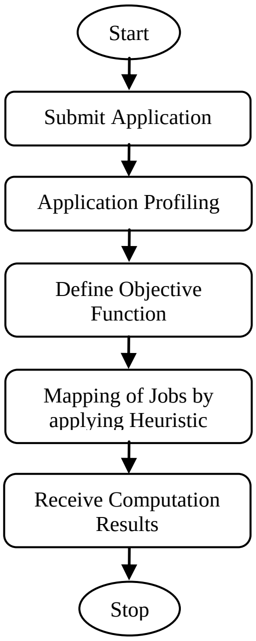

Start
Submit Application
Application Profiling
Define Objective Function
Mapping of Jobs by applying Heuristic
Receive Computation Results
Stop

Fig 11: Standard Conventional Scheduling Heuristics: Single Objective

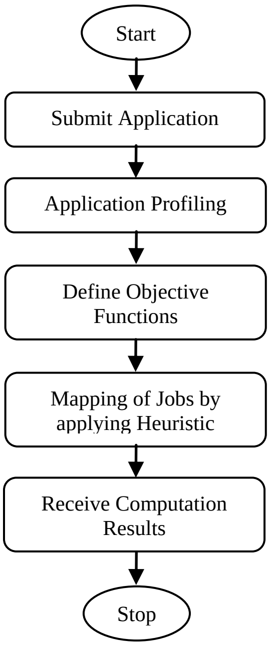

Start
Submit Application
Application Profiling
Define Objective Functions
Mapping of Jobs by applying Heuristic
Receive Computation Results
Stop

Fig 12: Standard Conventional Scheduling Heuristics: Multi Objective

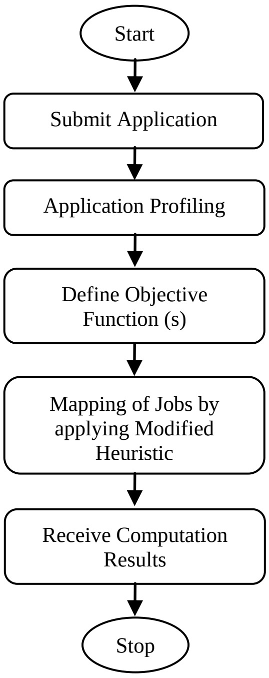

Start
Submit Application
Application Profiling
Define Objective Function (s)
Mapping of Jobs by applying Modified Heuristic
Receive Computation Results
Stop

Fig 13: Standard Conventional Scheduling Heuristics: Modified Conventional

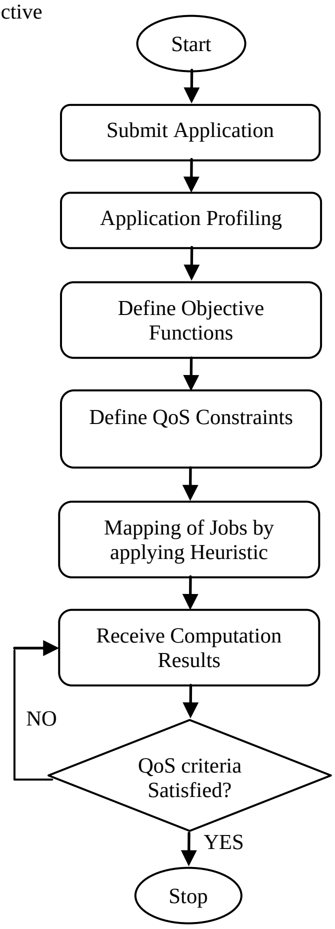


Fig 14: Standard Conventional Scheduling Heuristics: QoS based conventional

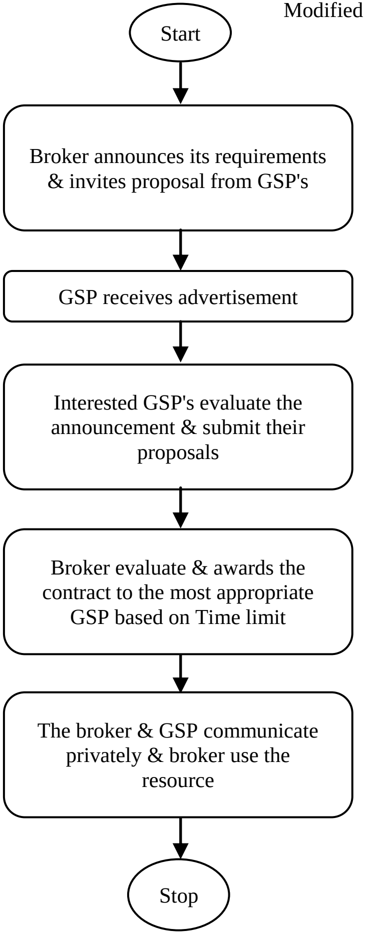

Start
Broker announces its requirements & invites proposal from GSP's
GSP receives advertisement
Interested GSP's evaluate the announcement & submit their proposals
Broker evaluate & awards the contract to the most appropriate GSP based on Time limit
The broker & GSP communicate privately & broker use the resource
Stop

Fig 15: Economic Scheduling Heuristics: Time based

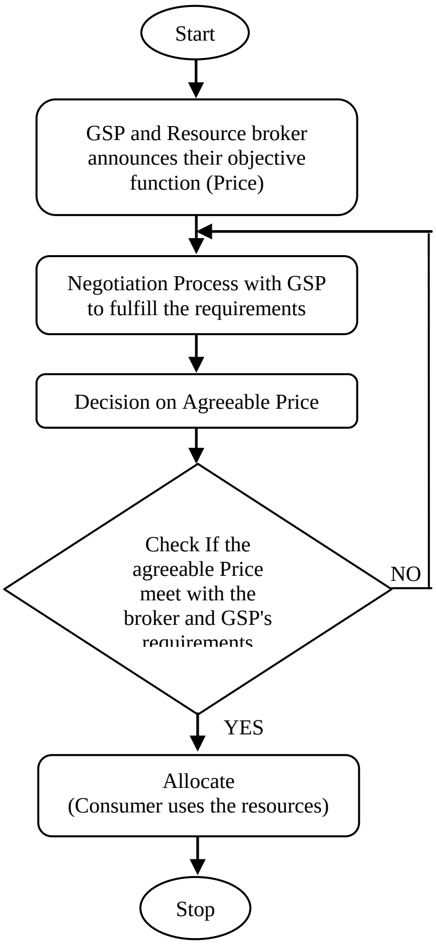

Start
GSP and Resource broker announces their objective function (Price)
Negotiation Process with GSP to fulfill the requirements
Decision on Agreeable Price
Check If the agreeable Price meet with the broker and GSP's requirements
NO
YES
Allocate (Consumer uses the resources)
Stop

Fig 16: Economic Scheduling Heuristics: Price based

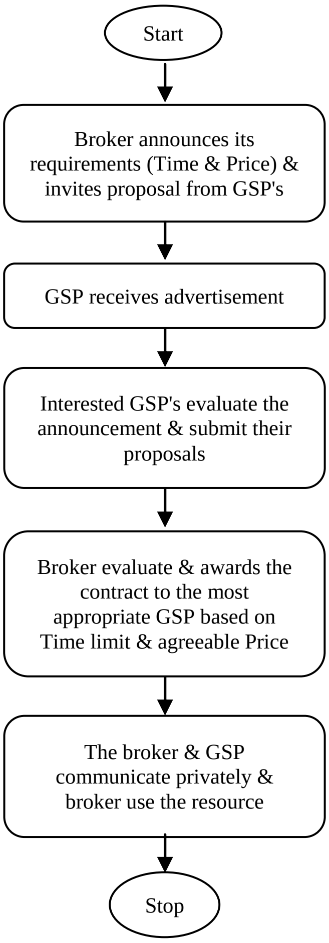

Start
Broker announces its requirements (Time & Price) & invites proposal from GSP's
GSP receives advertisement
Interested GSP's evaluate the announcement & submit their proposals
Broker evaluate & awards the contract to the most appropriate GSP based on Time limit & agreeable Price
The broker & GSP communicate privately & broker use the resource
Stop

Fig 17: Economic Scheduling Heuristics: Time & Price based

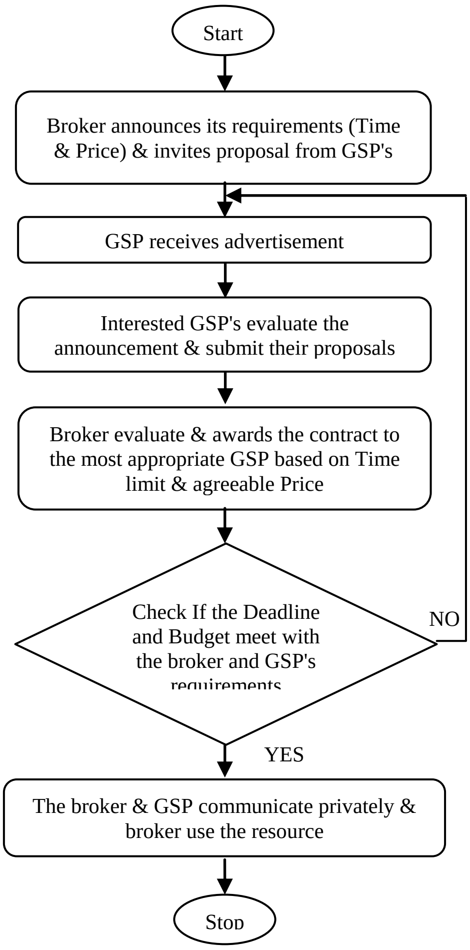

Start
Broker announces its requirements (Time & Price) & invites proposal from GSP's
GSP receives advertisement
Interested GSP's evaluate the announcement & submit their proposals
Broker evaluate & awards the contract to the most appropriate GSP based on Time limit & agreeable Price
Check If the Deadline and Budget meet with the broker and GSP's requirements
NO
YES
The broker & GSP communicate privately & broker use the resource
Stop

Fig 18: Economic Scheduling Heuristics: QoS based

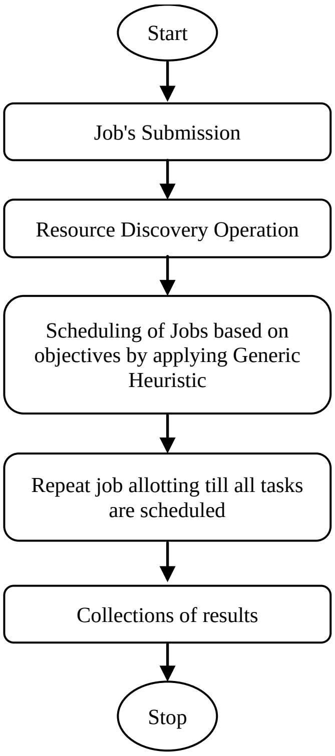

Start
Job's Submission
Resource Discovery Operation
Scheduling of Jobs based on objectives by applying Generic Heuristic
Repeat job allotting till all tasks are scheduled
Collections of results
Stop

Fig 19: Soft Computing based Heuristics: Generic

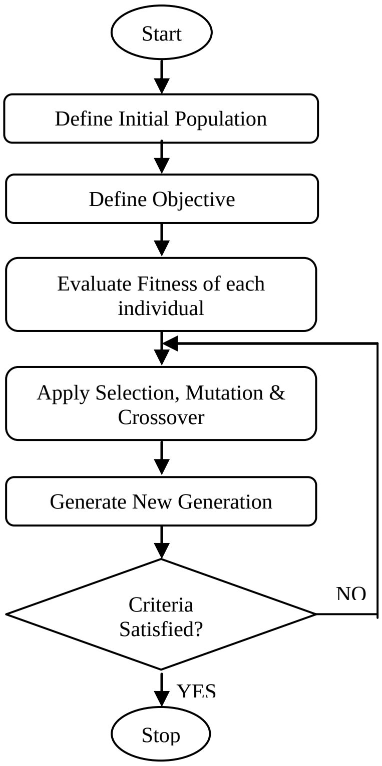

Start
Define Initial Population
Define Objective
Evaluate Fitness of each individual
Apply Selection, Mutation & Crossover
Generate New Generation
Criteria Satisfied?
NO
YES
Stop

Fig 20: Population based Heuristics: Single Population

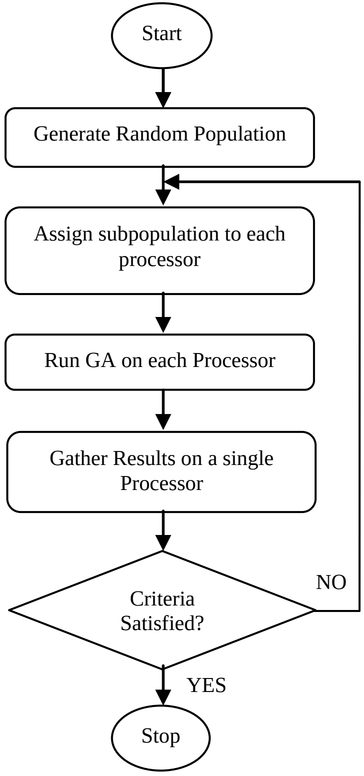

Start
Generate Random Population
Assign subpopulation to each processor
Run GA on each Processor
Gather Results on a single Processor
Criteria Satisfied?
NO
YES
Stop

Fig 21: Population based Heuristics:

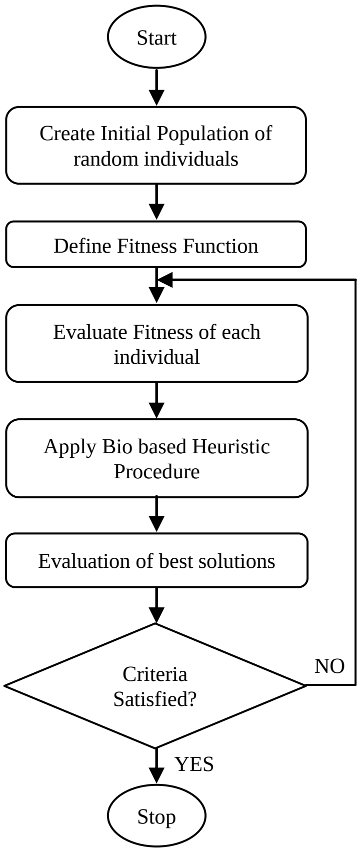

Start
Create Initial Population of random individuals
Define Fitness Function
Evaluate Fitness of each individual
Apply Bio based Heuristic Procedure
Evaluation of best solutions
Criteria Satisfied?
NO
YES
Stop

Fig 22: Nature Inspired Heuristics: Bio based

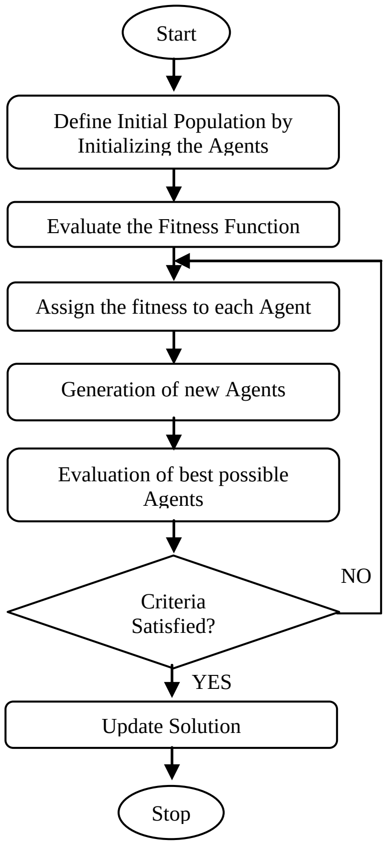

Start
Define Initial Population by Initializing the Agents
Evaluate the Fitness Function
Assign the fitness to each Agent
Generation of new Agents
Evaluation of best possible Agents
Criteria Satisfied?
NO
YES
Update Solution
Stop

Fig 23: Nature Inspired Heuristics: Swarm based

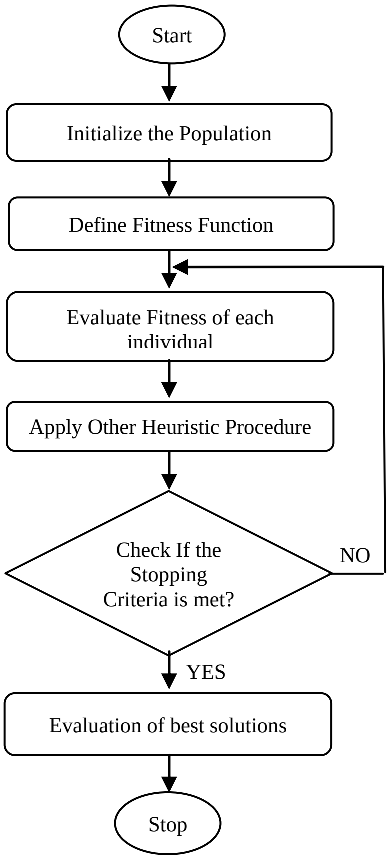

Start
Initialize the Population
Define Fitness Function
Evaluate Fitness of each individual
Apply Other Heuristic Procedure
Check If the Stopping Criteria is met?
NO
YES
Evaluation of best solutions
Stop

Fig 24: Nature Inspired Heuristics: Other

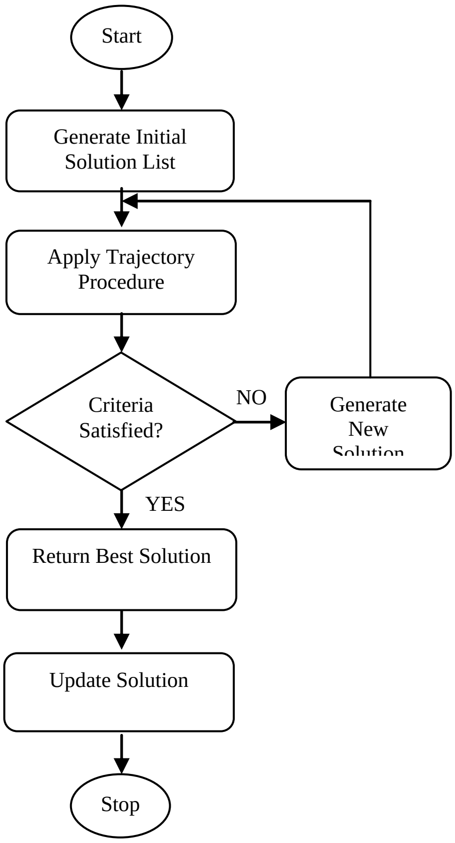


Fig 25: Trajectory Methods

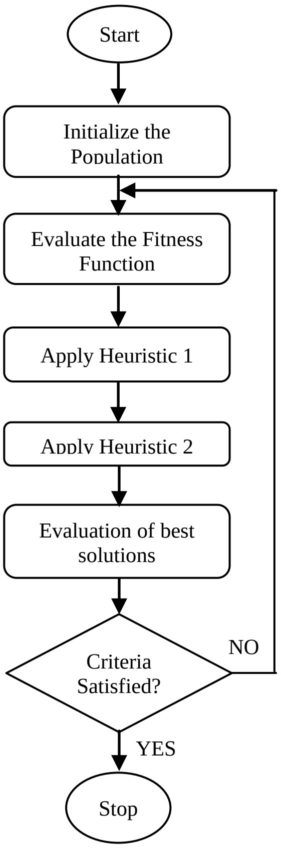

Start
Initialize the Population
Evaluate the Fitness Function
Apply Heuristic 1
Apply Heuristic 2
Evaluation of best solutions
Criteria Satisfied?
NO
YES
Stop

Fig 26: Soft Computing based Heuristics: Combined

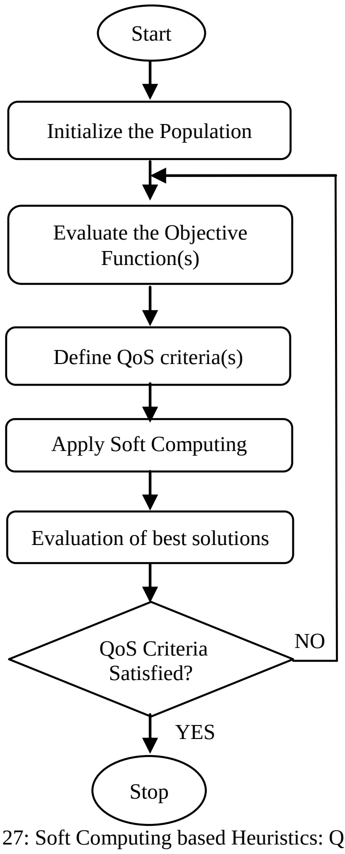

Start
Initialize the Population
Evaluate the Objective Function(s)
Define QoS criteria(s)
Apply Soft Computing
Evaluation of best solutions
QoS Criteria Satisfied?
NO
YES
Stop

Fig 27: Soft Computing based Heuristics: QoS based

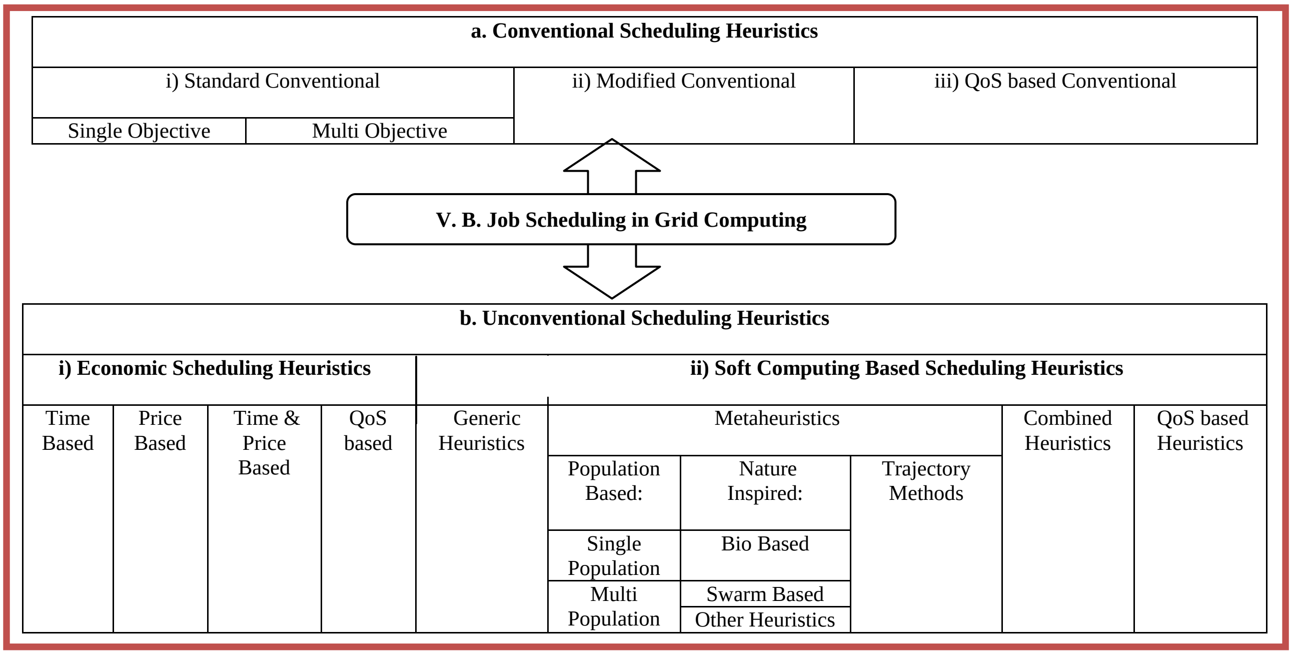


Fig 28: Overall view of the proposed Classification for Scheduling Heuristics

Table 4: Conventional Scheduling Heuristics: Single Objective Standard Conventional

Reference Number of Heuristics	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[64], [65], [116], [56], [117], [118]	Minimize Completion Time/ Minimize Makespan	Independent	Compute Intensive	Dynamic
[65]	Minimize Makespan	Independent	Compute Intensive	Static

Table 5: Conventional Scheduling Heuristics: Multi Objective Standard Conventional

Reference Number of Heuristics	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[66], [123]	Minimize Makespan, Maximize Resource Utilization	Independent	Compute Intensive	Static
[67]	High hit rate, Minimize Makespan	Independent	Compute Intensive	Static
[119], [120], [130], [131], [132], [138], [140], [141]	Minimize Makespan	Independent	Compute Intensive	Dynamic
[121], [124]	Minimizes Makespan, Load balancing	Independent	Compute Intensive	Static
[122]	Minimize Average waiting time, Average Turnaround time and Response time	Independent	Compute Intensive	Dynamic
[125], [133]	Minimize Makespan	Independent	Compute Intensive	Static
[126]	Minimize Cost	Independent	Compute Intensive	Dynamic
[127]	Minimize Makespan, Minimize Cost	Independent	Compute Intensive	Dynamic
[128]	N-independent criteria	Dependent	Compute Intensive	Dynamic
[129]	Minimize Makespan, Maximize Load balancing	Independent	Compute Intensive	Static & Dynamic
[134]	Minimize Execution Time, Load Balancing	Dependent	Compute Intensive	Dynamic
[135]	Minimize Turnaround time, Maximize CPU utilization and throughput	Independent	Compute Intensive	Dynamic
[136]	Execution Time	Independent	Compute Intensive	Static & Dynamic
[137]	Minimize Total Processing Time	Independent	Service Intensive	Dynamic
[139]	Minimize Makespan and Flowtime	Independent	Compute Intensive	Dynamic
[142]	Minimize Weighted Completion Time	Independent	Compute Intensive	Dynamic
[198]	Maximize resource utilization, Minimize Processing Time	Independent	Compute Intensive	Dynamic
[200]	Minimize Waiting time, Turnaround time , Average response time, Average total completion times	Independent	Compute Intensive	Dynamic

Table 6: Conventional Scheduling Heuristics: Modified Conventional

Reference Number of Heuristics	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[68]	Minimize Average weighted response time, Minimize Average weighted slowdown	Independent	Compute Intensive	Dynamic
[69]	Minimize Makespan, Minimize Cost	Independent	Compute Intensive	Dynamic
[143]	Minimize Makespan, Minimize Flowtime	Independent	Compute Intensive	Dynamic
[201]	Minimizing overhead time and Computation time	Independent	Compute Intensive	Dynamic

Table 7: Conventional Scheduling Heuristics: QoS based Conventional

Reference Number of Heuristics	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[71]	Minimize Makespan, Load balancing of resources and Utility QoS of user	Independent	Compute Intensive	Dynamic
[72]	Satisfy maximum number of tasks, Minimize Makespan, Maximize Resource Utilization	Independent	Compute Intensive	Dynamic
[144]	Minimize Makespan, Maximize reliability	Independent	Compute Intensive	Dynamic
[145]	Minimize Total Execution Time, Minimize the average waiting time of the high-priority task	Independent	Compute Intensive	Dynamic
[146]	Enhance performance of system and load balancing	Independent	Compute Intensive	Static
[147]	Optimize Makespan and Load balancing	Independent	Compute Intensive	Static
[148]	Minimize the scheduling time for those jobs that are urgent	Independent	Compute Intensive	Dynamic
[149]	Minimize Makespan, Enhance Resource Utilization and load balancing of resources	Independent	Compute Intensive	Static & Dynamic
[150], [155]	Minimize Makespan, Enhance Resource Utilization and load balancing of resources	Independent	Compute Intensive	Static & Dynamic
[151]	Minimize Makespan	Independent	Compute Intensive	Static
[152]	High tasks communication and throughput of computation	Dependent	Compute Intensive	Dynamic
[153]	Minimize Turn around time	Dependent	Compute Intensive	Dynamic
[154]	Maximize system utilization	Independent	Compute Intensive	Dynamic
[156]	Minimize Makespan and Minimize Cost	Independent	Compute Intensive	Static
[157]	Improve Makespan and Task accepted Performance	Independent	Compute Intensive	Static
[158]	Minimize Execution cost	Dependent	Service Intensive	Dynamic
[159]	Minimize Makespan	Independent	Compute Intensive	Dynamic

Table 8: Economic Scheduling Heuristics: Time based

Reference Number of Heuristics	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[74]	Optimize System Load and execution cost	Independent	Compute Intensive	Dynamic
[73]	Storage Sharing	Independent	Compute Intensive	Dynamic

Table 9: Economic Scheduling Heuristics: Price based

Reference Number of Heuristics	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[75]	Maximize Profit	Independent	Data Intensive	Dynamic
[76]	Minimize Makespan	Independent	Compute Intensive	Dynamic
[77]	Resource utilization for determining price for a service.	Independent	Compute Intensive	Dynamic
[78]	Maximize the job execution success rate and Minimize the deviation of fairness among resources	Independent	Compute Intensive	Dynamic
[79]	Optimize response time and wait time minimization benefit during the GRS (Grid Resource Supermarket) construction	Independent	Compute Intensive	Dynamic
[80]	Optimize task and resource utilization	Independent	Compute Intensive	Dynamic
[81]	Minimize job execution time and maximize revenue of Grid community	Independent	Compute Intensive	Dynamic
[73]	Case of Single GSP (Grid Service Provider), which dominates the market and also a single provider for a particular service.	Independent	Compute Intensive	Dynamic

Table 10: Economic Scheduling Heuristics: Time & Price based

Reference Number of Heuristics	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[82]	Minimize average overall time and cost of parallel tasks	Independent	Compute Intensive	Dynamic
[83]	Optimize time and cost	Dependent	Compute Intensive	Static

Table 11: Economic Scheduling Heuristics: QoS based

Reference Number of Heuristics	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[84]	Maximize resource utilization and throughput of system, Cost-Time optimization of tasks	Independent	Compute Intensive	Dynamic
[85]	Reduce network overhead and number of requests during processing	Independent	Compute Intensive	Dynamic

Table 12: Soft Computing based Scheduling Heuristics: Generic Heuristics

Reference Number of Heuristics	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[87]	Minimize Makespan, Minimize congestion, Improve the efficiency of simulation and increase the efficiency of resource utilization	Independent	Compute Intensive	Dynamic
[88]	Reduce the Turn around time, Better speed-up ratio	Independent	Compute Intensive	Dynamic
[89]	Higher accuracy, Minimize Makespan/ completion time	Independent	Compute Intensive	Dynamic
[90]	Reduce execution time estimation misses, Minimize total execution Time	Independent	Compute Intensive	Dynamic

Table 13: Population based Heuristics: Single Population

Reference Number of Heuristics	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[94]	Minimize Makespan, Minimize Flowtime	Independent	Compute Intensive	Dynamic
[93]	Minimize average completion time of jobs	Independent	Service Intensive	Dynamic

Table 14: Population based Heuristics: Multi Population

Reference Number of Heuristics	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[95]	Minimize makespan and Minimize flowtime	Independent	Compute Intensive	Dynamic
[96]	Maximum Speed up	Independent	Compute Intensive	Dynamic

Table 15: Genetic Algorithm: Single Objective GA

Reference Number of Heuristics	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[160]	Minimize makespan	Independent	Compute Intensive	Dynamic

Table 16: Genetic Algorithm: Multi Objective GA

Reference Number of Heuristics	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[161]	Minimize Makespan, Minimize cost, Maximize reliability	Independent	Compute Intensive	Dynamic
[162]	Maximize load balancing, Minimize total time, Minimize cost, Minimize makespan	Independent	Compute Intensive	Dynamic
[163]	Minimize makespan, Minimize flowtime, Minimize fault index	Independent	Compute Intensive	Static
[164]	Minimize Makespan and Energy Consumption	Independent	Compute Intensive	Static and Dynamic
[165]	Minimize Makespan and Flowtime	Independent	Compute Intensive	Dynamic
[166]	Optimize Grid Task scheduling length, Total security effective value, Reliability, Cost of scheduling	Independent	Compute Intensive	Dynamic
[167]	Maximize the number of executions of workflow, minimize the variance of waiting Time among tasks of each workflow, Maximize job completion ratio (JCR), Minimize waiting time variance (WTV)	Dependent	Compute Intensive	Dynamic
[168]	Minimize the completion time, Maximize the utilization of resources	Independent	Compute Intensive	Dynamic

Table 17: Nature Inspired Heuristics: Bio Inspired

Reference Number of Heuristics	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[99]	Minimize Makespan, Maximize Reliability	Dependent	Compute Intensive	Dynamic
[100]	Minimize mean Makespan and Run-time	Independent	Compute Intensive	Static
[144], [145]	Minimize Makespan	Independent	Compute Intensive	Dynamic

Table 18: Nature Inspired Heuristics: Swarm Based

Reference Number of Heuristics	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[101]	Minimize Makespan, Optimize termination time , Reduce average waiting time, Reduce number of required processor, No memory overflow	Dependent	Compute Intensive	Dynamic
[102]	Minimize execution time and Idle time	Independent	Compute Intensive	Dynamic
[171]	Minimize the total completion time	Independent	Compute Intensive	Dynamic
[172], [174]	Minimize Makespan	Independent	Compute Intensive	Dynamic
[173]	Minimize Cost and Makespan	Independent	Compute Intensive	Dynamic
[175]	Minimize makespan and Total data file transfer time	Dependent	Data Intensive	Dynamic
[176]	Minimize Makespan and Flowtime	Independent	Compute Intensive	Dynamic
[177]	Decrease total task's completion time, cost and percentage of those tasks which are unsuccessful and to avoid resource failure problem	Independent	Compute Intensive	Dynamic
[178]	Minimize the maximal total tardiness time	Independent	Compute Intensive	Dynamic
[199]	Minimize Total Response Time, Increase Utilization	Independent	Compute Intensive	Dynamic

Table 19: Nature Inspired Heuristics: Other

Reference Number of Heuristic	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[113]	Improve cost and execution time under deadline and budget constraint	Dependent	Compute Intensive	Dynamic
[114]	Minimize makespan under cost and deadline constraints	Independent	Compute Intensive	Dynamic
[115]	Optimize makespan, Lateness and tardiness	Independent	Compute Intensive	Dynamic
[193]	Optimize QoS parameter preferred by user, QoS constraints- Deadline, Budget and Reliability	Dependent	Compute Intensive	Dynamic
[194]	Minimize makespan, Improve utilization of resource under the QoS constraints- deadline, Budget and priority of tasks	Independent	Compute Intensive	Dynamic
[195], [196]	Maximize the machine usage and jobs that satisfies the deadline.	Independent	Compute Intensive	Dynamic
[197]	Optimize time astringency	Dependent	Service Intensive	Dynamic

Table 20: Metaheuristics: Trajectory Methods

Reference Number of Heuristic	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[106]	Minimize Flowtime and Makespan	Independent	Compute Intensive	Static and Dynamic
[180]	Minimize the number of Jobs that do not satisfies deadline	Independent	Compute Intensive	Dynamic
[107]	Enhance Processor utilization, Minimize Bounded slowdown	Dependent	Compute Intensive	Dynamic
[181]	Minimize Makespan	Dependent	Data Intensive	Dynamic

Table 21: Soft Computing based Scheduling Heuristics: Combined

Reference Number of Heuristics	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[110]	Minimize Makespan and the number of tasks that are missing their deadline	Independent	Compute Intensive	Static
[111]	High Speed and Performance	Independent	Compute Intensive	Dynamic
[112]	Reduce number of generation to find best solution	Independent	Compute Intensive	Dynamic
[182]	Minimize overall cost of task executions and Makespan	Independent	Compute Intensive	Static
[183]	Minimum Response time, Maximum profit	Independent	Compute Intensive	Dynamic
[184]	Minimize Makespan and Missed tasks	Independent	Compute Intensive	Dynamic
[185]	Minimum Makespan and Flowtime	Independent	Compute Intensive	Static
[186]	Minimize Makespan and cummulative delay	Independent	Compute Intensive	Dynamic
[187]	Minimum Makespan and Flowtime	Independent	Compute Intensive	Dynamic
[188]	Improve convergence speed and accuracy of convergence	Dependent	Compute Intensive	Dynamic
[189]	Load balancing, Scalability (Expandability), Optimum Scheduling	Independent	Compute Intensive	Dynamic
[190]	Minimize the Turn around time, Optimum speed-up ration	Independent	Compute Intensive	Dynamic
[191], [192]	Higher convergence speed	Dependent	Compute Intensive	Dynamic

Table 22: Soft Computing based Scheduling Heuristics: QoS based

Reference Number of Heuristic	Objectives	Type of Job Scheduling	Functionality of Grid	Nature of Grid
[179]	Minimize makespan, Find optimal solution in less time	Independent	Compute Intensive	Dynamic
[105]	Load balancing, Minimize makespan, Increase utilization of resources, Minimize mean square deviation	Independent	Compute Intensive	Dynamic
[103]	Minimize Makespan, Minimize the number of missed tasks	Independent	Compute Intensive	Static
[104]	Minimize Makespan	Independent	Compute Intensive	Dynamic

Table 23: Observation of Grid Environmental Parameters used in Heuristics under each classification

Heuristics	**Total**	**Grid Environmental Parameters**							
		Type of Job Scheduling		**Functionality of Grid**			**Nature of Grid**		
		Dependent	**Independent**	**Compute Intensive**	**Data Intensive**	**Service Intensive**	**Static**	**Dynamic**	**Static & Dynamic**
Conventional Scheduling Heuristics									
Single Standard Conventional Objective	9	0	9	9	0	0	1	8	0
Multi Objective Standard Conventional	31	2	29	30	0	1	7	22	2
Modified Conventional	7	0	7	7	0	0	0	7	0
QoS based Conventional	18	3	15	17	0	1	6	10	2
Unconventional Scheduling Heuristics									
A. Economic Scheduling Heuristics									
Time based Economic	2	0	2	2	0	0	0	2	0
Price based Economic	8	0	8	7	1	0	0	8	0
Time & Price based Economic	2	1	1	2	0	0	1	1	0
QoS based Economic	2	0	2	2	0	0	0	2	0
B. Soft Computing Based Heuristics									
Generic	4	0	4	4	0	0	0	4	0
Single Population based	2	0	2	1	0	1	0	2	0
Multi Population based	2	0	2	2	0	0	0	2	0
Single Objective GA	1	0	1	1	0	0	0	1	0
Multi Objective GA	8	1	7	8	0	0	1	6	1
Bio Inspired	4	1	3	4	0	0	1	3	0
Swarm based	10	2	8	9	1	0	0	10	0
Other	4	0	4	4	0	0	1	3	0
Trajectory Methods	4	2	2	3	1	0	0	3	1
Combined Heuristics	14	3	11	14	0	0	3	11	0
QoS based Heuristics	8	3	5	7	0	1	0	8	0
Total	140	18	122	133	3	4	21	113	6
Percentage	**100**	**12.85**	**87.14**	**95**	**2.14**	**2.85**	**15**	**80.71**	**4.28**

Table 24: A summarized observation of frequently used Grid Environmental Parameters (All Values are in Percentage)

Heuristics		Grid Environmental Parameters							
		Types of Job Scheduling		Functionality of Grid			Nature of Grid		
		Dependent	Independent	Compute Intensive	Data Intensive	Service Intensive	Static	Dynamic	Static & Dynamic
Conventional	46.42	27.77	49.18	47.37	0.00	50.00	66.67	41.60	66.67
Unconventional	53.58	72.23	50.82	52.63	100	50.00	33.33	58.40	33.33
Conventional + Unconventional	**100**	**12.85**	**87.15**	**95**	**2.14**	**2.86**	**15**	**80.71**	**4.28**

Table 25: Conclusive Grid Environmental and Performance paramseters identified in this study

Heuristics	Objective Functions	Type of Job Scheduling	Functionality of Grid	Nature of Grid	QoS Constraints
Soft Computing	Minimize Makespan	Independent	Compute Intensive	Dynamic	Deadline
Multi Objective Standard Conventional	Minimize Cost				Reliability
	Maximize Resource Utilization				Budget
	Maximize Load Balancing				
	Minimize Flowtime				

The Pie-chart representations of the Table XXIII, XXIV

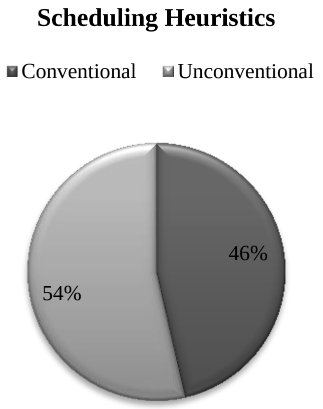


Fig 29: Pie chart representation of Scheduling Heuristics

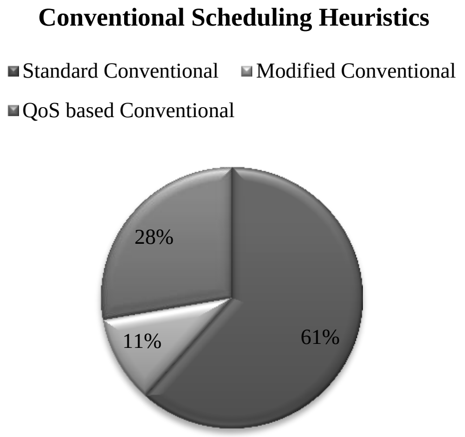
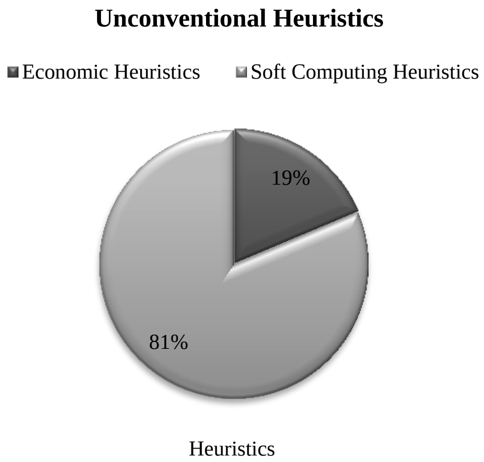


Fig 30: Pie chart representation of Conventional Scheduling Heuristics

Fig 31: Pie chart representation of Unconventional Scheduling



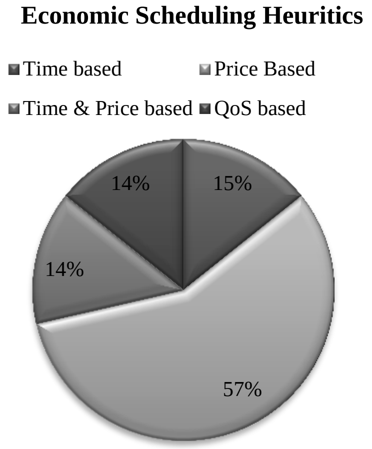


Fig 32: Pie chart representation of Economic Heuristics Classification Classification

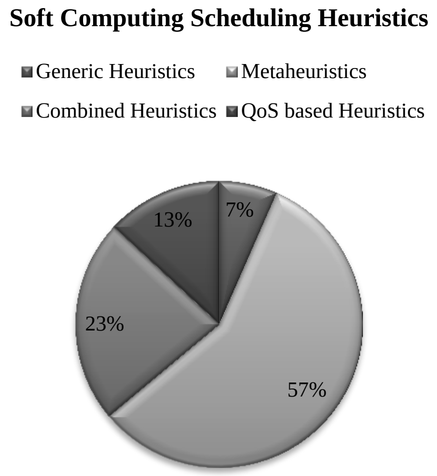


Fig 33: Pie chart representation of Soft Computing Heuristics

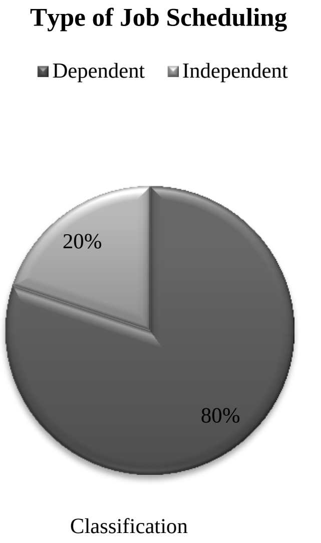


Fig 34: Pie chart representation of Types of Job Scheduling

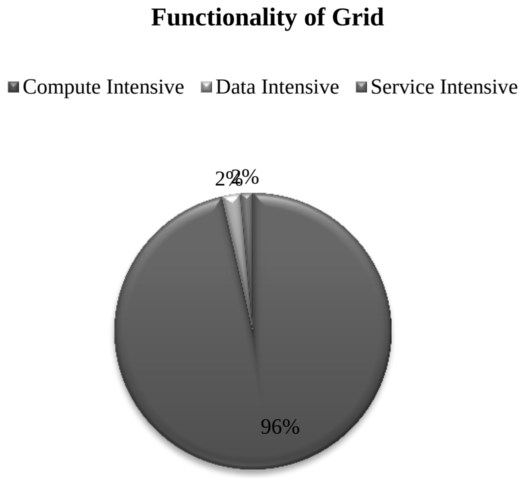


Fig 35: Pie chart representation of Functionality of Grid

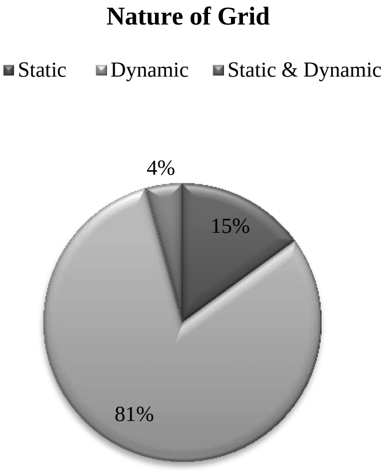


Fig 36: Pie chart representation of Nature of Grid

REFERENCES

[1] Ian Foster and Carl Kesselman. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1998.

[2] Ian Foster and Carl Kesselman. The Grid 2: Blueprint for a New Computing Infrastructure, second edition. Morgan Kaufmann, 2004.

[3] F. Dong, S.K. Akl, Scheduling algorithms for grid computing: State of the art and open problems, Technical Report No. 2006-504, School of Computing, Queen's University, Kingston, Ontario, Canada, January 2006.

[4] Dalibor Klusacek. Scheduling in Grid Environment. PhD thesis, Masaryk University, Brno, 2008.

[5] Survey On Grid Scheduling, Journal of Computer Applications, Vol. 3, No. 3, July-Sept 2010, pp. 22-29.

[6] D.I. George Amalarethinam and P. Muthulakshmi, "An Overview of the scheduling policies and algorithms in Grid Computing", International Journal of Research and Reviews in Computer Science, Vol. 2, No. 2, 2011, pp. 280-294.

[7] Thilagavathi, D., and Antony Selvadoss Thanamani. "Heuristics in Grid Scheduling." International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Vol. 2, Issue 8, August 2013.

[8] O. H. Ibarra and C. E. Kim, Heuristic algorithms for scheduling independent tasks on non identical processors, J. ACM 24, 2 (Apr. 1977), pp. 280-289.

[9] Hojjat Baghban, Amir Masoud Rahmani, " A Heuristic on Job Scheduling in Grid Computing Environment", In Proceedings of the seventh IEEE International Conference on Grid and Cooperative Computing, 2008, pp. 141-146.

[10] Li Wenzheng, Zhang Wenyue, "An Improved Scheduling Algorithm for Grid Tasks", International Symposium on Intelligent Ubiquitous Computing and Education, 2009, pp. 9-12.

[11] H. El-Rewini, T. Lewis, and H. Ali, Task Scheduling in Parallel and Distributed Systems, ISBN: 0130992356, PTR Prentice Hall, 1994.

[12] D. Fernandez-Baca. Allocating modules to processors in a distributed system. IEEE Transactions on Software Engineering, 15(11), 1989, pp. 1427–1436.

[13] Tracy D.Braun, Howard Jay Siegel, and Noah Beck, "A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous

Distributed Computing Systems", Journal of Parallel and Distributed Computing 61, 2001, pp.810-837.

[14] T.Braun, H.Siegel, N.Beck, L.Boloni, M.Maheshwaran, A.Reuther, J.Robertson, M.Theys, B.Yao, D.Hensgen, and R.Freund, "A Comparison Study of Static Mapping Heuristics for a Class of Meta-tasks on Heterogeneous Computing Systems", In 8th IEEE Heterogeneous Computing Workshop (HCW'99), 1999, pp. 15-29.

[15] M.R. Garey and D.S. Johnson. "Computers and Intractability– A Guide to the Theory of NP Completeness."W.H. Freeman and Co., 1979.

[16] Y. Zhu, A Survey on Grid Scheduling Systems, Department of Computer Science, Hong Kong University of science and Technology, 2003.

[17] Vivekanandan, K., Ramyachitra, D.: A Study on Scheduling in Grid Environment. International Journal on Computer Science and Engineering (IJSE) 3(2), 2011, pp. 940–950.

[18] Mr. Rakesh Kumar1, Navjot Kaur, "Job Scheduling in Grid Computers", IEEE, 2006.

[19] A Novel Scheduling Model for Computational Grid Economy Systems, Omer Ozan S onmez, Ko ̧c University, August 2005.

[20] P. Asadzadeh et al., "Global Grids and Software Toolkits: A Study of Four Grid Middleware Technologies," High-Performance Computing & mdash, Paradigm and Infrastructure, 2006.

[21] Ibrahim, S.; Hai Jin; Li Qi; Chunqiang Zeng, "Grid Maintenance: Challenges and Existing Models," 3rd International Conference Information and Communication Technologies: From Theory to Applications, ICTTA, April 2008. pp. 7-11.

[22] Load Balancing Approaches in Grid Computing Environment. International Journal of Computer Applications 72(12), June 2013, pp. 42-49.

[23] Franke, C., U. Schwiegelshohn, R. Yahyapour, Job Scheduling for Computational Grids, University of Dortmund, Technical Report 0206, 2006.

[24] Y. Zhu, A Survey on Grid Scheduling Systems, Department of Computer Science, Hong Kong University of science and Technology, 2003.

[25] A.Chandak, B. Sahoo & A. Turuk, Heuristic task allocation strategies for computational grid. International Journal advanced networking and applications, 5(2), 2011, pp. 804-810.

[26] Bharti Arora and Sami Anand. Article: Comparison of Efficient Job Scheduling Mechanisms Used in Grid Computing: A Review, International Journal of Computer Applications 70(11), Published by Foundation of Computer Science, New York, USA. May 2013. pp. 39-42.

[27] Kant Soni, V; Sharma, R.; Kumar Mishra, M., "An analysis of various job scheduling strategies in grid computing," 2nd International Conference on Signal Processing Systems (ICSPS), Vol.2, 5-7 July 2010, pp.V2-162-V2-166,

[28] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, Grid Information Services for Distributed Resource Sharing, in Proc. the 10th IEEE International Symposium on High- Performance Distributed Computing (HPDC-10), San Francisco, California, USA, August 2001, pp. 181-194.

[29] R. Wolski, N. T. Spring and J. Hayes, The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing, in the J. of Future Generation Computing Systems, Vol. 15, No. 5-6, January 1999 pp. 757--768.

[30] F.D. Sacerdoti, M.J. Katz, M.L Massie and D.E. Culler, Wide area cluster monitoring with Ganglia, in Proc. of IEEE International Conference on Cluster Computing, Hong Kong, December 2003, pp.289 – 298.

[31] A. Chervenak, E. Deelman, I. Foster, L. Guy, W. Hoschek, A. Iamnitchi, C. Kesselman, P. Kunszt and M. Ripeanu, Giggle: A Framework for Constructing Scalable Replica Location Services, in Proc. of the ACM/IEEE Conference on Supercomputing, Baltimore, Maryland USA, November 2002, pp.1-17.

[32] P. K. Suri & Sunita Rani , Resource Management in Grid Computing: A Review Global Journal of Computer Science and Technology, Network, Web & Security, Vol. 13, Issue 17, Version 1.0, 2013.

[33] D. Thilagavathi, Dr. Antony Selvadoss Thanamani, A Survey on Dynamic Job Scheduling in Grid Environment Based on Heuristic Algorithms, International Journal of Computer Trends and Technology, Vol. 3, Issue 4, 2012.

[34] Buyya, R., Venugopal, S., A gentle introduction to grid computing and technologies. CSI Communications 9, 9–19, July, 2005.

[35] Globus - http://www.globus.org.

[36] Legion: A Worldwide Virtual Computer, http://legion.virginia.edu, accessed Jan. 2013.

[37] UNICORE-Distributed computing and data resources, http://www.unicore.eu, accessed Jan. 2013.

[38] M. Baker, R. Buyya, and D. Laforenza, Grids and Grid Technologies for Wide-Area Distributed Computing, International Journal of Software: Practice and Experience (SPE), Vol. 32, Issue 15, Wiley Press, USA, December 2002, pp. 1437-1466.

[39] James Frey, Todd Tannenbaum, Miron Livny, Ian Foster, Steven Tuecke, "Condor-G: A Computation Management Agent for Multi Institutional Grids," Cluster Computing, Springer Netherlands, Vol. 5, no. 3, July 2002, pp. 237–246.

[40] F.Berman et al. "New Grid Scheduling and Rescheduling Methods in the GRaDS Project," International Journal of Parallel Programming, Springer Netherlands, Vol. 33, nos. 2-3, June 2005, pp. 209 – 229.

[41] NetSolve/GridSolve, http://icl.cs.utk.edu/netsolve, 2013.

[42] David Abramson, Rajkumar Buyya, and Jonathan Giddy, "A Computational Economy for Grid Computing and its Implementation in the Nimrod-G Resource Broker," Future Generation Computer Systems (FGCS), 2002.

[43] Goncalo Borges et al. "Sun Grid Engine, a New Scheduler for EGEE Middleware," Iberian Grid Infrastructure Conference – IBERGRID, May 2007.

[44] Allen, G. et al. "The Grid Application Toolkit: Toward Generic and Easy Application Programming Interfaces for the Grid," Proceedings of the IEEE, Vol. 93, No.3, March 2005, pp.534-550.

[45] EGI - European Grid Infrastructure, http://www.egi.eu/about/egi-inspire//, 2013.

[46] Condor-High Throughput Computing, http://research.cs.wisc.edu/htcondor/, 2013.

[47] Akshay Luther, Rajkumar Buyya, Rajiv Ranjan & Srikumar Venugopal, Peer-to-Peer Grid Computing and a .NET-based Alchemi Framework , High Performance Computing: Paradigm and Infrastructure, Laurence Yang and Minyi Guo (editors), ISBN: 0-471-65471-X, Wiley Press, New Jersey, USA, June 2005.

[48] Henri Casanova, "Simgrid: a Toolkit for the Simulation of Application Scheduling," Proceedings of the first IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID), 2001.

[49] Rajkumar Buyya, Manzur Murshed, "GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing,"

Concurrency and Computation: Practice and Experience, 14, 2002, pp. 1175 – 1220.

[50] David G.Cameron, Ruben Carvajal-Schaffino, A.Paul Millar, Caitriana Nicholson, Kurt Stockinger, Floriano Zini, "Evaluating Scheduling and Replica Optimisation Strategies in OptorSim," International Conference on Grid Computing, Proceedings of the 4th International Workshop on Grid Computing, 2003, page 52.

[51] Atsuko Takefusa, Satoshi Matsuoka, Kento Aida, Hidemoto Nakada, Umpei Nagashima, "Overview of a Performance Evaluation System for Global Computing Scheduling Algorithms," Proceedings of the 8th IEEE International Symposium on High Performance Distributed Computing, 1999, page 11.

[52] Guttromson, R.T.; Chassin, D.P.; Widergren, S.E., Residential energy resource models for distribution feeder simulation, IEEE 2003 Power Engineering Society General Meeting, Vol. 1, 13–17 July 2003.

[53] Cactus, http://cactuscode.org, 2013.

[54] J. M. Schopf. A general architecture for scheduling on the grid, 2002. Special issue of JPDC on Grid Computing.

[55] Ramin Yahyapour. Grid scheduling architecture, 2002. http://www-ds.e-technik.unidortmund. de/.

[56] Sanjoy Baruah, Shelby Funk, and Joel Goossens. Robustness results concerning EDF scheduling upon uniform multiprocessors. IEEE Transactions on Computers, 52(9), 2003, pp. 1185-1195.

[57] M. Dertouzos and A. K. Mok. Multiprocessor scheduling in a hard real-time environment. IEEE Transactions on Software Engineering, 15(12), 1989, pp. 1497-1506.

[58] Kwang S. Hong and Joseph Y.-T. Leung. On-line scheduling of real-time tasks. IEEE Transactions on Computers, 41(10), 1992, pp. 1326-1331.

[59] Bharti Arora and Sami Anand. Article: Comparison of Efficient Job Scheduling Mechanisms Used in Grid Computing: A Review. International Journal of Computer Applications 70(11), May 2013. Published by Foundation of Computer Science, New York, USA, pp. 39-42.

[60] Task Scheduling Heuristic in Grid Computing, International Journal of Computer Applications and Technology. Vol. 1, Issue 2, 2012, pp. 49-52.

[61] A. Abraham, R. Buyya and B. Nath, Nature's Heuristics for Scheduling Jobs on Computational Grids, in Proc. of 8th IEEE International Conference on Advanced Computing and Communications (ADCOM 2000), Cochin, India, December 2000, pp. 45-52.

[62] T. Casavant, and J. Kuhl, A Taxonomy of Scheduling in General-purpose Distributed Computing Systems, in IEEE Trans. on Software Engineering Vol. 14, No.2, February 1988, pp. 141-154.

[63] A. Andrieux, D. Berry, J. Garibaldi, S. Jarvis, J. MacLaren, D. Ouelhadj and D. Snelling, Open Issues in Grid Scheduling, Official Technical Report of the Open Issues in Grid Scheduling Workshop, Edinburgh, UK, October 2003.

[64] R. Sharma, V. K. Soni, M. K. Mishra, P. Bhuyan, U. C. Dey An Agent Based Dynamic Resource Scheduling Model with FCFS-Job Grouping Strategy in Grid Computing, ICCGCS-2010.

[65] New heuristics for flow shop problem to minimize makespan, D Bai and L Tang, Journal of the operational research society, 2010 61, pp. 1032-1040.

[66] V Vasudevan and R Vijayalakshmi. Article: Heuristic Algorithm for Balancing Load in Grid Task Scheduling. International Journal of Computer Applications 67(15), Published by Foundation of Computer Science, New York, USA, April 2013, pp. 38-41.

[67] Li Wenzheng; Zhang Wenyue, "An Improved Scheduling Algorithm for Grid Tasks," International Symposium on Intelligent Ubiquitous Computing and Education, 15-16 May 2009, pp.9-12.

[68] Sofia K. Dimitriadou, Helen D. Karatza, "Multi- Site Allocation Policies on a Grid and Local Level", Electronic Notes in Theoretical Computer Science 261, 2010, pp. 163–179.

[69] Li-Ya Tseng, Yeh-Hao Chin, Shu-Ching Wang, 'A minimized makespan scheduler with multiple factors for Grid computing systems", Expert Systems with Applications 36, 2009, pp. 11118-11130.

[70] Syed Muhammad Ahsan. ―A framework for QoS computation in web service and technology selection computer standards & Interfaces. 2006, 28(6),p.714-720

[71] Run-ze Wu; Ming-shan Wu; Xiang Mi; Qi An, "Task Scheduling Algorithm Based on Triangle Module in Grid Computing," 8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), 21-23 Sept. 2012.pp.1-4.

[72] Ding Ding; Siwei Luo; Lihua Ai; Yidong Li, "A User Preference Driven Approach for Multi-QoS Constrained Task Scheduling in Grid," 13th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), 14-16 Dec. 2012, pp. 493-498.

[73] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger. Economic Models for Resource Management and Scheduling in Grid Computing. The Journal of Concurrency and Computation, 14, 2002, pp. 1507-1542.

[74] Caramia, Massimiliano and Giordani, Stefano,"An Economic Model for Resource Allocation in Grid Computing", Operations Research, Vol. 59, No. 4, 2011, pp. 956-972.

[75] Baghban, H.; Hashemi, M., "A non-cooperative bargaining model for grid resource allocation," Humanities, Science and Engineering (CHUSER), 2011 IEEE Colloquium on, 5-6 Dec. 2011, pp.255-260.

[76] Yousif, A.; Abdullah, A.H.; Bashir, M.B., "Evolution and new architecture for bidding based grid resource selection," National Postgraduate Conference (NPC), 19-20 Sept. 2011, pp.1-5.

[77] Cendron, M.M.; Westphall, C.B., "A Price-Based Task Scheduling for Grid Computing," Seventh International Conference on Networking,. ICN 2008, 13-18 April 2008, pp.762-766.

[78] Ni L.M. Zhiwei Xu Lijuan Xiao, Yanmin Zhu. Incentive based Scheduling for Market Like Computational Grids. IEEE Transactions on Parallel and Distributed Systems, 19, 2008, pp. 903-913.

[79] Li Mingbiao; Li Jian; Xie Shengli, "Posted Price Model Based on GRS and Its Optimization Using in Grid Resource Allocation," International Conference on Wireless Communications, Networking and Mobile Computing, WiCom 21-25 Sept. 2007, pp.3172-3175.

[80] Market-based Resource Allocation in Grids Pourebrahimi, B.; Bertels, K.; Kandru, G. M.; Vassiliadis, S., "Market-Based Resource Allocation in Grids," Second IEEE International Conference on e-Science and Grid Computing, e-Science '06, Dec. 2006.

[81] Preetam Ghosh, Nirmalya Roy, Sajal K. Das, Kalyan Basu, A pricing strategy for job allocation in mobile grids using a non-cooperative bargaining theory framework, Journal of Parallel and Distributed Computing, Volume 65, Issue 11, November 2005, pp. 1366-1383.

[82] Bansal, S.; Hota, C., "Goodwill based scheduling algorithm for economy grid," IEEE 3rd International on Advance Computing Conference (IACC), 22-23 Feb. 2013,

pp. 56-60.

[83] Fard, H.M.; Deldari, H., "An Economic Approach for Scheduling Dependent Tasks in Grid Computing," 11th IEEE International Conference on Computational Science and Engineering Workshops, CSEWORKSHOPS '08m, 16-18 July 2008, pp.71-76.

[84] Juan Chen, "Economic Grid Resource Scheduling Based on Utility Optimization," 2010 Third International Symposium on Intelligent Information Technology and Security Informatics (IITSI), 2-4 April 2010, pp.522-525.

[85] Depoorter, W.; Van den Bossche, R.; Vanmechelen, K.; Broeckhove, J., "Evaluating the Divisible Load Assumption in the Context of Economic Grid Scheduling with Deadline-Based QoS guarantees," 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009. CCGRID '09, 18-21 May 2009, pp.452-459.

[86] A review of population based meta-heuristics algortihms, Zahra Beheshti, Siti Mariyam Hj. Shamsuddin Int. J. Advance. Soft Comput. Appl., Vol. 5, No. 1, March 2013.

[87] Zhihui Du, Man Wang, Yinong Chen, Yin Ye, Xudong Chai, The Triangular Pyramid Scheduling Model and algorithm for PDES in Grid, Simulation Modelling Practice and Theory, Vol. 17, Issue 10, November 2009, pp. 1678-1689.

[88] Jiayi Zhou, Kun-Ming Yu, Chih-Hsun Chou, Li-An Yang, Zhi-Jie Luo,A Dynamic Resource Broker and Fuzzy Logic Based Scheduling Algorithm in Grid Environment, Adaptive and Natural Computing Algorithms, Lecture Notes in Computer Science, Vol. 4431, 2007, pp 604-613.

[89] Jingbo Yuan; Shunli Ding; Cuirong Wang, "Tasks Scheduling Based on Neural Networks in Grid," Third International Conference on Natural Computation, ICNC 2007, Vol. 3, 24-27 Aug. 2007, pp.372-376.

[90] Teo, Y. M.; Wang, X.; Gozali, J. P., "A compensation-based scheduling scheme for grid computing," Seventh International Conference on High Performance Computing and Grid in Asia Pacific Region, 2004. Proceedings, 20-22 July 2004, pp.334-342.

[91] Glover F., (1986). Future paths for integer programming and links to artficial intelligence, Computers and Operations Research, 13, 1986, pp. 533-549.

[92] Laporte, G., Osman, I. H., “Routing problems: A bibliography”, Annals Operations Research, Vol. 61, 1995, pp. 227-262.

[93] Yang Gao, Hongqiang Rong, Joshua Zhexue Huang, Adaptive grid job scheduling with genetic algorithms, Future Generation Computer Systems, Volume 21, Issue 1, 1 January 2005, Pages 151-161.

[94] Xhafa, F.; Alba, E.; Dorronsoro, B., "Efficient Batch Job Scheduling in Grids using Cellular Memetic Algorithms," IEEE International on Parallel and Distributed Processing Symposium, IPDPS 2007, 26-30 March 2007, pp.1-8.

[95] Joanna Kolodziej ,Fatos Xhafa, Lukasz Kolanko , "Hierarchic Genetic Scheduler Of Independent Jobs In Computational Grid Environment", published in Proceedings 23rd European Conference on Modelling and Simulation, ECMS 2009, Madrid, spain.

[96] Dudy Lim, Yew-Soon Ong, Yaochu Jin, Bernhard Sendhoff, Bu-Sung Lee, Efficient Hierarchical Parallel Genetic Algorithms using Grid computing, Future Generation Computer Systems, Vol. 23, Issue 4, May 2007, pp. 658-670.

[97] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys, 35(3), 2003, pp. 268–308.

[98] Iztok Fister Jr., Xin-She Yang, Iztok Fister, Janez Brest, Duˇsan Fister, A Brief Review of Nature-Inspired Algorithms for Optimization, 2013.

[99] Mobini, M.H.; Entezari-Maleki, R.; Movaghar, A., "Biogeography-based optimization of makespan and reliability in grid computing systems," 4th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 3-5 Oct. 2012, pp.336-342.

[100] Pooranian Z, Shojafar M, Javadi B, Independent task scheduling in grid computing based on queen bee algorithm. IAES Int J Artif Intell 1(4), 2012, pp. 171–181.

[101] Seyyed Mohsen Hashemi, Ali Hanani, Solving the Scheduling Problem in Computational Grid using Artificial Bee Colony Algorithm, Advances in Computer Science: an International Journal, 2013.

[102] Maryam Rabiee and Hedieh Sajedi. Article: Job Scheduling in Grid Computing with Cuckoo Optimization Algorithm. International Journal of Computer Applications, Published by Foundation of Computer Science, New York, USA, 62(16), Jan. 2013, pp. 38-44.

[103] Pooranian Z, Harounabadi A, Shojafar M, Hedayat N, New hybrid algorithm for task scheduling in grid computing to decrease missed task. In: World academy of science, engineering and technology, Vol. 55, 2011, pp. 5–9.

[104] Fidanova, S., "Simulated Annealing for Grid Scheduling Problem," International Symposium on Modern Computing, JVA '06. IEEE John Vincent Atanasoff, 3-6 Oct. 2006, pp.41-45.

[105] Fatemeh Heydariand Hadi Shahriar Shahhoseini, Dynamic and Adaptive Load Balancing using Harmony Search in Grid Computing, IJCSNS International Journal of Computer Science and Network Security, Vol.12 No.9, September 2012.

[106] Fatos Xhafa, Javier Carretero, Bernabé Dorronsoro,Enrique Alba, A Tabu Search Algorithm for Scheduling Independent Jobs in Computational Grids, Vol. 28, No. 2, 2009, pp. 237–250.

[107] Qingjiang Wang; Yun Gao; Peishun Liu, "Hill Climbing-Based Decentralized Job Scheduling on Computational Grids," First International Multi-Symposiums on Computer and Computational Sciences, IMSCCS '06, Vol.1, 20-24 June 2006, pp.705-708.

[108] Talbi, E.G.: A Taxonomy of Hybrid Meta-heuristics. J. Heuristics 8(5), 2002, pp. 541–564.

[109] Meta-heuristics for Grid Scheduling Problems, Fatos Xhafa and Ajith Abraham, Metaheuristics for Scheduling in Distributed Computing Environments Studies in Computational Intelligence Volume 146, 2008, pp. 1-37.

[110] Zahra Pooranian, Mohammad Shojafar , Reza Tavoli, MukeshSinghal, Ajith Abraham "A Hybrid Metaheuristic Algorithm for Job Scheduling on Computational Grids", Informatica 37, 2013, pp.-157-164.

[111] Hamid salehi, Reza Boostani, Asou Aminnezhad, Task scheduling Poblem in Grid Networks by Combination of Genetic Algorithms and Gravity Algorithms, International Research Journal of Applied and Basic Sciences, Vol, 4 (5), pp. 1287-1296.

[112] Uma B. Gurav and L.D. Netak, Hybrid Genetic Algorithm Based Task Scheduling in Heterogeneous Grid, published in Proceedings of SPIT-IEEE Colloquium and International Conference, Mumbai, India, Vol. 5, 2012, pp. 37-41.

[113] Farzaneh Fatemipour, Farnoush Fatemipour, Scheduling

Scientific Workflows using Imperialist Competitive Algorithm, International Conference on Industrial and Intelligent Information (ICIII 2012) IPCSIT, Vol.31, IACSIT Press, Singapore, 2012.

[114] C.Kalpana, U.Karthick Kumar, R.Gogulan, "A Randomized Load Balancing Algorithm in grid using Max-Min PSO Algorithm", International Journal of Research in Computer Science, 2 (3), April 2012, pp. 17-23.

[115] Goswami, R.; Ghosh, T.K.; Barman, S., "Local search based approach in grid scheduling using Simulated Annealing," 2nd International Conference on Computer and Communication Technology (ICCCT), 15-17 Sept. 2011, pp.340-345.

[116] Liang-Teh Lee; Chin-Hsiian Liang; Hung-Yuan Chang, "An Adaptive Task Scheduling System for Grid Computing," The Sixth IEEE International Conference on Computer and Information Technology, CIT '06., Sept. 2006.

[117] Ahuva W. Mu'alem and Dror G. Feitelson, "Utilization, Predictability, Workloads, and User Runtime Estimates in Scheduling the IBM SP2 with Backfilling". IEEE Transactions on Parallel and Distributed Systems, 12: (6), 2001, pp. 529-543.

[118] Armstrong, R.; Hensgen, D.; Kidd, T., "The relative performance of various mapping algorithms is independent of sizable variances in run-time predictions," Heterogeneous Computing Workshop (HCW 98) Proceedings. Seventh, 30 Mar. 1998, pp.79-87.

[119] Jing Wang; Gongqing Wu; Bin Zhang; Xuegang Hu, "A Heuristic Algorithm for Scheduling on Grid Computing Environment," ChinaGrid Annual Conference (ChinaGrid), Seventh , 20-23 Sept. 2012, pp.36-42

[120] George Amalarethinam. D.I, Vaaheedha Kfatheen .S, "Max-min Average Algorithm for Scheduling Tasks in Grid Computing Systems", George Amalarethinam. D.I et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (2), 2012, pp. 3659-3663.

[121] Fahd Alharbi, "Simple Scheduling Algorithm with Load Balancing for Grid Computing", Asian Transactions on Computers (ATC),Vol. 02, Issue 02, May, 2012.

[122] Syed Nasir Mehmood Shah, Ahmad Kamil Bin Mahmood, Alan Oxley, Dynamic Multilevel Hybrid Scheduling Algorithms for Grid Computing, Procedia Computer Science, Vol. 4, 2011, pp. 402-411.

[123] T. Kokilavani, Dr. D.I. George Amalarethinam, "Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing",International Journal of Computer Applications, Vol. 20, No.2, April 2011.

[124] D. Doreen Hephzibah Miriam K. S. Easwarakumar, "A Double Min Min Algorithm for Task Metascheduler on Hypercubic P2P Grid Systems", IJCSI, Published in Vol. 7, Issue 4, No 5, July 2010, pp. 8-18.

[125] Kamalam.G.K and Murali Bhaskaran.V, A New Heuristic Approach: Min-mean Algorithm For Scheduling Meta-Tasks On Heterogeneous Computing Systems, IJCSNS International Journal of Computer Science and Network Security, Vol.10 No.1, Jan. 2010.

[126] Syed Nasir Mehmood Shah, Ahmad Kamil Bin Mahmood and Alan Oxley, Modified Least Cost Method for Grid Resource Allocation, International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2010.

[127] Ang Li; Nianming Yao; Peiyu Hong, "A cost and time balancing algorithm for scheduling parallel tasks on Computing Grid," International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE), Vol.1, 24-26 Aug. 2010, pp.185-188.

[128] Guoqi Yang, Yanming Shen, Keqiu Li, Wenyu Qu, A Comprehensive Task Scheduling Algorithm in Grid, The Fifth Annual ChinaGrid Conference, 2010.

[129] Li Wenzheng; Zhang Wenyue, "An Improved Scheduling Algorithm for Grid Tasks," International Symposium on Intelligent Ubiquitous Computing and Education, 15-16 May 2009, pp.9-12.

[130] Quan Liu; Yeqing Liao, "Grouping-Based Fine-Grained Job Scheduling in Grid Computing," First International Workshop on Education Technology and Computer Science, ETCS '09, Vol.1, 7-8 March 2009, pp.556-559.

[131] Hesam Izakian, Ajith Abraham, Senior Member, IEEE, Václav Snášel, Comparison of Heuristics for Scheduling Independent Tasks on Heterogeneous Distributed Environments, 2009.

[132] Saeed Parsa and Reza Entezari-Maleki,"RASA: A New Task Scheduling Algorithm in Grid Environment",World Applied Sciences Journal 7 (Special Issue of Computer & IT), 2009, pp. 152-160.

[133] Ehsan Ullah Munir, Jianzhong Li, Shengfei Shi, Zhaonian Zou and Donghua Yang.,MaxStd: A Task Scheduling Heuristic for Heterogeneous Computing Environment. Information Technology Journal, 7 (4), 2008, pp. 679-683.

[134] Diana Moise, Izabela Moise, Florin Pop, Valentin Cristea, "Resource CoAllocation for Scheduling Tasks with Dependencies, in Grid", International Workshop on High Performance in Grid Middleware (HiPerGRID 2008), Bucharest, Romania, 2008.

[135] Wu, H.; Chong-Yen Lee; Wuu-Yee Chen; Tsang-Yean Lee, "A Job Schedule Model Based on Grid Environment," First International Conference on Complex, Intelligent and Software Intensive Systems, CISIS, 10-12 April 2007, pp.43-52.

[136] Etminani .K, and Naghibzadeh. M, "A Min-min Max-min Selective Algorithm for Grid Task Scheduling," The Third IEEE/IFIP International Conference on Internet, Uzbekistan, 2007.

[137] Ng Wai Keat, Ang Tan Fong, Ling Teck Chaw, Liew Chee Sun, " Scheduling Framework for Bandwidth-Aware Job Grouping", Malaysian Journal of Computer Science, Vol. 19(2), 2006.

[138] Braun, T.D. et al. "A comparison study of static mapping heuristics for a class of meta-tasks on heterogeneous computing systems," Heterogeneous Computing Workshop, (HCW '99) Proceedings. Eighth, pp.15-29, 1999.

[139] A. Abraham, R. Buyya, and B. Nath, "Nature's heuristics for scheduling jobs on computational grids", In: The 8th IEEE International Conference on Advanced Computing and Communications, India, 2000.

[140] Casanova, H.; Legrand, A.; Zagorodnov, D.; Berman, F., "Heuristics for scheduling parameter sweep applications in grid environments," Heterogeneous Computing Workshop, (HCW 2000) Proceedings. 9th, 2000, pp.349-363.

[141] M. Macheswaran, S. Ali, H. J. Siegel, D. Hensgen and R. F. Freund, "Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems," Journal of Parallel Distributed Computing, Vol. 59, No. 2, 1999, pp. 107-131.

[142] V.S. Anil Kumar, Madhav V. Marathe, Srinivasan Parthasarathy, Aravind Srinivasan, Minimum Weighted Completion Time, ,1999 .

[143] Joanna Kolodziej, Fatos Xhafa, "Meeting Security and User Behavior Requirements in Grid Scheduling",

Simulation Modeling Practices and Theory, 2010.

[144] Amit Agarwal and Padam Kumar "Multidimensional Qos Oriented Task Scheduling In Grid Environments", International Journal of Grid Computing & Applications (IJGCA) Vol.2, No.1, March 2011.

[145] Juefu Liu; Gang Li, "An improved MIN-MIN grid tasks scheduling algorithm based on QoS constraints," International Conference on Optics Photonics and Energy Engineering (OPEE), Vol.1, 10-11 May 2010, pp.281-283.

[146] Kunfang SONG, Shufen RUAN, Minghua JIANG, "A Flexible Grid Task Scheduling Algorithm Based on QoS Similarity", Journal of Convergence Information Technology Vol. 5, No. 7, Sept. 2010.

[147] Sameer Singh Chauhan and R.C. Joshi, "Multiple QoS Guided Heuristic for Independent Task Scheduling in Grid", Information and Communication Technologies Communications in Computer and Information Science, Vol. 101, 2010, pp. 136-141.

[148] Nawfal A. Mehdi, Ali Mamat, Hamidah Ibrahim, Shamala A/P K, "Multiphase Scalable Grid Scheduler Based on Multi-QoS Using Min-Min Heuristic", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 1, No. 3, Sept. 2010.

[149] Chauhan, S.S.; Joshi, R.C., "A heuristic for QoS based independent task scheduling in Grid environment," International Conference on Industrial and Information Systems (ICIIS), July 29-Aug. 1, 2010, pp.102-106.

[150] Sameer Singh Chauhan, R.C. Joshi, QoS Guided Heuristic Algorithms for Grid Task Scheduling, International Journal of Computer Applications, Vol. 2, No.9, June 2010.

[151] Singh. M and Suri. P.K, QPS- A QoS Based Predictive Max-Min, Min-Min Switcher Algorithm for Job Scheduling in a Grid, Information Technology Journal, Vol. 7, Issue: 8, 2008, pp. 1176-1181.

[152] Baraglia, R.; Ferrini, R.; Tonellotto, N.; Ricci, L.; Yahyapour, R., "QoS-constrained List Scheduling Heuristics for Parallel Applications on Grids," 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing,. PDP, 13-15 Feb. 2008, pp.379-383.

[153] Khanli, L.M.; Analoui, M., "Grid_JQA: A QoS Guided Scheduling Algorithm for Grid Computing," Sixth International Symposium on Parallel and Distributed Computing,. ISPDC '07. , 5-8 July 2007, pp. 34.

[154] Chunlin, Li; Meilai, Feng; Layuan, Li, "Multiple QoS modeling and algorithm in computational grid," Systems Engineering and Electronics, Journal of , Vol. 18, No.2, June 2007, pp.412-417.

[155] Ehsan Ullah Munir, Jianzhong Li and Shengfei Shi, QoS Sufferage Heuristic for Independent Task Scheduling in Grid. Information Technology Journal, 6, 2007, pp. 1166-1170.

[156] Yin-Yun Shen; Xiao-Ping Li; Qian Wang; Ying-Chun Yuan, "A Hybrid QoS-Based Algorithm for Independent Tasks Scheduling in Grid," International Conference on Machine Learning and Cybernetics, 2006, 13-16 Aug. 2006, pp.4511-4515.

[157] Fang Dong; Junzhou Luo; Lisha Gao; Liang Ge, "A Grid Task Scheduling Algorithm Based on QoS Priority Grouping," Fifth International Conference on Grid and Cooperative Computing, GC, Oct. 2006, pp.58-61.

[158] J. Yu, R. Buyya and C.K. Tham, “QoS-based Scheduling of Workflow Applications on Service Grids”, Technical Report, GRIDS-TR-2005-8, Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia, June 9, 2005.

[159] XiaoShan He,Xianhe Sun and Gergor von Laszewski. ―QoS guided Min-Min heuristic for grid task scheduling. Journal of Computer Science and Technology, 18(4), 2003, pp.442-451.

[160] Abdulal, W.; Al Jadaan, O.; Jabas, A.; Ramachandram, S.; Kaiiali, M.; Rao, C. R., "Rank-Based Genetic Algorithm with Limited Iteration for Grid Scheduling," First International Conference on Computational Intelligence, Communication Systems and Networks, CICSYN '09, 23-25 July 2009, pp.29-34.

[161] Shahista Navaz, Uzma Ansari- An Evolutionary Algorithm in Grid Scheduling by multi-objective Optimization using variants of NSGA - published at: "International Journal of Scientific and ResearchPublications (IJSRP), Vol. 2, Issue 9, Sept. 2012 Edition".

[162] Salimi, R.; Motameni, H.; Omranpour, H., "Task scheduling with Load balancing for computational grid using NSGA II with fuzzy mutation," 2nd IEEE International Conference on Parallel Distributed and Grid Computing (PDGC), 6-8 Dec. 2012, pp.79-84.

[163] Chinmoy Kar, Vineet Kumar Rakesh, Tapas Samanta and Sreeparna Banerjee, "A New Approach to Grid Scheduling using Random Weighted Genetic Algorithm with Fault Tolerance Strategy", International Journal of Computer Applications, Published by Foundation of Computer Science, New York, USA, 48(23): June 2012, pp. 42-47.

[164] Joanna Kołodziej, Joanna Kołodziej, Fatos Xhafa, Genetic Algorithms for Energy-aware Scheduling in Computational Grid.

[165] M. Camelo, Y. Donoso, and H. Castro, “MAGS—an approach using multi-objective evolutionary algorithms for grid task scheduling,” International Journal of Applied Mathematics and Informatics, Vol. 5, No. 2, 2011.

[166] Hai ZhuYuping Wang and Lei Fan and Xiaoli Wang, Grid Independent Task Scheduling Multi-Objective Optimization Model and Genetic Algorithm, 2010.

[167] Benedict, S., Rejitha, R.S., Vasudevan, V., An Evolutionary Hybrid Scheduling Algorithm for Computational Grids. Journal of Advanced Computational Intelligence and Intelligent Informatics 12(5), 2008, pp. 479–484.

[168] Grosan, C., Abraham, A., Helvik, B.: Multi-objective Evolutionary Algorithms for Scheduling Jobs on Computational Grids. In: Guimaraes, N., Isaias, P. (eds.) International Conference on Applied Computing, Salamanca, Spain, 2007, pp. 459–463.

[169] S. Selvi, D. Manimegalai, and A. Suruliandi, "Efficient job scheduling on computational grid with differential evolution algorithm," International Journal of Computer Theory and Engineering, Vol. 3, 2011, pp. 277-281.

[170] Wu Yang; Yuanshu Sun, "An Improved Shuffled Frog Leaping Algorithm for Grid Task Scheduling," International Conference on Network Computing and Information Security (NCIS), Vol.1, 14-15 May 2011 pp.342-346.

[171] Zhou Wei; Bu Yan-ping; Zhou Ye-qing, "The application of an improved cultural algorithm in grid computing," Control and Decision Conference (CCDC), 2013 25th Chinese , 25-27 May 2013, pp.4565-4570

[172] Pooranian Z, Shojafar M, Abawajy JH, Singhal M, GLOA: a new job scheduling algorithm for grid computing. Int J Artif Intell Interact Multimed 2(1), 2013, pp. 59–64.

[173] Rajni, Inderveer Chana, Bacterial foraging based hyper-heuristic for resource scheduling in grid computing, Future Generation Computer Systems, Vol. 29, Issue 3,

March 2013, pp. 751-762.

[174] Adil Yousif, Abdul Hanan Abdullah, Sulaiman Mohd Nor, Adil Ali Abdelaziz, Scheduling Jobs on Grid Computing using Firefly algorithm, Journal of Theoretical and Applied Information Technology, Vol. 33 No.2, 30th Nov. 2011, pp. 155-164.

[175] Javid Taheri, Young Choon Lee, Albert Y. Zomaya, Howard Jay Siegel, A Bee Colony based optimization approach for simultaneous job scheduling and data replication in grid environments, Computers & Operations Research, Vol. 40, Issue 6, June 2013, pp. 1564-1578.

[176] Saeed Farzi, "Efficient job scheduling in grid computing with modified artificial fish swarm algorithm," International Journal of computer theory and engineering, Vol. 1, No. 1, April 2009.

[177] Nikkhah, M.; Rahmani, A.M.; Yektaie, M.H.; Nikkhah, M., "A Novel Fault-tolerant Particle Swarm Optimization Scheduler for Scheduling Independent Task in Grid Computing Environment," Eighth IEEE/ACIS International Conference on Computer and Information Science, ICIS, 1-3 June 2009, pp.489-493.

[178] S. Lorpunmanee, M. N. Sap, A. H. Abdullah, and C. Chompoo-inwai, "An ant colony optimization for dynamic job scheduling in grid environment," Proceedings of World Academy of Science, Engineering and Technology, 2007, pp.314-321.

[179] Zahra Pooranian, Mohammad Shojafar, Bahman Javadi, Ajith Abraham, Using Imperialist Competition Algorithm for Independent Task Scheduling in Grid Computing, Journal of Intelligent and Fuzzy Systems, 2013.

[180] Klusacek, D., Matyska, L., Rudova, H.: Local Search for Grid Scheduling. In: Doctoral Consortium at the International Conference on Automated Planning and Scheduling (ICAPS 2007), Providence, RI, USA, 2007.

[181] J. Blythe, S. Jain, E. Deelman, Y. Gil, K. Vahi, A. Mandal, and K. Kennedy. 2005. Task scheduling strategies for workflow-based applications in grids. In Proceedings of the Fifth IEEE International Symposium on Cluster Computing and the Grid (CCGrid'05) - Vol. 2. IEEE Computer Society, Washington, DC, USA, pp. 759-767.

[182] Sara Kardani-Moghaddam, Farzad Khodadadi, Reza Entezari-Maleki, Ali Movaghar, A Hybrid Genetic Algorithm and Variable Neighborhood Search for Task Scheduling Problem in Grid Environment, Procedia Engineering, Vol. 29, 2012, pp. 3808-3814

[183] Attraction Search algorithm (HYGAGA) to Solve Grid Task Scheduling Problem, International Conference on Soft Computing and its Applications (ICSCA'2012) Kuala Lumpur (Malaysia), Aug. 25-26, 2012, pp. 158-162.

[184] Z. Pooranian, A. Harounabadi, M. Shojafar, J. Mirabedini, “Hybrid PSO for Independent Task scheduling in Grid Computing to Decrease Makespan,” in Proc. of International Conference on Future Information Technology,IPCSIT'11, Vol. 13, 2011, pp. 435-439.

[185] Kolodziej, J.; Xhafa, F., "A Game-Theoretic and Hybrid Genetic Meta-Heuristics Model for Security-Assured Scheduling of Independent Jobs in Computational Grids," Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on , 15-18 Feb. 2010, pp. .93-100.

[186] Bhupinder Singh, Seema Bawa, HybridSGSA: SexualGA and Simulated Annealing based Hybrid Algorithm for Grid Scheduling, Global Journal of Computer Science and Technology, Vol. 10, No. 9, 2010.

[187] Xhafa, F., Gonzalez, J.A., Dahal, K.P., Abraham, A.: A GA(TS) hybrid algorithm for scheduling in computational grids. In: Hybrid Artificial Intelligent Systems, Lecture Notes in Computer Science, vol. 5572, 2009, pp. 285–292.

[188] Youchan Zhu; Xueying Guo, "Grid Dependent Tasks Scheduling Based on Hybrid Adaptive Genetic Algorithm," Intelligent Systems, 2009. GCIS '09. WRI Global Congress on, vol.2, no., 19-21 May 2009, pp.35-38.

[189] Hao Tian, "A New Resource Management and Scheduling Model in Grid Computing Based on a Hybrid Genetic Algorithm," Computing, Communication, Control, and Management, 2008. CCCM '08. ISECS International Colloquium on, Vol.3, 3-4 Aug. 2008, pp.113-117.

[190] Kun-Ming Yu ,Zhi-Jie Luo, Chih-Hsun Chou , Cheng-Kwan Chen, Jiayi Zhou,A Fuzzy Neural Network Based Scheduling Algorithm for Job Assignment on Computational Grids, First International Conference, NBiS 2007, Regensburg, Germany, 2007, pp.533-542.

[191] Wanneng Shu; Shijue Zheng; Li Gao; Xiong Wang, "An Hybrid Evaluative Algorithm Applied to Task Scheduling," Communications, Circuits and Systems Proceedings, 2006 International Conference on, Vol. 3, 25-28 June 2006, pp. 2070-2073.

[192] Shijue Zheng; Wanneng Shu; Li Gao, "Task Scheduling using Parallel Genetic Simulated Annealing Algorithm," IEEE International Conference on Service Operations and Logistics, and Informatics, SOLI '06, 21-23 June 2006, pp.46-50.

[193] Wei-Neng Chen; Jun Zhang, "An Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem With Various QoS Requirements," IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, , Vol. 39, No.1, Jan. 2009, pp.29-43.

[194] Shi Lei; Xu Hui-hui, "Grid resource scheduling algorithm based on QoS guided GA," IEEE International Symposium on IT in Medicine & Education, ITIME '09, Vol.1, 14-16 Aug. 2009, pp.1281-1285.

[195] Dalibor Klusacek and Hana Rudova. Improving QoS in computational Grids through schedule-based approach. In Scheduling and Planning Applications Workshop at the Eighteenth International Conference on Automated Planning and Scheduling (ICAPS'08), Sydney, Australia, 2008.

[196] Klusacek, D., Matyska, L., Rudova, H.: Local Search for Deadline Driven Grid Scheduling. In: Third Doctoral Workshop on Mathematical and Engineering Methods in Computer Science (MEMICS 2007), 2007, pp. 74–81.

[197] Chunhua Hu; Min Wu; Guoping Liu; Wen Xie, "QoS Scheduling Algorithm Based on Hybrid Particle Swarm Optimization Strategy for Grid Workflow," Sixth International Conference on Grid and Cooperative Computing, GCC 2007, 16-18 Aug. 2007, pp.330-337.

[198] Mishra, M.K.; Sharma, R.; Kant Soni, V.; Parida, B.R.; Das, R.K., "A memory-aware dynamic job scheduling model in Grid computing," International Conference on Computer Design and Applications (ICCDA), Vol.1, 25-27 June 2010, pp. 545-549.

[199] Molaiy, Saeed; Effatparvar, Mehdi, "Scheduling in Grid Systems using Ant Colony Algorithm", I.J. Computer Network and Information Security, 2014, Vol. 6, Issue 2, Jan. 2014, pp. 16-22.

[200] Fotohi R, Effatparvar M., "A Cluster Based Job Scheduling Algorithm for Grid Computing. I.J. Information Technology and Computer Science, 2013, Vol. 5, No. 12, pp. 70-77.

[201] M. K. Mishra, P. Mohanty and G B Mund., "A Time-Minimization Dynamic Job Grouping-based Scheduling

in Grid Computing", International Journal of Computer Applications, Vol. 40, No. 16, February 2012. Published by Foundation of Computer Science, New York, USA, pp. 16-25.

Authors' Profiles

Prof. Manoj Kumar Mishra is currently working as Assistant Professor in the department of Computer Science and Engineering at KIIT University, Bhubaneswar, Odisha and also involved as a Research Associate in Innovadors Lab Pvt. Ltd, Bhubaneswar. His research interests include Grid Computing, Parallel Computing, Distributed Systems and Algorithms.

Yashwant Singh Patel is currently working as M.Tech research scholar, in the department of Computer Science and Engineering at KIIT University, Bhubaneswar, Odisha. His research interests include Grid Computing, Multiobjective Optimization, Evolutionary Algorithms, Parallel Computing and Algorithms.

Yajnaseni Rout is currently working as Assistant Professor, in the department of Information Technology at Silver Oak College of Engineering & Technology, Ahmedabad. Her research interests include Grid Computing and Multiobjective Optimization.

Dr. G.B.Mund is currently working as a professor, in the department of Computer Science and Engineering at KIIT University, Bhubaneswar, Odisha. He received his PhD in Computer Science and Engineering from IIT Kharagpur. His current research interests include Grid Computing, Program Analysis and Algorithms.