Heuristic Task Allocation Strategies for Computational Grid

Ashish V. Chandak
Department of Computer Science and Engineering, National Institute of Technology, Rourkela, India
Email: avchandak@rediffmail.com
Bibhudatta Sahoo
Department of Computer Science and Engineering, National Institute of Technology, Rourkela, India
Email: bdsahu@nitrkl.ac.in
Ashok Kumar Turuk
Department of Computer Science and Engineering, National Institute of Technology, Rourkela, India
Email: akturuk@ nitrkl.ac.in

ABSTRACT

Task allocation in computational grid is NP complete. Heuristic techniques have been widely used by the researchers to solve task allocation problem in grid computing. In this paper we survey heuristic based task allocation strategies and their efficiency. This strategy optimizes various performance parameters such as makespan, resource utilization, response time, workload balancing, service reliability, fairness deviation and throughput. A task life cycle model has been suggested in computational grid. We have also proposed a classification of heuristic task allocation strategies for computational grid.

Keywords - Task allocation; Grid Computing; Heuristic Classification.

Date of Submission: 12 December 2010 Date of Acceptance: 02 February 2011

I. INTRODUCTION

Because of computational capacity of grid they are useful in many fields like medical science e.g. drug design modeling, brain activity analysis, cellular micro physiology and scientific applications such as weather forecasting, aerospace modeling and commercial application such as stock market portfolio maintenance. These applications require exhaustive data analysis and computational power. Thus the growing computational power for these applications purport the need of distributed computing which can provide huge computational and highly available resources. Thus grid application attract those organization for fulfillment their requirements. Grid computing [1, 2] is considered to be wide area distributed computing. Grid provides sharing, selection and aggregation of distributed resources and makes them use for computational purpose. The important aspect of grid is the scheduling of jobs since there exist high heterogeneity of resources (PCs, Workstations, clusters, and supercomputers) which are geographically distributed and having different time zones, fabric management policies and scheduling, application requirement and design patterns. Scheduling is simple when it comes to single machine it just simply assigning the task to machine but when it comes to grid computing because of their distributed, autonomous and heterogeneity property this strategy will not work.

Grid scheduling is NP complete problem [3]. Various heuristics have been developed to solve this Grid scheduling problem. The four basic heuristic are economic heuristic [4, 5, 6], meta-heuristic [7, 8], population based heuristic [9, 3, 10, 11, 12], hybrid heuristic [13, 14, 15, 16, 17]. A grid scheduler acts as an interface between the user and distributed resources. It hides the complexities of the computational grid from the grid user [3]. This paper presents a brief discussion on various heuristics and their importance in grid scheduling.

II. GRID SYSTEM MODEL

Computational grid is a system having a number of independent sites as shown in Fig.1. A site may have a single computing resources or a number of computing nodes connected in a distributed manner. Resources are not exclusively dedicated to grid usage. Sites can freely participate in grid computing by offering resources. The interaction between those grid resources during the execution of a job requires a scheduling layer that uses a different scheduling paradigm. The multi-site resources are assigned to the applications (user request) with the help of grid scheduler.

A grid can be considered as consisting set of independent sites S each having N computing nodes and M no of tasks to be executed at different sites. So, mathematically the grid can be represented as $G=<S, T_M>$.

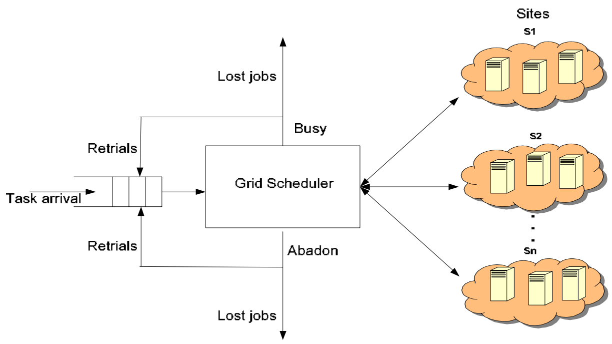

Fig. 1. Grid system model

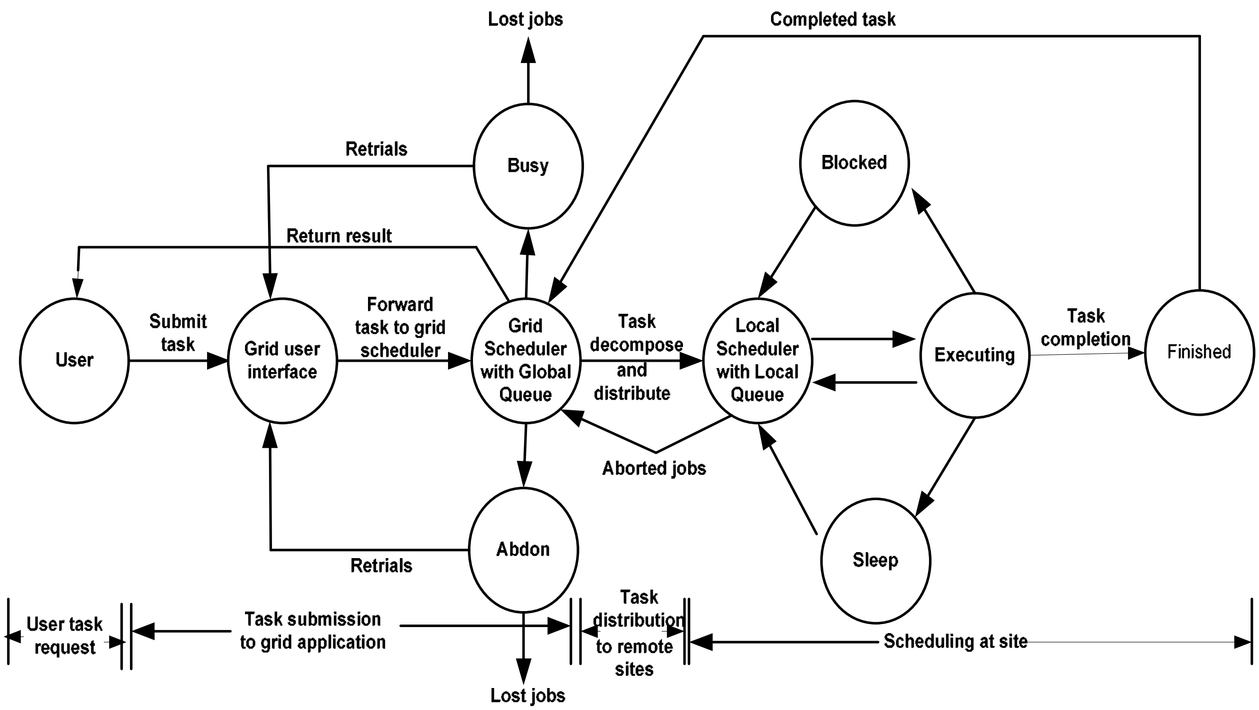

Fig. 2. Task life cycle in computational grid

Where, $S = \{S_i \mid i=1\ldots N\}$, the set of available sites on that grid and $T = \{T_i \mid i=1\ldots M\}$, the set of tasks to be executed. Each site can be considered as consisting of a set of resources R of different types such as storage, computational and network resources. Every site having some attribute associated with it viz. status of the site, maximum load/capacity of the site. So, a site S can be represented as:

S < {R}, status, max_load>

Where,

R = the set of resources available at site S

status = status of the site S i.e., whether the status is working or not working

max_load = the maximum capacity of the site S

Again R can be represented as:

$R = \{M\} \cup \{C\} \cup \{N\}$

Where, M = set of resources of I/O type,

C = set of resources of computational type,

N = set of resources of data type.

III. ALLOCATION AND REMOTE COMPUTING OF TASK IN GRID

The state transition diagram depicted in the Fig. 2 has the following description. It consist of sites at which group of nodes serve incoming jobs. An arriving job that finds scheduler busy signal then it is blocked from entering the system otherwise it is put in global queue. Grid scheduler checks workload and resource availability of each site and if any of the sites satisfy job request then it is forwarded to that site for execution. Scheduling of jobs is done by grid scheduler and starting of job at site is done by local scheduler.

Volume: 02, Issue: 05, Pages: 804-810 (2011)

If sites are busy then the arriving job waits in queue to get service. Job service discipline is first-come, first-served (FCFS). Aborted job from site again comes to global queue. Jobs that are coming for execution in the grid scheduler are divided into two categories new jobs and squirted jobs where squirted jobs means jobs that are aborted from sites while new jobs means jobs that has not any service from time of arrival. To distinguish these two types of jobs it maintains two queue one for new jobs those are arrived in grid application but not scheduled and second queue for squirted jobs. New queue keep jobs in the order of arrival while squirted queue keep job based on the size of CPU consumption. The job having smallest CPU consumption is place at the head of queue while job having largest CPU consumption is place at the tail of queue. Scheduling is done in such way job in new queue are scheduled first and then squirted queue jobs are scheduled in sites. Once a job exits from site it releases the resources it used and these resources again become available to arriving jobs. A fraction of jobs that do not receive service become retrials that attempt to reenter in grid application. The remaining blocked and abandoned jobs are lost. Finally, served jobs return to user.

Table 1 Parameter Summary

Parameter	**Meaning**
ET	Expected execution time
Comm	Communication cost
Comp.	Computation cost
S	Sites
CT	Expected completion time
b	Starting time
α	Size of job
β	Bandwidth of network

ET_{ij} of task T_i having workload W_i and deadline D on site S_j is defined as the amount of time taken by S_j to execute T_i when T_i is assigned $ET(T_{i,} S_j)$ = Comm. (T_i) +Comp (T_i)

Where Comm. $(T_i) = \beta_{i1} + \beta i_2 + \beta i_3$

β_{i1} represents the communication cost between user and grid user interface.

β_{i2} represents the communication cost between grid user interface and grid scheduler.

β_{i3} represents the communication cost between grid scheduler and remote site.

The expected completion time of i^{th} task on j^{th} remote site is defined as time at which site S_j completes task T_i .Let the arrival time task T_i in grid application be a_i and starting time of task T_i be b_j then from the above definition $CT(T_{i,} S_j) = b_j + ET (T_{i,} S_j)$

The b_j can be defines as

$$b_j = \text{free } (j) + 1$$

Grid Scheduler with global queue uses different heuristic algorithm to distribute thae task to different sites. The local scheduler at a particular site is reponsible for actual execution of the assigned task.

IV. PROPOSED HEURISTIC CLASSIFICATION

Heuristic methods uses combination of quantitative and qualitative method to arrive at consequence of problem set of predicted outcomes [18]. Fig. 3 hierarchically organizes different Class of heuristic strategies for task allocation.

A. Population Based Heuristic

It is computational method that optimizes problem by taking population of individuals. This heuristic calculates the fitness of each individual and based on their fitness individual are takes out .The new population is used for next iteration of algorithm. When satisfactory fitness level is reached then algorithm stops.

We have identified three categories of population based heuristic: Genetic algorithm (GA), Particle swarm optimization (PSO), Memetic algorithm (MA), Ant colony optimization (ACO).

Xhafa et al. uses Memetic algorithm for grid task allocation [9]. The ant colony optimization algorithm is a probabilistic technique for solving NP complete problems. Applications of ACO for ask allocation in computational Grid have been studied in [3, 19]. A new variant of ACO namely Balanced Ant Colony Optimization (BACO) for job scheduling in grid is discussed in [20]. The proposed BACO (Balanced ACO) algorithm is compared with ACO (Improved ACO), FPLTF (Fastest Processor to Largest Task First), dynamic FPLTF, sufferage, and random selection method in the experiments and it is found out that BACO is capable of achieving system load balance better than other job scheduling algorithms. GA for task allocation is used in [11, 21].

B. Meta Heuristic

It designates a computational method that optimizes a problem by iteratively trying to improve a candidate solution.

Meta heuristics make few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. Meta heuristics are used for combinatorial optimization in which an optimal solution is sought over a discrete search-space.

We distinguished two meta-heuristics viz. simulated annealing (SA), tabu search (TS). SA for scheduling has been identified by [7] while [8] has implemented TS for scheduling which uses fuzzy sets to model the uncertain processing time.

C. Hybrid Heuristic

Hybrid heuristic are use to optimize the problem by taking two or more heuristic for scheduling jobs in grid e.g. population heuristic + Meta heuristic.

Hybridization of GA and TS is discussed in [13], [14] attempted to address hybridization of genetic algorithm, tabu search and simulated annealing. Hao Tian, discussed the hybridization of GA and ACO for Resource Management and Scheduling [15]. In paper [16], combined of two evaluative algorithms GA and SA bring forward a hybrid evaluative algorithm (HEA) and has been applied to solve task scheduling problem in grid computing. Another hybrid approach for grid scheduling is introduced in [17] that combine GA and game theory.

D. Economic Based Heuristic

In competitive market there is always scarcity of resources. Economic heuristic deals with matching jobs to available resources in economical way such that resource provider and consumer get sufficient incentive to stay and play in competitive market.

Macroeconomic based grid resource allocation introduced in [4] while [22] introduced various economic approaches such as commodity market model, posted price model, bargaining model, tendering/contract-net model, auction model, bid-based, proportional resource sharing model, and community/coalition/bartering model for grid resource allocation. Izakian et al. introduced a continuous double auction (CDA) for grid resource allocation in which resources are considered as provider agent and users as consumer agent [6]. A combinatorial auction based resource allocation protocol in which a user bids a price value for each of the possible combinations of resources required for its tasks execution is the finding in [23]. It uses approximation algorithm for solving combinatorial auction problem and a grid resource allocation model in which user and grid service providers participate in the combinatorial double auction for the resource allocation is the research finding in [24]. Resource allocation in Grid can be possible through commodity market model in which value of grid resource is directly determined by buyer and seller [5]. This model uses smale's method for finding price equilibrium in grid market. Another novel approach proposed in [25] which is incentive based scheduling scheme for grid resource allocation which utilizes a peer-to-peer decentralized scheduling framework.

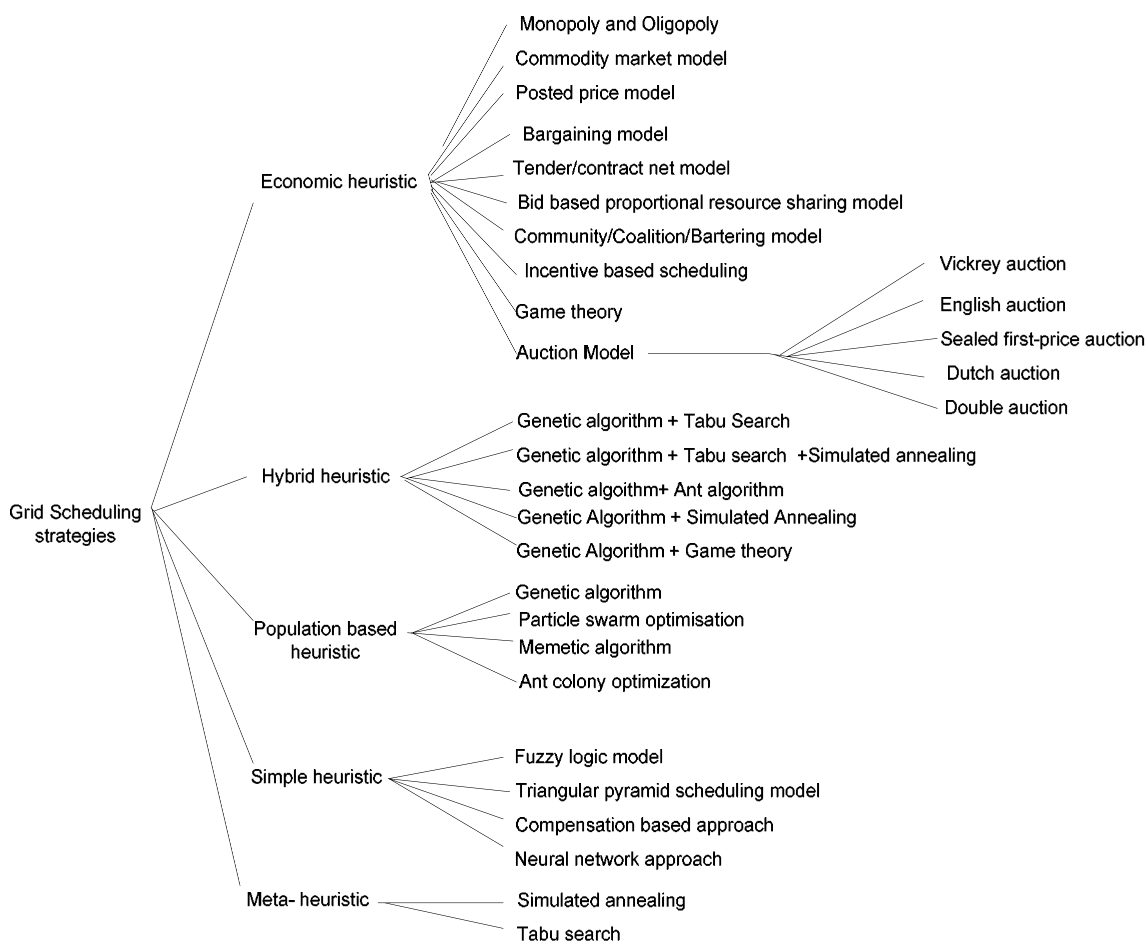

Fig. 3 Task allocation strategies classification

E. Simple Heuristic

Resource allocation in computational grid can be modeled as variant of the 0-1 multidimensional knapsack problem in [26]. This model uses knapsack problem which can find optimal mapping of task to resources. A fuzzy model, that uses fuzzy controller to learn relationship between application requirements and resource needs for specific requirement and request arriving rate [27]. Here controller continuously monitoring resource requirement based on current application request. A discussion on triangular pyramid scheduling (TPS) model for resource allocation in grid can be found in [29]. It uses triangular approach in which resources are decompose into three categories and it consider in-depth relationship between them in order to find better scheduling solutions. An compensation based strategies for grid resource allocation is based upon the compensation of resource loss during application execution by dynamically allocating additional resources [28]. It consist three components namely Execution time Estimator (Deadline Estimator), Performance Monitor, Resource Compensator. When request comes from user it calculates estimated application execution time using execution time estimator and send that estimated time to user. If user is agreed then it compensates resources for that job while [31] presented backpropogation (BP) neural network model for task scheduling in grid.

F. Emergence of Heuristic in Grid Task Scheduling

1. Easier job control and improve allocation efficiency:- Heuristic methods when combined with grid scheduler ensures that access to adequate resources to perform correct scheduling and competition for limited resources does not deadlock where two or more jobs unable to complete resolving such situations where they do occur, and terminating jobs that for any reason are not performing as expected.

2. Improve resource utilization: - The resource utilization is defined as amount of resource is busy in executing jobs. In order for grid resource to be efficiently used the grid manager should increase their utilization rate to decrease their idle time. Population based heuristic such as Ant Colony optimization (ACO), Particle Swarm Optimization [12], Fuzzy Model [27] and Economic based heuristic viz. Continuous Auction (CDA) [6] have proven to improve resource utilization.

3. Minimize makespan*:* - Makespan is measure throughput of completion time. It can be calculated as maximum of completion time. Some population based heuristic such as Memetic Algorithm [9], Ant Colony Optimization (ACO) [10], Particle Swarm Optimization [12], meta-heuristic such as Simulated Annealing [19], hybrid heuristic such as Genetic Algorithm (GA) and Game Theory [17] and simple heuristic such as Triangular Pyramid Scheduling Model (TPS) [29] ensures minimization makespan in grid task scheduling.

4. Minimize response time: - The response time of a job is the time interval between this job's arrivals into the system until it is completed. Response time includes the waiting time in the queues and the service time in the server [30]. Heuristic such as Fuzzy Model [27] minimizes response time when used in grid task scheduling.

5. Better workload balancing: - Workload balancing is the routing of tasks among a group of sites according to the availability and capacity levels of sites. Population based heuristic such as Ant Colony Optimization (ACO) [3, 10] ,meta heuristic such as Simulated Annealing [7], hybrid heuristic such as GA and an Ant Algorithm [15],economic based heuristic such as Macroeconomic Approach [4] achieve better load balancing in grid application.

6. Service reliability*:* - It is defined as

a) Accessibility: - Service is available when desired i.e. when consumer want service.

b) Continuity - Consumer has uninterrupted service over desired duration.

c) Performance - Meets the consumer expectations.

Population based heuristic e.g. Genetic Algorithm [11] achieve service reliability in grid job allocation.

7. Minimize flowtime: - Flow time is sum of finishing time of job. Population based heuristic such as Memetic Algorithm [9], hybrid heuristic such as GA and Game Theory [17] minimize flowtime in grid scheduling.

8. Minimize fairness deviation: - The fairness of the market means that each resource owner has an equal opportunity to offer its resource and it can obtain a fair profit according to its capability [25]. Economic based heuristic viz. Continuous Double Auction (CDA) [6], Commodity market model [5], Incentive based scheduling [25] minimizes fairness deviation among resources.

9. High throughput: - This metric is defined as number of processes that complete their execution per time unit. Meta heuristic viz. Simulated Annealing [7], hybrid heuristic GA and Game Theory [17] and simple heuristic such as Fuzzy Model [27] gives high throughput.

Conclusion

In this paper, we have proposed a classification of heuristic task allocation strategies for computational grid. We identify different types of heuristic such as population based heuristic, meta-heuristic, simple heuristic, hybrid heuristic and economic heuristic in grid task scheduling. Emergence of heuristic in grid task scheduling is discussed against the various performances metric.

References

[1] I. 29, C. Kesselman, S. 28, The anatomy of the grid: Enabling scalable virtual organizations, *International J. Supercomputer Applications 15.*

[2] I. 29, C. Kesselman (Eds.), *The Grid 2: Blueprint for a New Computing Infrastructure*, 2nd ed., Morgan Kaufmann, 2003.

[3] Kousalya.K and Balasubramanie.P,Ant Algorithm for Grid Scheduling Powered by Local Search, *Int.*

J. Open Problems Compt. Math., *Vol. 1*, No. 3, December 2008.

[4] Peijie Huang, Hong Peng, Piyuan Lin and Xuezhen Li,Macroeconomics based Grid resource allocation, *Future Generation Computer Systems* 24 (2008) ,694–700.

[5] Gunther Stuer, Kurt Vanmechelen and Jan Broeckhove,A commodity market algorithm for pricing substitutable Grid resources, *Future Generation Computer Systems* 23 (2007) ,688–701.

[6] Hesam Izakian, Ajith Abraham and Behrouz Tork Ladani, *An auction method for resource allocation in computational grids*, Future Generation Computer Systems 26 (2010), 228-235.

[7] Stefka Fidanova, Simulated Annealing for Grid Scheduling Problem, jva, pp.41-45, IEEE John Vincent Atanasoff 2006 *International Symposium on Modern Computing* (JVA'06), 2006.

[8] Fayad, C., Garibaldi, J.M., Ouelhadj, D., Fuzzy Grid Scheduling Using Tabu Search, *Fuzzy Systems Conference*, 2007. FUZZ-IEEE 2007. *IEEE International, vol*., no., 1-6, 23-26 July 2007.

[9] Fatos Xhafa, Enrique Alba and Bernabe Dorronsoro, Efficient Batch Job Scheduling in Grids using Cellular Memetic Algorithms, ipdps, 247, 2007 *IEEE International Parallel and Distributed Processing Symposium*, 2007.

[10] Ruay-Shiung Chang, Jih-Sheng Chang and Po-Sheng Lin, An ant algorithm for balanced job scheduling in grids, *Future Generation Computer Systems 25* (2009), 20–27.

[11] Yang Gaoa, Hongqiang Rongb and Joshua Zhexue Huangc,Adaptive grid job scheduling with genetic algorithms, *Future Generation Computer Systems 21* (2005), pp.151–161.

[12] Lei Zhang, Yuehui Chen and Bo Yang, Task Scheduling Based on PSO Algorithm in Computational Grid, *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications* (ISDA'06).

[13] Xhafa, F., Gonzalez, J.A., Dahal, K.P., and Abraham, A., A GA (TS) Hybrid Algorithm for Scheduling in Computational Grids. In: Corchado, E., Wu, X., Oja, E., Herrero, A., Baruque, B. *(eds.) HAIS 2009. LNCS (LNAI)*, vol. 5572, 285–292, Springer, Heidelberg (2009).

[14] Ajith Abraham, Rajkumar Buyya and Baikunth Nath,Nature's Heuristics for Scheduling Jobs on Computational Grids, *in Proc. of 8th IEEE International Conference on Advanced Computing and Communications (ADCOM)*, pp. 45–52, India (2000).

[15] 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, no., pp.113-117, 3-4 Aug. 2008.

[16] 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, no., pp.2070-2073, 25-28 June 2006.

[17] J. Kolodziej, F. Xhafa, A Game-theoretic and hybrid Genetic meta-heuristics model for security-assured scheduling of independent jobs in Computational Grids, *in: Proceedings of the 4th International Conference on Complex, Intelligent and Software Intensive Systems* (CISIS-2010) February, 15th-18th 2010, Krakow, Poland.

[18] Roger Stough , Brian H. 32rts, *Regional economic development: analysis and planning strategy*, page no.202

[19] Li Liu, Yi Yang, Lian Li and Wanbin Shi,Using Ant Colony Optimization for Super Scheduling in Computational Grid**,** *Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing (APSCC'06)*.

[20] Ruay-Shiung Chang, Jih-Sheng Chang and Po-Sheng Lin,An ant algorithm for balanced job scheduling in grids, Future Generation Computer Systems 25 (2009), 20–27.

[21] Yuan-Shun Da and Xiao-Long Wang,Optimal resource allocation on grid systems for maximizing service reliability using a genetic algorithm, *Reliability Engineering and System Safety* 91 (2006), 1071–1082.

[22] 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 (13-15) (2002), 1507-1542.

[23] Anubhav Das and Daniel Grosu, Combinatorial Auction-Based Protocols for Resource Allocation in Grids, ipdps, vol. 14, pp.251a, 19th *IEEE International Parallel and Distributed Processing Symposium* (IPDPS'05) - Workshop 13, 2005.

[24] Li Li , Liu Yuan-an, Liu Kai-ming, Ma Xiao-lei and Yang Ming,Pricing in combinatorial double auction-based grid allocation model, *The Journal of China Universities of Posts and Telecommunications***,** June 2009, 16(3),59–65.

[25] Lijuan Xiao, Yanmin Zhu, Ni, L.M., Zhiwei Xu , Incentive-Based Scheduling for Market-Like Computational Grids, *Parallel and Distributed Systems, IEEE Transactions on* , vol.19, no.7, 903-913, July 2008..

[26] Daniel C. Vanderster, Nikitas J. Dimopoulos, Rafael Parra-Hernandez and Randall J. Sobie,Resource allocation on computational grids using a utility model and the knapsack problem, *Future Generation Computer Systems* 25 (2009),35–50.

[27] Jiang Wei-wei, Cui Hong-yan and Chen Jian-ya,A fuzzy modeling based dynamic resource allocation strategy in service grid, *The Journal of China Universities of Posts and Telecommunications*, September 2009, 16(Suppl.), 108–113.

[28] Y.M. Teo, X. Wang, and J.P. Gozali,A Compensation-based Scheduling Scheme for Grid Computing, *Proceedings of the Seventh International Conference on High Performance*

Computing and Grid in Asia Pacific Region (HPCAsia'04).

[29] Zhihui Du, Man Wanga, Yinong Chen, Yin Ye and Xudong Chai, The Triangular Pyramid Scheduling Model and algorithm for PDES in Grid, *Simulation Modelling Practice and Theory* 17 (2009) ,1678–1689.

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

[31] Jingbo Yuan, Shunli Ding, Cuirong Wang, Tasks Scheduling Based on Neural Networks in Grid, *Natural Computation, 2007. ICNC 2007. Third International Conference on* , vol.3, no., 372-376, 24-27 Aug. 2007

.