2010 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing

Game-theoretic, Market and Meta-Heuristics Approaches for Modelling Scheduling and Resource Allocation in Grid Systems

Fatos Xhafa
Department of Computer Science and Information Systems
Birkbeck, University of London
Malet Street, Bloomsbury, London WC1E 7HX, UK.
Email: fatos@dcs.bbk.ac.uk

Joanna Kołodziej
Department of Mathematics and Computer Science
University of Bielsko-Biała
ul. Willowa 2, Bielsko-Biała, Poland
Email: jkolodziej@ath.bielsko.pl

***Abstract*—Task scheduling and resource allocation are the crucial issues in any large scale distributed system, such as Computational Grids (CGs). However, traditional computational models and resolution methods cannot effectively tackle the complex nature of Grid, where the resources and users belong to many administrative domains with their own access policies, users' privileges, etc. Recently, researchers are investigating the use of game theoretic approaches for modelling task and resource allocation problems in CGs. In this paper, we present a compact survey of the most relevant research proposals in the literature to use game-based models for the resource allocation problems and their resolution using metaheuristic methods. We emphasize the need of the translation of the traditional economical models into the game scenarios and the use of metaheuristic schedulers for solving such games in order to address the new complex scheduling and allocation criterions. We study the case of asymmetric Stackelberg game used for modelling the Grid users' behavior, where the security and reliability criterions are aggregated and defined as the users' costs functions. The obtained results show the efficiency of the hybridization of heuristic-based approaches with game models, which enables to include additional requirements and features into the computational models and tackle more effectively the resolution of the applied schedulers.**

***Keywords*-Computational Grid, Scheduling, Non-cooperative Games, User behavior, Security, Meta-heuristic.**

I. Introduction

Grid Computing systems have become popular for the resolution of large-scale complex problems from science, engineering, finance, etc. Task scheduling and resource allocation are the crucial issues in Computational Grids (CGs). These issues are commonly investigated using traditional computational models and resolution methods that yield to near-optimal allocation strategies. One drawback of such approaches is that they cannot effectively tackle the complex nature of CGs. On one hand, such systems account for many administrative domains with their own access policies, user privileges, etc. On the other hand, CGs have a hierarchical nature and therefore any computational model should be able to effectively express the hierarchical architecture to the optimization model. Market and heuristic-based approaches usually partially cover the needs of the Grid-enabled applications. Recently, researchers are investigating the use of game theoretic approaches for modelling task and resource allocation problems in CGs.

The aim of this paper is to present a compact survey of the most relevant research proposals in the literature to use game-based models for the resource allocation problems and their resolution using meta-heuristic methods. Differently from the other comprehensive surveys (see e.g. [Buyya, 2009]) we focus mainly on the hybridization of market- and heuristic-based approaches with game models, which enables to include more requirements and features into the computational models and tackle more effectively the resolution of the applied schedulers. We study the case of asymmetric Stackelberg game used for modelling the Grid users' behavior, where the security and reliability criterions are aggregated into the scheduling cost functions specified for the users.

The remainder of the paper is organized as follows. In Sec. II we introduce preliminary concepts on Grid architecture, scheduling and resource allocation and game-theoretical models. In Sec. III we briefly survey most relevant research proposals in the literature for using game theoretic models for allocation problems. Then, in Sec. IV, we present some recent findings on using of game theoretic models for modelling allocation problems and their resolution using meta-heuristics. A case study on Stackelberg game applied to independent batch scheduling is presented in Sec. V. We end the paper in Section VI with some conclusions and indications for future work.

II. Preliminaries

A. Models and Architectures of Computational Grids

Because of the large ultimate scale of a CG, one of the most important issues is to design its appropriate architectural model, which allows an efficient management of the geographically distributed resources. Depending on the component organization, scheduling and resource management systems are usually classified into three different types, namely centralized, decentralized and hierarchical Grid models.

978-0-7695-4237-9/10 $26.00 © 2010 IEEE
DOI 10.1109/3PGCIC.2010.39

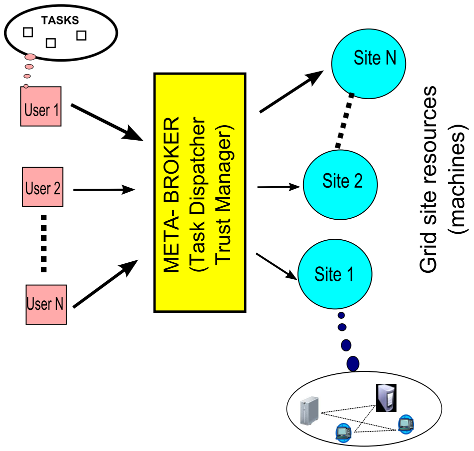

Figure 1. The model of secure-assured Grid.

In *centralized model*, there is a central authority, who has a complete knowledge of the system. The main disadvantages of this model are the limited scalability, lack of fault-tolerance, and the difficulty in accommodating local multiple policies imposed by the resource owners.

In *decentralized model*, local schedulers interact with each other to manage the tasks pool. In this model, there is no central authority responsible for resource allocation. Hence, the model naturally addresses issues such as fault-tolerance, scalability, site-autonomy, and multi-policy scheduling.

Finally, in the *hierarchical model*, there is a central meta-scheduler (or meta-broker), which interacts with local job dispatchers in order to define the optimal schedules. The local schedulers have knowledge about resource clusters, but they cannot monitor the whole system. The hierarchy in such a system usually consists of two or three levels.

An example of the hierarchical bi-level Grid is the Meta-Broker model (MB. In this model, Grid users submit their tasks/applications to the meta-broker, who uses also the information supplied by the resource owners to match the users' tasks/applications to appropriate resources.

Kwok et al. [Kwok et al., 2007] presented a hierarchical Grid model with three main levels: the global, the inter-site and the intra-site levels. At the intra-site level, there is a federation of autonomous machines. The resource owners send an information about the computational capacities of the machines to the local job dispatcher, which defines the "Grid site reputation index" and send it to the global scheduler, who performs the tasks scheduling according to a certain scheduling algorithm.

Recently, Kolodziej and Xhafa [Kołodziej et al., 2010] proposed an extension of the meta-broker model to define the security-assured grid site (see Fig. 1). In that approach the meta-broker plays additionally the role of the trust manager, which is responsible for the verification of the secure-assurance condition for any task-machine matching.

All examples of the hierarchical systems presented above can be easily translated into the game models in order to push the concept of CG into mainstream computing (market-based approaches) and to address the new requirements in Grid scheduling.

B. Scheduling and Resource Allocation in Computational Grids

Unlike traditional distributed systems, in CGs the users and distributed resource clusters work in different autonomous domains. Whereas Grid users require assurance on the level, type, and quality of service being provided by the resources, resource owners are usually concerned about maintaining the local control on how resources are being utilized. Thus, an effective mapping of computational tasks or data transfers to resources that meet the requirements (cost, performance, security and other service metrics) remains challenging.

Due to different demands imposed by Grid-enabled applications, scheduling can be defined as a family of global optimization problems. Its complexity comes from the number of objectives to optimize (single *vs*. multi-objective), the type of the environment (static *vs*. dynamic), the processing mode (immediate *vs*. batch), tasks interrelations, etc.

Among several types of scheduling problems, Independent Job Batch Scheduling is one of the simplest and fundamental scheduling problem version. It usually arises in data intensive computing such as data mining and massive processing applications. For this class of Grid-enabled applications, batch mode is appropriate task processing mode. In such a case, jobs or applications are grouped in *batches* and scheduled as a group.

The independent scheduling problem can be formalized using the Expected Time To Compute (ETC) matrix model [Ali et al., 2000]. In this model the following input data has to be specified:

- A number of independent tasks to be allocated to Grid resources in non-preemptive mode.
- A number of machines candidates to participate in the allocation of tasks.
- The workload (in millions of instructions) of each task.
- The computing capacity of each machine (in Mips).
- The ready times indicating when machines will have finished the previously assigned tasks.
- The ETC matrix of size $nb\_tasks \times nb\_machines$, where $ETC[j][m]$ is the value of the expected time to compute task j in machine m.

C. Game theory based scheduling and resource management models

Game theory is playing an important role in computer science, where it is being used as a means for modelling interactive computations or multi-agent systems. Recently, Internet computing is a new domain of applications of game theory, which in combination with economic theory can develop algorithms for finding equilibria in computational markets, computational auctions, Grid and P2P systems and secure information transfers.

There are three main types of game scenarios applied in CGs:

- **Non-cooperative game:** In this game, players act independently of each other. This model is based on the premise about the users' behavior in a realistic Grid, in which cooperation is difficult to happen at large scale. Grid users usually submit their tasks/applications to the Grid system independently of each other. Also, the resource owners act selfishly in order to maximize the resource utilization.
- **Cooperative game:** In this game scenario, players can form a coalition to plan in advance their actions. This model is useful for the intra-site Grid negotiations, where the local job dispatchers can define the joint "execution capabilities" parameters for the clusters of the Grid sites and declare them to the global scheduler.
- **Semi-cooperative game:** In this model, any player can choose (randomly) another player for cooperation. This game is usually proposed as a multi-round auction to incorporate the task rescheduling.

In the following two sections (Sec. III, Sec. IV) we present some well-known and recently proposed examples of Grid models, on which the above game scenarios are supported.

III. Game-theoretical Models and Market-based Approaches

A. Market-based Approaches

Computational economy is one of the most popular mechanism for the design of resource management architecture of Grid systems. It allows the Grid resource owners, acting as sellers, to earn money by letting others (mainly the Grid users, acting as buyers) use their (idle) computational resources. The pricing of resources is driven by demand and supply.

A number of economic models for Grid resource management has been proposed in the literature. We highlight in Table I three most popular approaches in Grid Computing, namely commodity market, auction and bargaining models. Each model can be easily modified to meet the Grid users' or resource owners' specified requirements. For example, the Posted Price Model [Buyya, 2009] is an extension of commodity market by considering posted prices (usually lower than regular ones) as a special offer to the Grid users.

The auction mechanism can be also defined in many ways (e.g. English, Dutch, Second Price auctions). All of them differ in terms of whether they are performed as open or closed auctions and the offer price for the highest bidder. Wolski et al. [Wolski et al., 2001] proposed a model called G-Commerce in which computational resources among different Grid sites are traded in a barter manner. This model can be interpreted as a combination of the commodity market and auction approaches. A comprehensive survey on traditional market-based models is presented in [Buyya, 2009].

B. Hybridization of Game Models and Market-based Approaches

Market-oriented approaches are suitable to exploit the interaction of different scheduling layers. However, Grid users and resource owners are likely to behave in different (selfish or/and cooperative) manners and their behavior cannot be characterized using conventional techniques. Game models are quite natural tools for solving this problem, because each market-based scenario can be easily translated into the game framework.

The mechanism of first price bidding auctions was applied by Kwok et al. [Kwok et al., 2007] to define the game-based resource management and global scheduling policy at the intra- and inter-site levels in the 3-levels hierarchical Grid structure. In the intra-site bidding each machine owner in the site declares the "execution capability" of the resource. The local job dispatcher moderates these amounts and sends a single value to the global scheduler. In the inter-site bidding the global scheduler should allocate tasks according to the values sent by the local dispatchers. The authors showed that the cooperation of the players at both levels are the optimal strategies for both games. In fact there is also a third game defined for the resource owners, who behave selfishly. The objective of this non-cooperative game is to maximize the resource utilization. The game scenarios at each level are very simple and the authors focused in fact on the optimization of two scheduling criteria: minimization of the task deadlines (user's requirement) and maximization of the resource utilization (resource owner's requirement). However, for the successful execution of all those games some synchronization mechanism must be introduced, which can make the whole system inefficient in a large-scale dynamic environment.

An early approach of modified auction mechanisms can be found in [Regev et al., 1998]. The authors defined POPCORN market for trading online CPU times. In their system, a virtual currency called "popcoin" was used between buyers and sellers communicating via internet. The social efficiency and price stability were studied using the Vickrey auction game. In this scenario cooperation between players to form a coalition and win the auction, is possible, but usually the players behave selfishly.

Ghosh et al. [Ghosh et al., 2004] study the load balancing issues in a mobile CG, in which there is a wireless access point (WAP) which mediates the requests from different mobile devices. The problem is modeled as a bargaining cooperative game between each mobile device and the WAP server. The solution of the whole game is the Nash Bargaining Solution (NBS). In this approach an explicit payment

Table I
MARKET-BASED MODELS FOR RESOURCE MANAGEMENT IN COMPUTATIONAL GRIDS

Model	Characteristic	Optimization criteria	Grid architecture model	Main References
Commodity Market	- pricing policies are based on the demand from the users and the supply of resources,	- minimizing the cost of the resource utilization (paid by the user under budget constraints) and the overall task execution time	Meta-Broker	[Garg et al., 2009] [Buyya et al. 2002]
Auctions	- offering buying and selling items for bids	- minimizing the highest bidder price (first price auction)	- All hierarchical models	[Ghosh et al., 2004]
	- managing the bids (auctioneers)	- maximizing the resource sellers pay-offs	- Decentralized models	[Buyya et al., 2000a]
Bargaining	- resource brokers bargain with resource providers for lower access price and higher usage duration	- minimizing the cost of the resource utilization	- All hierarchical models	[Regev et al., 1998]
	- the negotiation is guided by user requirements (e.g., deadline)	- maximizing the resource sellers pay-offs	- Decentralized models	[Subrata, et al., 2010]

scheme must be enforced in the system. A recent study on the bargaining cooperative game application in optimizing the energy consumption in Grid is proposed in [Subrata, et al., 2010].

IV. GAME-THEORETICAL MODELS AND HEURISTIC APPROACHES

A. Heuristic approaches

Due to multi-constraints and different optimization criteria in a dynamic environment, scheduling in Grids still remains a challenging, very complex and computationally hard global optimization problem. Thus, the heuristic approaches are candidates for effectively designing Grid schedulers. Meta-heuristic scheduler could achieve robustness and are appropriate for tackling the various scheduling attributes, like immediate and batch scheduling, multi-objectivity, decentralized and hierarchical Grid architectures, etc.

The heuristic methods are usually classified into three main groups, namely calculus-based (greedy algorithms and ad-Hoc methods), stochastic (guided and non-guided methods) and enumerative methods (dynamic programming and branch-and-bound algorithm). The most popular and efficient methods in Grid scheduling are ad-hoc, local search-based and population-based methods. We briefly review them in Table II.

Another important feature of meta-heuristics, which is useful in Grid scheduling, is that they can be easily hybridized with other approaches. It makes the Grid schedulers better adapted to the various Grid types and specific types of applications. Abraham et al. [Abraham et al., 2000] present a model for hybridization of GA, SA and TS heuristics; each GA-based hybrid, namely GA+SA and GA+TS improves the efficiency of the genetic scheduler. Ritchie and Levine [Ritchie, and Levine, 2003] combine an ACO with a TS algorithm for the problem.

B. Hybridization of Game Models and Heuristic-based Approaches

As Grid scheduling problem is multi-objective in its general formulation, meta-heuristic algorithms must deal with various conflicting optimization criteria at the same time. Certainly, this could increase their computation cost. For example, classical GA-based schedulers usually do not consider in the same time many QoS constraints of concurrent users such as budget and job deadlines. Thus, an single objective is usually defined as a weighted function aggregating just two main schedulers performance metrics, namely makespan and flowtime. Of course it does not cope with other more complex scheduling scenarios, where additional users' and service providers' requirements, such as security and reliability of the resources, have to be considered.

To overcome such limitations, the modified versions of classical meta-heuristics, like Multi-Objective Genetic Algorithm (MOGA), have been proposed. The other approaches addressed for multi-objective problems, which has recently attracted researchers' attention in the literature, are based on the idea of using game models for scheduling problems in combination with different meta-heuristics for solving the resulting games.

An application of modified heuristic methods in the non-cooperative game of Grid users in the commodity market model is presented in [Garg et al., 2009]. The authors define two heuristics, namely Min-Min Cost Time Trade-off (MinCTT) and Max-Min Cost Time Trade-off (Max-CTT) algorithms, for optimizing jointly cost and execution time of user application in Utility Grids. Both methods are based on the framework of Min-Min and Max-Min methods, which were adapted to the 'user application/time slot' pairs. This approach is an example of the Grid users' game, in which the task execution and resource utilization costs are defined as the bi-objective players' cost functions.

In [Song et al., 2006] the authors considered the risky and

Table II

SELECTED HEURISTIC AND META-HEURISTIC METHODS IN GRID SCHEDULING

Meta-heuristic class	Class characteristic	Scheduler type	Methods	Main references
Ad-hoc	- used in single-objective optimiz. - low computational cost - useful in generating the initial solutions for population-based schedulers	Immediate mode	*Minimum Completion Time (MCT)* *Minimum Execution Time* *Opportunistic Load Balancing*	[Braun et al., 2001]
		Batch mode	*Min-Min, Max-Min, Sufferage* *Relative Cost, LJFR-SJFR*	[Xhafa et al., 2007]
Local search	- explore the solution space starting from an initial solution - construct a path in solution space		*Hill Climbing* *Simulated Annealing* *Tabu Search*	[Yarkham et al., 2002] [Abraham et al., 2000] [Xhafa, et al., 2009a]
Population-based	- explore of the search space by the populations of individuals - require a large running time - effective in finding near-optimal solutions	Single population	*Genetic Algorithms (GAs)* *Memetic Algorithms (MAs)* *Particle Swarm Opt.(PSO)* *Ant Colony Optimization (ACO)*	[Abraham et al., 2000] [Liu et al., 2009]
		Multi-population	*Hierarchic Genetic Strategy (HGS)* *Grid-Enabled Hierarchical Parallel Genetic Algorithm (GE-HPGA)*	[Kołodziej et al., 2009] [Lim et al., 2007]

insecure conditions in online scheduling in Grids caused by software vulnerability and distrusted resources. They applied a game-theoretical model introduced in [Kwok et al., 2007] for defining the resource owners selfish behavior in the hierarchical Grid structure. A Space Time Genetic Algorithm (STGA) was implemented as main mechanism of three risk-resilient meta-heuristics, named as risky, preemptive and replicated STG algorithms.

The results presented in [Song et al., 2006] were extended by Wu et al. in [Wu et. al, 2010] by considering the heterogeneity of fault-tolerance mechanism in a security-assured Grid job scheduling. The authors defined four types of GA-based online schedulers for the simulation of some fault-tolerance mechanisms, including job retry, job migration (with and without checkpointing) and job replication mechanisms.

It can be observed that many scheduling and resource management criteria, like security and resource utilization and reliability, are addressed separately in most of current approaches, while, because of the complex nature of Grid systems, it is necessary to integrate all those features into the Grid schedulers. Thus, we defined in our recent work [Kołodziej et al., 2010] a non-cooperative Grid users' game model for addressing the requirements for the security and resource reliability in Grid scheduling. A simple case study on Stackelberg asymmetric game approach is presented in the following section.

V. A CASE STUDY: NON-COOPERATIVE ASYMMETRIC GRID USERS' GAME FOR INDEPENDENT BATCH SCHEDULING

To meet additional scheduling requirements, like security and resource reliability, we adapt a general concept of the Meta-broker, which is in our approaches responsible for checking the security condition and the resource availability. The scheduling problem has been translated into the non-cooperative game of Grid users. This game model takes into account the realistic feature that Grid users usually act independently.

In this case study we define the asymmetric Stackelberg game for modelling the users behavior, in which one user acts as a *Leader* and the rest of players (users) are his *Followers*. The definition of the Leader in this game arises naturally in the situation where a user has the full access to the resources as opposed to the other users having the limited access. The game is defined for solving the independent scheduling problem, in which tasks are proceeded in the batch mode. The Leader in our approach is the owner of a large portion of the tasks in the batch.

Game scenario: Formally the Stackelberg game for the Grid users can be defined as as two-level game in the following way:

- **Leader's Level: Leader's action I -** Leader chooses his initial strategy. His strategy vector $\widehat{x^1} = [\widehat{x_1}, \dots, \widehat{x_{k_1}}]$ represents Leader's initial action, and k_1 denotes the number of tasks submitted by the Leader.
- **Followers' Level: Followers' action -** Each Follower minimizes his cost function relative to the Leader's strategy:

$$\begin{cases} x_F^2 = arg\min_{x^2 \in J_2} \{Q_2(\widehat{x^1}, x^2, \dots, x^N)\} \\ \vdots \\ x_F^N = arg\min_{x^N \in J_N} \{Q_N(\widehat{x^1}, \dots, x^N)\} \end{cases} \tag{1}$$

where $\{J_1, \dots, J_N\}; l = 1, \dots, N$ are the sets of strategies of the users (N is the number of players), and $\{Q_1, \dots, Q_N\}; Q_l : J_1 \times \dots \times J_N \to \mathbb{R}^+; \forall l = 1, \dots, N$ is the set of users' game cost functions. We assume that J_1 is the set of the Leader's strategies. Let us denote by $x_F = [\widehat{x^1}, x_F^2, \dots, x_F^N]$

the vector which is interpreted as the result of the Followers' action.

- **Leader's Level: Leader's action II -** Leader updates his strategy by minimizing his cost function Q_l taking into account the result of Followers' action. The following vector $x_G = [x_L, x_F^2, \ldots, x_F^N]$, where:

$$x_L = arg \min_{x^1 \in J_1} Q(x^1, x_F^2, \ldots, x_F^N) \tag{2}$$

is a solution of the whole game.

It has to be noted that the Followers play an ”ordinary” non-cooperative game, but they must know the Leader's action first. An optimal solution of the whole game is called *Stackelberg Equilibrium*.

The players cost functions are composed of the following two factors:

$$Q_l = Q_l^{(s)} + Q_l^{(ab)}, \tag{3}$$

where: $Q_l^{(s)}$ indicates the cost of security-assured allocation of the user tasks, $Q_l^{(ab)}$ is the cost of possible abortion of the user's task due the resource unavailability, and denotes a resource utilization cost.

The values of the function $Q_l^{(s)}$ depend on the scheduling strategy. We consider in this work two scheduling strategies:

- **risky mode** - in which all risky and failing conditions are ignored by the users. In this case $Q_l^{(s)} = 0$, $l = 1, \ldots, N$.
- **secure mode** - in which $Q_l^{(s)}$ function is defined as follows:

$$Q_l^{(s)} = \sum_{j=(k_1+\ldots+k_{l-1}+1)}^{(k_1+\ldots+k_l)} \frac{P_f[j][x_j] \cdot ETC[j][x_j]}{(ETC)_{m(l)} \cdot k_l}, \tag{4}$$

where $ETC[j][x_j]$ is an element of the matrix *ETC* and $(ETC)_{m(l)}$ is the (expected) maximal computation time of the tasks of the user l in a given schedule.

We denoted by $P_f[j][x_j]$ in Eq. 4 the probability of the failure of machine x_j during the execution of task j. This probability is usually modelled by the negative exponential distribution:

$$P_f[j][x_j] = \begin{cases} 0 & , \quad sd_j \leq tl_{x_j} \\ 1 - e^{-\lambda(sd_j - tl_{x_j})} & , \quad sd_j > tl_{x_j} \end{cases} \tag{5}$$

where λ is interpreted as a failure coefficient and is a global parameter of the model, and sd_j and tl_{x_j} are the elements of *security demand* SD and *trust level* TL vectors ([Song et al., 2006]).

The cost of user's tasks abortion $Q_l^{(ab)}$ is defined using the following formulae:

$$Q_l^{(ab)} = \frac{\sum_{j=(k_1+\ldots+k_l+1)}^{(k_1+\ldots+k_l)} P_{ab}(j) \cdot ETC[j][x_j]}{(ETC)_{m(l)} \cdot k_l}, \tag{6}$$

where $P_{ab}(j)$ is a task abortion probability, which is calculated as $P_{ab}(j) = (1 - P_{x_j})$, where P_{x_j} denotes a reliability probability specified for a given machine x_j.

The aim of playing the game is the minimization of the *joint cost function*, which is defined as follows:

$$Q = Q_1(x^1, x_F^2, \ldots, x_F^N), \tag{7}$$

where x^1 denotes the vector of Leader's decision variables and $x_F^2, \ldots, x_F^N$ are interpreted as the results of the minimization of the Followers' cost functions. This function must be minimized at both Leader's and Followers' levels.

GA-based hybrid schedulers: For solving the game we defined the genetic-based hybrid meta-heuristic, which combines GAs and a modified version of Minimum Completion Time method, namely PMCT algorithm. Each component of the hybrid, i.e. GA and PMCT, operates as a scheduler at the global (Leader's) and Followers' levels, respectively. Each player (including Leader) operates just on his own decision variables. In the initialization process Leader defines an initial population for the GA-based global scheduler, which contains just the values of the Leader's decision variables. All those “incomplete” chromosomes are sent to the Followers.

At the Followers' level we implemented PMCT algorithm, which is a modification of Minimum Completion Time (MCT), which is applied to the tasks of the individual Followers separately. Firstly, any incomplete schedule received from the Leader's module is scanned and the completion times of the machines are computed. Then MCT procedure, restricted to just the tasks of a given Follower, is executed. The process is repeated (sequentially for all Followers) for all schedules (chromosomes) received from the GA-scheduler.

Experimental analysis: In the experimental evaluation of the proposed models we considered two scenarios: *risky* and *secure* mode. We then consider two hybrid metaheuristics: Secure GA-PMCT and Risky GA-PMCT. For this we integrated the schedulers with the discrete event-based Grid simulator *HyperSim-G* [Xhafa et. al., 2009b]. The experiments were conducted on two benchmarks composed by a set of static and dynamic instances. In both static and dynamic cases four Grid size scenarios are considered: small (32 hosts/512 tasks), medium (64 hosts/1024 tasks), large (128 hosts/2048 tasks), and very large (256 hosts/4096 tasks). The settings for the simulator are presented in Table III.

We set the number of evolution steps for the GA at Leader's level to 2000, the population size to 68 the crossover and mutation probabilities to 0.9 and 0.2 respectively. There are 15 Followers in our game and the number of the Leader's tasks is a half of the whole task batch. The coefficients of SD, TL vectors and the machines reliability probabilities P_{x_j} are defined as the uniformly generated fractions in the ranges [0.6;0.9], [0.3;1] and [0.85;1] respectively. The value of failure coefficient λ is 3.

Table III
SETTING FOR THE GRID SIMULATOR FOR STATIC AND DYNAMIC CASES

	Small	Medium	Large	Very Large
	Static setting			
Nb. of hosts	32	64	128	256
Resource cap. (in MIPS)	$N(1000, 175)$			
Total nb. of tasks	512	1024	2048	4096
Workload of tasks	$N(250000000, 43750000)$			
	Dynamic setting			
Init. hosts	32	64	128	256
Max. hosts	37	70	135	264
Min. hosts	27	58	121	248
Resource cap. (in MIPS)	$N(1000, 175)$			
Add host	$N(625000, 93750)$	$N(562500, 84375)$	$N(500000, 75000)$	$N(437500, 65625)$
Delete host	$N(625000, 93750)$			
Total tasks	512	1024	2048	4096
Init. tasks	384	768	1536	3072
Workload	$N(250000000, 43750000)$			
Interarrival	$E(7812.5)$	$E(3906.25)$	$E(1953.125)$	$E(976.5625)$

To evaluate the scheduling performance we used *makespan* and *flowtime* metrics (see [Xhafa et al., 2010]). Each experiment was repeated 30 times under the same configuration and the averaged results are presented in Fig. 2.

The obtained results suggest that it is worth for the Grid users to pay some additional cost of the verification of the security conditions and possible task abortion in order to achieve an efficient allocation of tasks to the trustful and reliable resources.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have addressed the need of using new computational paradigms to solve efficiently the scheduling and resource allocation problems in Computational Grids. By surveying most important approaches in the literature for the problem, we observed that the proposed approaches fail to effectively cast additional requirements of Grid scheduling such as security and reliability of resources. Also, in Computational Grids different user behaviors are present such as symmetric, asymmetric, selfish, etc. Game-theoretic approaches have shown to be effective to translate such requirements and behaviors into the computational model for Grid scheduling. We have presented some recent findings on the use of game-theoretic for modelling allocation problems and their resolution using meta-heuristic methods. In the case study the Stackelberg game is implemented for the asymmetric user's behavior and it is solved by using the hybrid heuristic methods. The implemented approach has been tested using a Grid simulator showing the efficiency of the hybridization of heuristic-based approaches with game models, which enables to include more requirements and features into the computational models and tackle more effectively the resolution of the applied schedulers.

REFERENCES

[Abraham et al., 2000] Abraham, A., Buyya, R., and Nath B.: "Natures heuristics for scheduling jobs on computational grids", *Proc. of the 8th IEEE ACC*, India, 2000.

[Ali et al., 2000] Ali, S., Siegel, H.J., Maheswaran, M., and Hensgen, D.: "Task execution time modeling for heterogeneous computing systems", *Proceedings of Heterogeneous Computing Workshop*, pp. 185–199, 2000.

[Braun et al., 2001] Braun, T.D., Siegel, H.J., Beck, N., Boloni, L.L., Maheswaran, M., Reuther, A.I., Robertson, J.P., Theys, M.D., Yao, B., Hensgen, D., and Freund, R.F.: "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(6): 810-837, 2001.

[Buyya et al. 2002] Buyya, R., Abramson, D., Giddy, J., and Stockinger,H.: "Economic Models for Resource Management and Scheduling in Grid Computing", *Journal of Concurrency and Computation: Practice and Experience (CCPE)*, Wiley Press, USA, May 2002.

[Buyya et al., 2000a] Buyya, R., Abramson, D. and Giddy, J.: "An Economy Driven Resource Management Architecture for Global Computational Power Grids",*Proceedings of PDPTA 2000*, Las Vegas, USA, 26-29 June 2000.

[Buyya, 2009] Buyya, R., and Bubendorfer, K. (Eds.): *"Market Oriented Grid and Utility Computing"*, Wiley Press, USA, 2009.

[Garg et al., 2009] Garg, S.K., Buyya, R., and Segel, H.J.: "Scheduling Parallel Aplications on Utility Grids: Time and Cost Trade-off Management", *In Proc. of the 32nd ACSC*, Wellington, Australia, 2009, CRPIT, Vol. 91, Bernard Mans Ed.

[Ghosh et al., 2004] Ghosh, P., Roy, N., Basu, K., and Das, S.K.: "A Game Theory based Pricing Strategy for Job Allocation in Mobile Grids", *Proc. of the 18th IEEE International Parallel and Distributed Processing Symposium (IPDPS 04)*, Santa Fe, New Mexico, 2004.

[Kołodziej et al., 2009] Kołodziej, J., Xhafa, F., Kolanko, Ł.: "Hierarchic Genetic Scheduler of Independent Jobs in Computational Grid Environment", *Proc. of 23rd ECMS, Madrid, 9-12.06.2009, in J. Otamendi, A. Bargieła, J.L. Montes and L.M. Doncel Pedrera eds.*, IEEE Press, Dudweiler, Germany, 2009, pp. 108–115.

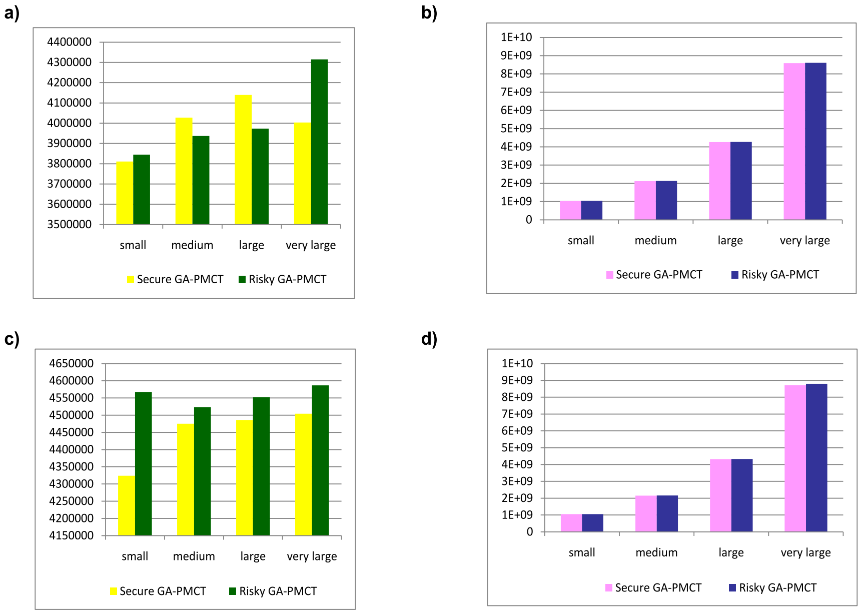

Figure 2. Experimental results achieved by four hybrid schedulers: in static case - (a) average makespan, (b) average flowtime ; in dynamic case - (c) average makespan, (d) average flowtime.

[Kołodziej et al., 2010] Kołodziej, J. and Xhafa, F.: "A Game-Theoretic and Hybrid Genetic meta-heuristic Model for Security-Assured Scheduling of Independent Jobs in Computational Grids", *Proc. of CISIS 2010*, IEEE Press, , USA, 2010, pp. 93–100.

[Kwok et al., 2007] Kwok,Y.-K., Hwang, K. and Song, S.: "Selfish Grids: Game-Theoretic Modeling and NAS/PSA Benchmark Evaluation", *IEEE Transactions on Parallel and Distributing Systems*, Vol. 18 No. 5, 1-16, 2007.

[Lim et al., 2007] Lim, D., Ong, Y.-S., Jin, Y.: "Efficient Hierarchical Parallel Genetic Algorithms Using Grid Computing", *Future Generation Computer Ssytem*, 23(4): 658–670, 2007.

[Liu et al., 2009] Liu, H., Abraham, A., and Hassanien, A.E.: "Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm", *Future Generation Computer Ssytem*,doi:10.1016/j.future.2009.05.022.

[Regev et al., 1998] O. Regev and N. Nisan: "The POPCORN Market - an Online Market for Computational Resources", ICE 1998

[Ritchie, and Levine, 2003] Ritchie, G., and Levine J.: "A fast effective local search for scheduling independent jobs in heterogeneous computing environments",*TechRep*, Centre for Intelligent Systems and Their Applications, University of Edinburgh, 2003.

[Song et al., 2006] Song, S., Hwang, K., and Kwok, Y-K.: "Risk-resilient Heuristics and Genetic Algorithms for Security- Assured Grid Job Scheduling, *IEEE Transactions on Computers*, vol. 55 no 6, pp. 703-719, 2006

[Subrata, et al., 2010] Subrata, R., Zomaya, A. Y., and Landfeldt, B.: "Cooperative power-aware scheduling in grid computing environments", *J. Parallel Distrib. Comput.*, 70 (2010) 84–91.

[Wolski et al., 2001] Wolski, R., Plank, J. S., Bryan, T. and Brevik, J.: "G-Commerce: Market Formulations Controlling Resource Allocation on the Computational Grid"' *Proc. IPDPS*, 2001.

[Wu et. al, 2010] Wu, C.-C., and Sun, R.-Y.: "An integrated security-aware job scheduling strategy for large-scale computational grids", *Future Generation Computer Systems*, 26 (2010), pp. 198–206.

[Xhafa et al., 2007] Xhafa, F., Barolli, L. and Durresi, A.: "Batch Mode Schedulers for Grid Systems", *International Journal of Web and Grid Services*, 3(1): 19–37, 2007.

[Xhafa et. al., 2009b] Xhafa, F., and Carretero, J.: "Experimental Study of GA-Based Schedulers in Dynamic Distributed Computing Environments", Chapter 24, In *Alba et al. Eds. Optimization Techniques for Solving Complex Problems*, Wiley, 2009.

[Xhafa, et al., 2009a] Xhafa, F., Carretero, J., Alba, E., and Dorronsoro, B.: "Tabu Search Algorithm for Scheduling Independent Jobs in Computational Grids", *Computer And Informatics Journal,special issue on "Intelligent Computational Methods"*, J.Burguillo-Rial, J.Kołodziej and L. Nolle eds., Vol. 28, No 2, 2009, pp 237–249.

[Xhafa et al., 2010] Xhafa, F., Abraham, A.: "Computational models and heuristic methods for Grid scheduling problems", *Future Generation Computer Systems*, 26 (2010), pp. 608–621.

[Yarkham et al., 2002] Yarkhan, A., and Dongarra, J.: "Experiments with scheduling using simulated annealing in a grid environment", *Proc. of the 3rd International Workshop on Grid Computing*, pp. 232–242, 2002.