COMPUTATIONAL METHODS IN SCIENCE AND TECHNOLOGY **12(2)**, 123-138 (2006)

Scheduling Jobs on the Grid – Multicriteria Approach

Krzysztof Kurowski[1*], Jarek Nabrzyski[1*], Ariel Oleksiak[1*], Jan Węglarz[1,2]**

[1]*Poznan Supercomputing and Networking Center*
[2]*Institute of Computing Science, Poznan University of Technology*

e-mail: {krzysztof.kurowski/naber/ariel}@man.poznan.pl[*], Jan.Weglarz@cs.put.poznan.pl[**]

(Rec. 28 August 2006)

Abstract: In the paper we present two different models of Grid resource management problems: (i) Grid scheduling problems with no time characteristics available, and (ii) scheduling of jobs in presence of time characteristics achieved by using some prediction techniques, and resource reservation mechanisms. We focus on demonstrating how these two scenarios, which are important examples of Grid environments, can be modeled as multi-criteria decision support problems. We also discuss advantages and disadvantages of these models as well as practical applications.

Key words: Grid computing, Grid resource management and scheduling, multicriteria decision support

1. INTRODUCTION

The term Grid was coined in the mid-1990s to describe a technological vision of a shared computing infrastructure for researchers. This vision was later refined and described the Grid as an infrastructure for 'coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations' [12]. Recently the Network of Excellence CoreGRID [1] produced a definition of Grid as 'a fully distributed, dynamically reconfigurable, scalable and autonomous infrastructure to provide location independent, pervasive, reliable, secure and efficient access to a coordinated set of services encapsulating and virtualizing resources (computing power, storage, instruments, data, *etc*.) in order to generate knowledge'.

A set of individuals and/or domains defined by common resource sharing rules forms a VO [13]. As pointed out in [20] the objectives of *users*, *resource providers* and *VO administrators*, *i.e.* Grid resource management and scheduling processes *stakeholders* (or *decision makers*), may often be inconsistent or even in conflict with each other. In our case we consider a user as an entity who utilizes available resource(s) by running his/Her computing jobs or performing any operations consuming resource unit(s) for a certain period of time. On the other hand, a resource provider is an entity that manages computational units (single computers or local resource management systems) within a single administrative domain. Both users and resource providers can be organized dynamically into a VO, each with different policy requirements and preferences. Finally, a VO administrator is an entity responsible for maintaining and control of cross-domain resource sharing policies and security rules. The VO may range from small, ad-hoc groups of stakeholder that exist for only a short period of time, to larger, long term VO structures. In both cases, the expression of resource policies may also change frequently. Therefore, the dynamic data describing jobs and resources and the knowledge about the preference structure of many decision makers in Grid environments are key factors that must be considered during problem and model formulation.

Grid computing requires the use of specialized middleware to mitigate the complexity of integrating distributed resources within an Enterprise or any Virtual Organization (VO). This middleware usually consists of such services, as data management, information and monitoring, brokering and resource management, authorization, accounting and other ones. One of the central Grid middleware services are these responsible for resource management, brokering and scheduling.

Grid resource management and *scheduling* [27] is defined as the process of identifying requirements, matching resources to computing jobs (computing applications), allocating those resources, and scheduling and monitoring Grid resources over time in order to run Grid applications as efficiently as possible. Grid applications compete for resources that are very different in nature, including processors, data, scientific instruments, networks, and other ones. Complicating this situation is the lack of detailed and current information about the system and the competing needs of users and resource providers (administrators of local resource management systems).

Grids are becoming almost commonplace today, with many projects using them for production runs. The initial challenges of Grid computing – how to run a job, how to transfer large files, how to manage multiple user accounts

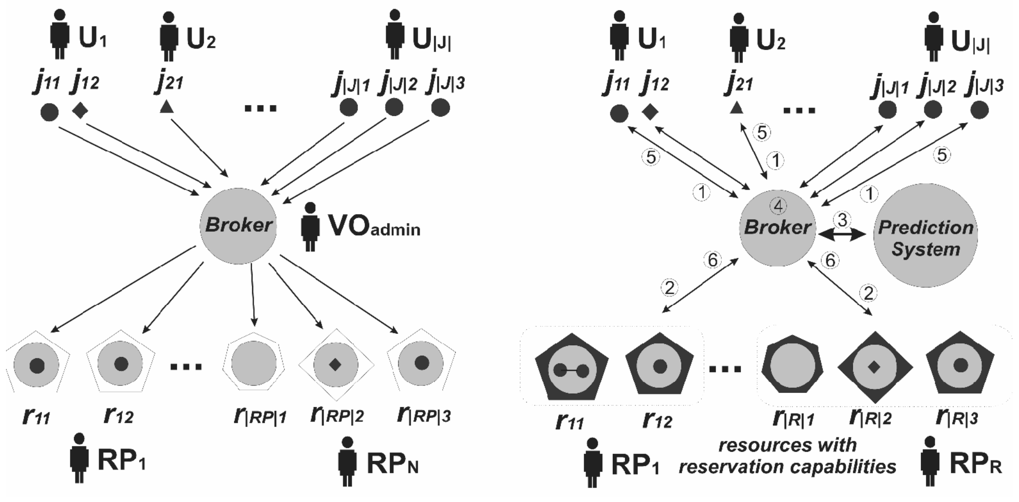


Fig. 1. Two different Grid environments (i) and (ii) together with main players and characteristics

on different systems – have been resolved to first order, so users and researchers can now address the issues that will allow more efficient use of the resources. While Grids have become almost commonplace, the use of good Grid resource management tools is far from ubiquitous because of the many open issues in the field. Some of the issues include:

- **Multiple layers of schedulers**. Grid resource management involves many players and possibly several different layers of schedulers. At the highest level are Grid-level schedulers that may have a more general view of the resources but are very 'far away' from the resources where the users' jobs will eventually run. At the lowest level is a local resource management system that manages a specific resource or set of resources.
- **Lack of control over resources**. Grid-level scheduler usually does not have ownership or control over the resources. Most of the time, jobs will be submitted from a higher-level Grid scheduler to a local set of resources with no more permissions than the user would have.
- **Shared resources and variance**. Related to the lack of control is the lack of dedicated access to the resources. Most resources in a Grid environment are shared among many users and projects. Such sharing results in a high degree of variance and unpredictability in the capacity of the resources available for use. The heterogeneous nature of the resources involved also plays a role in varied capacity.
- **Conflicting performance goals**. Often, resource providers and users have different performance goals: from optimizing the performance of a single application for a specified cost goal to getting the best system throughput or minimizing response time. In addition, most resources have local policies that must be taken into account.

The level of existence of each of the issues mentioned above depends strongly on specific usage scenarios, and, very often, on particular Grid core infrastructure. Thus, it is very difficult to provide a single scheduling method that could be used in many different settings. In [20], general multi-criteria decision support methodology for Grid resource management was presented. The work assumed that all time characteristics of jobs were known a priori. Variance of resources and jobs was ignored. This is of course an idealistic situation, never met in real environments.

In the current paper we investigate multi-objective Grid scheduling problems for two cases: (i) Grid scheduling problems with no time characteristics, and (ii) scheduling of Grid jobs in presence of time characteristics coming from prediction mechanisms and from resource reservation mechanisms.

In the first case a resource broker does not have access to any knowledge about job start and execution times. Decisions concerning allocation of resources are made based on security and resource management policies, dynamic descriptions of resources and job requirements. Such environments are the most common among current Grid infrastructures. The second case assumes availability of information on job start and execution times by means of mechanisms such as advance resource reservation and job runtime and queue waiting time prediction. There are many attempts towards this approach [6, 34, 22, 35] in Grid research.

In the next sections, we focus on demonstrating how these two scenarios, which are important examples of Grid environments, can be modeled as a multi-criteria decision support problems. We show how in both cases stakeholders' requirements and preferences can be expressed using hard and soft constraints concepts described in [20]. However, some additionally unique and specific features together with advantages and disadvantages of these models are described.

2. RELATED WORK

Various projects attempt to solve the problems of Grid resource management and scheduling, and provide certain features that are also taken into account in our models. In this section we survey related work in mentioned areas. One of the well known Grid schedulers is Workload Manager that has been developed as a part of the Enabling Grids for E-sciencE (EGEE) project [10]. Workload Manager by default is configured according to a FCFS policy for a given user, so all the users' jobs are opportunistically scheduled for execution on available resource providers that may reside at several different Grids. As it is discussed in [26] Grid-level management processes can be improved by applying different FCFS queues in the Grid broker. Another example is the work in [3], which studies the problem of Grid scheduling processes based on Community Scheduler Framework (CSF) [4]. CSF is an open source framework for implementing a Grid metascheduler which provides an environment that can dispatch users jobs to various underlying resource providers in a system independent fashion (over fabric Globus Toolkit based services). CSF by default offers only simple scheduling and reservation capabilities, provided that all jobs are submitted to resource providers via CSF. Authors in [17] describe a relatively new Grid broker called GridWay that gives a user an easier and more efficient execution of Grid jobs in a *submit* and *forget* fashion. In order to obtain a reasonable degree of performance, job execution is adapted to dynamic resource conditions and application demands. Internal GridWay scheduling mechanisms are relatively simple and are based on the "greedy approach", implemented by the round-robin algorithm. The current release of GridWay supports any form of multi-domain scheduling.

As mentioned in the previous section, in spite of many issues related to the use of advance resource reservation and time prediction mechanisms, there were some attempts to apply scheduling methods based on knowledge of job completion times in Grids. They can be categorized into those focused on the use of performance prediction techniques and those assuming availability of resource reservation facilities. One of the first and major work belonging to the former category brought results in a form of prediction methods based on a concept of job templates [33] and their use in Grid scheduling [35]. The estimated times were used to select resources for jobs. The same authors published their results concerning scheduling with reservations [35]. They made interesting observations on the influence of reservations on traditional scheduling. Authors used some simple list algorithms such as FCFS and LWF with backfilling mechanism. In [9] authors assume knowledge of job completion times, too. Moreover, they use multicriteria optimization to schedule jobs in the Grid. However, they assume two static global criteria for evaluation of the whole schedules: C_{MAX} and the weighted minimal average completion time (minsum). No individual stakeholders' preferences are considered. In addition to these scientific results concerning scheduling algorithms we must note that more and more work has been done on design and implementation of protocols and mechanisms providing means for advance reservation and service level agreements negotiations. For example, these ongoing efforts include works on protocols and description language for resource reservation, service level agreements, and negotiations, all done within the Global Grid Forum (GGF) [14] by Grid Resource Allocation and Agreement Protocol (GRAAP-WG) group [15]. This group has recently published the WS-Agreement specification that defines language and protocol for advertising the capabilities of resource providers, checking how offers fit to each other, and establishing agreements based on these offers. Many existing projects and research groups started to incorporate these mechanisms into their resource management tools (for instance in [10]).

In general a lot of efforts have been made on Grid scheduling in environments similar to the both models presented in this paper. Nevertheless most of these methods are based on simple list procedures (*e.g.* FCFS, SJF), backfilling, matchmaking and load balancing. There are no solutions that fully support expression and exploitation of stakeholders' preferences allowing definition of multiple evaluation criteria and multicriteria optimization.

3. MULTICRITERIA GRID RESOURCE MANAGEMENT AND SCHEDULING

Grid resource management and scheduling is multicriteria in nature. This is so, because there exist various players that usually use different criteria and have different preferences involved in the management of Grid resources. They need to share and utilize resources in a highly controlled and secure way and all conditions under which sharing occurs are clearly defined. Unfortunately, well-studied resource management strategies in traditional computing and manufacturing systems do not apply well to the aforementioned environments. Traditional computing resources are exposed by resource providers through well-defined remote interfaces of computing services on which users perform certain operations, in particular job submission, control and monitoring. For this reason, a lot of efforts around resource management in Grids have so far been focused only on the definition of standard resource management protocols exposing resources remotely as well as standard mechanisms for expressing resource and job requirements [5, 6].

The multicriteria approach while solving the job scheduling and resource management problems fits well to such Grid environments. Here we sketch out some basic assumptions and problem characteristics that are common for both considered models:

- Different stakeholders (decision makers) are involved in resource management process, namely users, resource providers and a VO administrator,
- Stakeholders may have different constraints, criteria and preferences. All hard constraints defined by them must be taken into account before a final decision about job assignment or scheduling is made,
- All criteria and preference models of all stakeholders are known in advance (provided explicitly by all stakeholders or automatically deduced and exploited from the past stakeholders' decision actions),
- There is no optimal solution because various criteria are considered,
- Resource management problem we consider here is a deterministic one.

The main goal is to meet all requirements in order to maximize a satisfaction (by finding a global compromise solution) of various stakeholders by assigning jobs to available resources. A list of additional resource and job characteristics assumed in both models are presented below:

- Each job can be assigned only to one resource,
- Jobs and resources are heterogeneous,
- Jobs' requirements and resource policies are known in advance,
- Resources are renewable,
- Resources are discrete, however attributes describing resources could have different domains,
- Both jobs and resources form finite sets.

Let us emphasize here that in real Grid environments the assignment of jobs to resources involves various and often complex technical operations, such as resource discovery, file staging etc. As many Grid environments are in constant flux, stakeholders can not be expected to keep up with the configuration and status of the Grid. This information must be provided with automatic and dynamic resource brokering for discovering, ranking and selecting resources that meet constraints and preferences. Therefore, an additional entity, called *broker*, *meta-scheduler* or *resource management system* is often introduced. Its role is to mediate access to shared resources on behalf of users. On the other hand, the broker also represents resource providers. It has a knowledge about available jobs and resources as well as it may enforce all security and resource policies controlled by a VO administrator. One of the most important assumptions here is that a final *decision* (assignment of jobs to resources) and *enforcement* are made by the broker. In the rest of the paper we discuss two different models, one with no time characteristics and the other one with available time characteristics of jobs. Both models are based on multicriteria analysis. The models are implemented within the Grid Resource Management System (GRMS) [21] as two of several other options for scheduling jobs in GRMS.

3.1 Preference modeling

There are various ways of preference modeling that can be adopted in Grid resource management [20]. In general, we can distinguish two ways of preferences acquisition: (i) preferences are given explicitly by stakeholders, or (ii) stakeholders preferences are discovered on the basis of their previous decisions (*e.g.* comparison of potential solutions). In situation (i) a stakeholder specifies an importance of every single criterion or a relative importance between criteria (based, for instance, on the utility theory or lexicographic order of criteria). This can be done by a definition of criteria weights or criteria ranking (sorting them from the most important to the least important ones). In case (ii), first, in a learning phase a stakeholder can, for instance, retrieve a set of possible solutions (that meet requirements and are Pareto-optimal) and rank them. A resource broker can learn stakeholders' preferences using *e.g.* approaches based on generation of decision rules based on rough sets [32]. When sufficient knowledge has been gathered a resource broker can decide automatically which solutions are the best according to the modeled preferences. A choice of the method depends on two major aspects: first, whether stakeholders are aware of their preferences and know how to express them, and, second, whether their preferences are relatively stable. If preferences change for each job submission, *e.g.* due to different application requirements or certain unpredictable aspects, an automated learning of users' preferences is very difficult. This is quite common in Grid environments since stakeholders can change their preferences concerning time because of approaching deadlines, personal plans, or additional budget allocation.

For these reasons we base our approach on the functional model in which an explicit expression of preferences is requested from the stakeholders. We apply the following formula to aggregate criteria:

$$F_u(CR_u) = \max_k\left\{w_k\left(c_k^* - s\left(c_k\right)\right)\right\} + \\ + \xi\sum_{k=1}^{|CR_u|} w_k\left(c_k^* - s\left(c_k\right)\right) \tag{1}$$

where w_k is a weight of criterion k, c_k is a value of criterion k, c_k^* is a nadir point [20], ξ is a constant value and $\left|CR_u\right|$ is a number of criteria. Note that depending on a criterion type an appropriate scaling function $s(x)$ should be used. We used for cost criteria the scaling function formulated as

$$s(x) = \frac{\max_x - x}{\max_x - \min_x},$$

and for gain criteria as

$$s(x) = \frac{x - \min_x}{\max_x - \min_x}.$$

As min and max parameters the minimal and maximal available criteria values have been used, respectively. For better modeling of stakeholder's preferences these values should be given by a stakeholder (or learned from stakeholder's choices).

Formula (1) is a utility function based on L_∞ (also known as *Chebyshev*) and L_1 norms (see Fig. 2). It reduces certain drawbacks of simple utility functions such as weighted mean (used *e.g.* in [20]). First, a compensation of criteria values is not as significant as for that simple utility function. Therefore, criteria values that are much worse than the best values of particular criteria (are far from the nadir point) have bigger influence on a total evaluation of schedules. In this way, additional Pareto-optimal solutions (gray points in Fig. 2) can be discovered. The second component of the formula increases the angle between isoquants of the Chebyshev norm. It allows to avoid discovering the so-called weakly efficient solutions(having the same maximal weighted distance between the nadir point and its criteria values). The actual angle between isoquants can be controlled by weights w and ξ. We used $\xi = 0.01$ for both presented models.

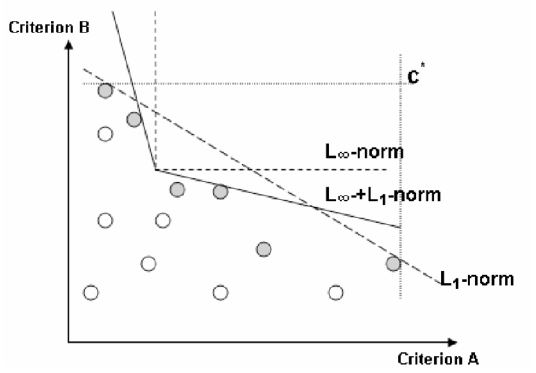


Fig. 2. Criteria aggregation function: L_∞- + L_1-norm

Please note that we also adopted the utility function based on *Chebyshev* and L_1 norms for aggregation of evaluations of all stakeholders. In this case formula (1) is slightly modified:

$$F(U) = \max_u \left\{ w_u \left(c_u^* - F_u \left(CR_u \right) \right) \right\} + \xi \sum_{u=1}^{|U|} w_u \left(c_u^* - F_u \left(CR_u \right) \right) \quad (2)$$

F_u denotes an aggregation function for particular stakeholders according to formula (1) and aggregation is performed for $|U|$ stakeholders. In this way we aggregate total evaluations of schedules by stakeholders instead of criteria values (as in formula (1)). These total evaluations are calculated using formula (1). Formula (2) is also used for both model (i) and model (ii) assuming $\xi = 0.1$. The two models are presented now in the following section.

4. MODEL (i): SCHEDULING GRID JOBS WITH UNKNOWN TIME CHARACTERISTICS

In order to effectively use the computational resources available in large cluster environments, most clusters have a local resource management system to manage and allocate resources (*e.g.* SGE, Condor, PBS/Torque, LSF). This system manages submitted users' jobs and the resource pool to complete the user requests. The system often spans the resources residing inside an organization and local domain boundaries usually set up and defined according to administrative and security policies. A significant early piece in the Grid computing vision was a uniform interface to local resource management systems providing an abstraction for remote and cross-domain job submission and control with additional powerful features such as strong security and file transfer (Globus GRAM/GridFTP, OpenDSP DRMAA/WS-Attachment). Those Grid interfaces have been widely adopted in a large number of administrative domains and allow both local administrators (resource providers) to expose the access to resources for remote users.

Virtualization of resources exposed to users through resource providers and their capabilities creates new opportunities for easier resource management of large distributed infrastructures. The increasing number and size of jobs more and more often outnumbers the available resources exposed by a resource provider in one domain. Thus we have to deal with a common problem in Grid environments: how to allocate a large number of jobs to limited resources over time. This implies a natural need for a Grid broker which enables users transparent and coordinated job submission and control to many resource providers located in different administrative domains. Additional motivations we see important for a definition of an abstract model (i) and various multicriteria analysis are presented below.

- **Facilitate the development of third-party applications without detailed descriptions of job characteristics such as job execution time**. Grid users and their applications can relay on Grid-level management capabilities provided by a Grid broker and retrieve dynamically resources they need for various computing and data operations without specifying very hard to predict performance characteristics. In particular, time characteristics are difficult to obtain and depend strongly on the performance and workload of a resource that is finally assigned to a job. Moreover, local resource management systems exposed by resource providers do not offer QoS and resource reservation mechanisms today.

Consequently, it is hard to guarantee a service level agreement and provide resources for a job during the exact period of time. To overcome this problem in the model (i) we assume that dynamic resource provider discovery and information services are available for the Grid broker.

- **Policy-based dynamic resource management.** As it was discussed in the previous sections, we assume that a set of users and providers defined by common cross-domain *resource sharing policies* creates a virtual organization (VO). Resource sharing policies can be considered as an initial agreement among all stakeholders from different domains which in practice must be obeyed. As it was mentioned, we model resource sharing rules as an appropriate set of hard constraints must be satisfied before a Grid broker submits users' jobs to resource providers. The satisfaction of hard constraints means that a user who owns a job has appropriate access rights to perform an operation (*e.g.* job submission) on a resource provider. Note that resource providers may also change dynamically resource sharing policies based on past users behavior or changing priorities. All those changes of resource provider characteristics as well as sharing policies are taken into account in the model (i).
- **Needs for user-centric and preference driven resource management**. From a resource management perspective, the degree of coordination at the resource provider and underlying resource level can vary from one VO to another. We have observed that relatively simple resource management strategies have been proposed to resource management in Grids so far. Available Grid brokers, such as Condor-G, CSF, EGEE Workload Manager, or Nimrod-G, use mostly relatively simple techniques which often do not fit to the dynamic and cross-domain nature of the Grid and considered virtual organizations today. The possibility of expressing and taking into account constraints and preferences of users as well as resource providers in a certain VO is becoming an important issue. For example, users and resource providers in the VO may have conflicting performance goals, from optimizing the performance of a single application for a goal to getting the best system throughput or minimizing waiting time.
- **Missing time characteristics of jobs and tasks to be scheduled.** In many Grid environments it is not possible to know all time characteristics of jobs a priori. Time characteristics depend strongly on the performance and workload of a resource that is finally assigned to a job. The exact times are known only after the job is finished. Sometimes the users are able to give an estimate for their jobs. However, these estimates are very often far from the actual execution times.
- **Lack of resource reservation mechanisms.** Another issue is lack or little support resource reservation and QoS mechanisms for most of the resources in Grids. Although there is a lot of work being done in this area still, huge limitations and technical constraints remain. On the other hand many Grid setups exist that support resource reservation.

4.1. Main Assumptions

As it was defined in the first section, we assume that different stakeholders (users, resource providers and a VO-level administrator) are involved in a resource management process in the VO. As many VOs are in constant flux, stakeholders can not be expected to keep up with the configuration, description and status of Grid resources, thus they must be provided with automatic and dynamic resource brokering for selecting, ranking and choosing resource providers meeting constraints and preferences they specify. Those operations logically should be performed by a Grid broker on behalf of all stakeholders. Note that on the other hand, the Grid broker represents resource providers, and has the knowledge about all available jobs, status and characteristics of resource providers and may enforce security and resource cross-domain policies defined for the specific VO in a certain period of time. Therefore, in contrast to resource management strategies available in mentioned Grid brokers, the model (i) attempts to capture the distinguishing aspects of various stakeholders with different criteria and preferences regarding job requirements and resource providers in the VO. In other words, we assume in our abstract model that the Grid broker is a decision point in which compromise jobs assignments to resource providers are made periodically according to defined hard and soft constraints in a certain moment.

After the assignment at the Grid-level each resource provider allocates jobs to Grid resources as soon as the allocation is possible. Resource providers can allocate or schedule jobs to underlying resources according to various scheduling strategies such as FIFO (First-In-First-Out), LSF (Largest-Size-First), Backfilling, etc. From the Grid broker perspective different local scheduling strategies used by resource providers are actually additional characteristics that should be taken into account during the assignment process at the Grid-level. However, we assume that the Grid broker may not (usually does not) have ownership or control over resources exposed by resource providers so it can only take into account dynamically changing resource providers characteristics.

One of the main assumptions in the model (i) is that job time characteristics are unknown. This assumption has many implications on the whole Grid resource management process but due to its practical application it has been taken into account seriously in our considerations. A list of additional characteristics concerning our model are presented below:

- the heterogeneity of resource providers which make up the VO (underlying resources have different hardware with different performance characteristics; jobs have dif-

ferent resource requirements and time characteristics are unknown),
- resources are renewable and can be used periodically,
- there is no optimal solution because various criteria are considered.

4.2. Assignment Procedure

The difficult part of the considered Grid resource management problem is to balance sharing policy enforcement with resource optimization during the process of users jobs assignment to resource providers. Essentially one can think of the Grid broker performing the following loop periodically: a) Decision: select jobs from a Grid broker queue according to resource management policies and then assign them to available resource providers (based on hard and soft constraints to maximize a global compromise solution). b) Preprocessing of input data needed for jobs to remote resource providers and start jobs. c) Stop jobs and post-processing after completion. d) Repeat. Note, that the presented assignment procedure differs from the scheduling procedure in the model (ii) presented in the next section.

4.3. Problem Definition

To allow dynamic, flexible and federated users' jobs assignment to resource providers in the model (i) we propose a novel constraint-based resource management framework for expressing and combining distributed management policies for the VO. The resource management problem is modeled as a user-centric multicriteria choice problem with aggregated criteria.

We consider the problem in which users from a finite set $U = u_1, u_2, \dots, u_{|U|}$ need to submit their jobs $J_u = j_{u1}, j_{u2}, \dots, j_{u|J|}$, to resources belonging to different resource providers from a finite set $RP = rp_1, rp_2, \dots, rp_{|RP|}$. The main goal in our model is to meet all job requirements and resource sharing polices (hard constraints – *HC*) in order to maximize a satisfaction (global compromise solution) of users, resource providers and a VO administrator viewpoints (soft constraints – *CR*) by assigning jobs to available resource providers and later to underlying resources over time. A resource provider and its exposed resources are described by a finite set of *attributes* $Q = \{q_1, q_2, \dots, q_{|Q|}\}$, also called properties, features, characteristics of resources. Resource attributes may describe for instance performance factors, QoS-based parameters, reliability as well as specific characteristics such as type of resource, specific configuration.

Before an objective function for the model (i) is presented let us introduce an important definition of a *feasible assignment* (*alternative*), denoted as *a*. A feasible assignment ($\rightarrow$) is a mapping of a job request j_{ui} to a resource r_{rpn} if and only if $\forall hc \in HC = hc_1, hc_2, \dots, hc_{|HC|}$ are met, where *u* denotes a user and *i* his job, and *rp* denotes a resource provider and *n* its resource. An example hard constraint can be defined as a relation $\propto$ between resource attribute (q_{rpnl}) and a job requirement concerning this attribute (q_{kl}^{req}), for each job *k* and its requirements Q_k^{req}, $k = 1, \dots, |J|$ concerning resource attributes Q_{ij}. This relation occurs if and only if q_{ijl} matches q_{kl}^{req}, *e.g.* is less, equal or greater than required values depending on particular attributes. Additionally, we assume that a user who owns a job has always appropriate access rights to perform an operation (e.g.. job submission) on a resource provider's resource. This hard constrain is often considered in many practical deliberations and in our model denoted as $\uplus$

Note that criteria are soft constraints, and they are tightly related to the stakeholders' preferences. In contrast to hard constraints that must be obeyed they are of secondary significance but as we demonstrate in the example they should be taken into consideration to satisfy all stakeholders as far as possible. The final solution of the considered problem is a best *compromise assignment* of jobs to resource providers and their resources. The compromise assignment is a collection of feasible assignments $\{a_1, a_2, \dots, a_m\} \in A$ that maximize a satisfaction level of all stakeholders involved in the resource management process at a certain moment. Therefore, the main goal of the Grid broker is to maximize the level of satisfaction by choosing the best compromise assignment of jobs to resource providers with respect to all considered criteria defined by stakeholders. In general, in the model (i) we consider two types of stakeholders: users and resource providers. However, in the example presented in the next subsection only user preferences are taken into account.

4.4. Model (i) – an example

In order to illustrate the advantage of multicriteria analysis in our approach the following example soft constraints (criteria) can be used by users to evaluate resource providers and their resources in the VO:
- the estimated waiting time in a queue on *rp* (resource provider): *e*
- the speed of processor unit of underlying *rp* resources: *s*
- the size of available memory of underlying *rp* resources: *m.*

In this simple example we take into account only users' preferences to find a compromise solution. As it was presented in [20] also preferences of resource providers and a VO administrator can considered during multicriteria analysis. A simple feasible assignment for a certain user's job can be evaluated according to the formula (1), where criteria are aggregated by a utility function based on *Chebyshev* and L_1 norms (presented in Section 3). In his particular case $|CR_u|$ equals 3.

Finally, the problem of finding a compromise assignment can be formulated as a multi-criteria decision support problem as follows:

$$\max \textit{ or } \min\{f_1(a), f_2(a), \dots, f_m(a)\}, \tag{3}$$

subject to:

$$\forall_{a\in A}\forall_{hc\in HC}\; a \uplus hc$$

$$\forall_{qijl\in Q_{ij}}\left(q_{ijl} \propto q_{kl}^{req}\right)$$

$$\forall_{(j_k\to r_{il})} a=\left\{j_1\to r_{il},\, j_2\to r_{il},\ldots,j_{|J|}\to r_{il}\right\}$$

where m is a number of feasible assignment,

$$i\in\{1,\ldots,|RP|\},\;\; l\in\{1,\ldots,|R_i|\},\;\; k\in\{1,\ldots,|J|\}.$$

For the aggregation of stakeholders' satisfaction (the total satisfaction) $F(U)$ again a utility function based on *Chebyshew* and L_1 norms has been used (see Formula (2)).

Let us now define example sets of users $U = \{A, B, C\}$ and resource providers $|RP| = 4$ in a VO as well as example values for attributes and weights of criteria. In this example we compare a common resource management strategy a) with multicriteria analysis b). The strategy a) is commonly used in mentioned earlier Grid brokers in which jobs are usually submitted to the most available resource providers (for example to resource providers with empty local job queues or a relatively small number of waiting jobs). The strategy b) is based on multicriteria analysis and steps formulated in the model (i) and described in the 4.2 section. Note that in this example rp_1 has resources with very fast CPUs. On rp_3 assigned jobs have to wait in a local queue much longer than on other resource providers but its resources have a lot of memory available. Note that we assume that each $rp \in RP$ meets hard constraints so all users have rights to submit their jobs to it and all characteristics describing resource providers are equal or better then jobs requirements.

Note that in this example we try to minimize criterion e (cost criterion) and maximize both s and m criteria (gain criteria). However, we have inverted the direction of optimization for e to maximize all criteria describing resource providers. Values for every single user have been calculated using weighted mean of criteria values. These values were first normalized so that they are in the range <0, 1> and are *maximized*. The final evaluation of example feasible solutions are presented as well.

Table 1. Jobs and users' preferences

Jobs	Job profiles	Criteria weights		
		w	s	m
J_A	Batch job (urgent task and very important user)	3	1	1
J_B	Batch job (regular calculation)	1	5	1
J_C	Batch job (memory intensive application)	1	1	5

Table 2. Criteria values

RPs	rp_1			rp_2			rp_3			rp_4		
Jobs	w	s	m	w	s	m	w	s	m	w	s	m
J_A	*50*	*4.5*	*1*	**0**	**2.2**	**4**	70	2.4	4	0	1.6	2
J_B	**50**	**4.5**	**1**	*0*	*2.2*	4	70	2.4	4	0	1.6	2
J_C	50	4.5	1	0	2.2	4	**70**	**2.4**	**4**	0	1.6	2

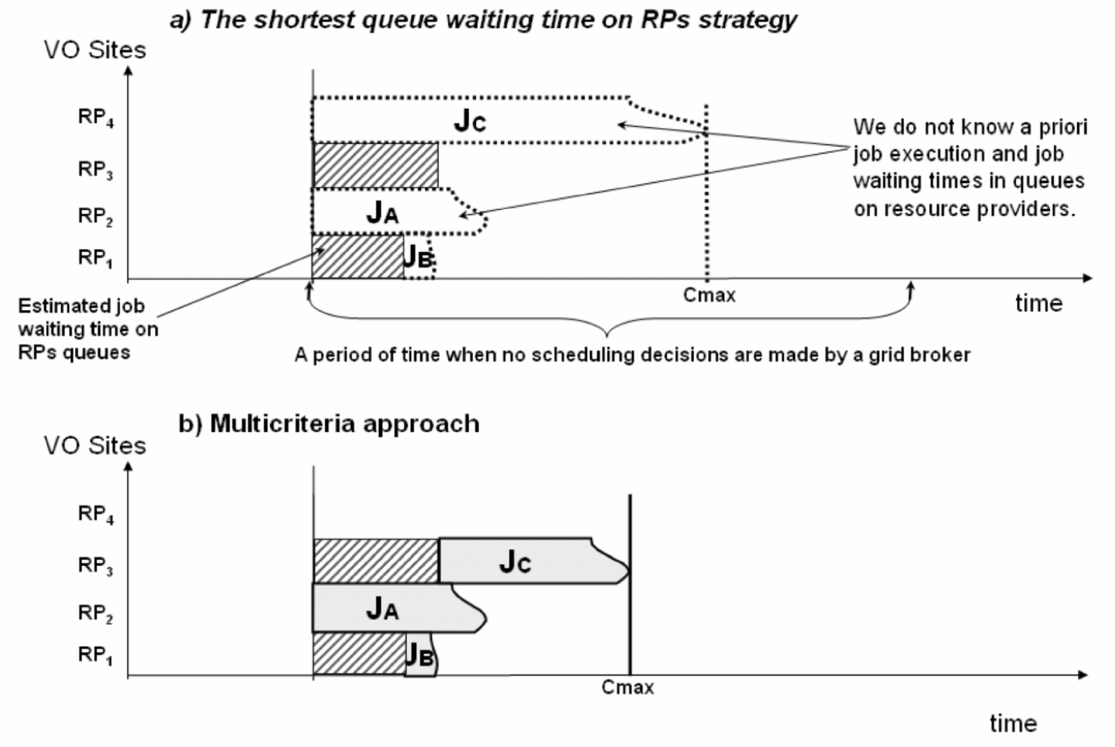

a) The shortest queue waiting time on RPs strategy
VO Sites
RP4
RP3
RP2
RP1
Jc
JA
JB
We do not know a priori job execution and job waiting times in queues on resource providers.
Cmax
time
Estimated job waiting time on RPs queues
A period of time when no scheduling decisions are made by a grid broker
b) Multicriteria approach
VO Sites
RP4
RP3
RP2
RP1
Jc
JA
JB
Cmax
time

Fig. 3. Illustration of model (i) example using: a) The shortest queue waiting time on RPs strategy and b) Multicriteria approach

First, values of criteria describing four resource providers have to be normalized so that they are in the range <0, 1> and they are maximized. A satisfaction value for each user and his job is calculated based on a scalar function consisting of Chebyshev and *L*1 norms. In the next step, all satisfaction values are aggregated and normalized based the same aggregation function. The evaluation of a few feasible assignments is presented in the table. Please observe that for example $A \rightarrow 2\ B \rightarrow 1\ C \rightarrow 4$ assignment obtained a relatively high total satisfaction (1,24), low $C_{\max}$ (160) and all jobs are executed on resource providers immediately. However, due to the fact that user *C* prefers a resource provider with a lot of memory available according to the multicriteria strategy his job *C* was allocated to rp_3 instead of rp_4. and the total satisfaction equals 1,41.

Table 3. Evaluation of feasible assignments

Solutions	Multicriteria evaluation					
	User A	User B	User C	Total satisfaction	C_{MAX}	Estimated time on RPs
*A*1*B*2*C*4	0,565	0,439	0,517	0,524	160	50
*A*1*B*2*C*3	0,565	0,439	0,854	1,166	135	60
*A*2*B*1*C*4	0,842	0,854	0,517	1,241	160	50
*A*2*B*1*C*3	0,363	0,371	0,758	**1,41**	95	60
*A*4*B*3*C*2	0,842	0,854	0,854	1, 301	135	60
*A*4*B*3*C*1	0,796	0,476	0,277	1,022	180	60

5. MODEL (II): SCHEDULING GRID JOBS IN PRESENCE OF PREDICTION INFORMATION AND RESOURCE RESERVATION MECHANISMS

Although the model (i) described in the previous section can be applied to many of existing Grid environments and improves the so-called "best-effort" approach no quality of service (QoS) guarantees are provided to stakeholders and scheduling may turn out to be far from optimal due to lack of Grid broker's control over local systems and resources. Model (ii) assumes that resource providers provide offers based on their own local policies, which means that they evaluate and select incoming requests so that handling these requests were the most profitable for them (for a description of interactions between Grid resource broker and resource providers see Section 4.2). Therefore, a Grid resource broker acts in this model on behalf of users instead of taking into account preferences of all stakeholders. Thus, in the subsequent part of the model (ii) description we focus on modeling and aggregation of preferences of users.

In particular, a few specific important motivations of introducing the model of Grid job scheduling with known job start and completion times can be identified:

- **Easier expression of user preferences.** Non-skilled users do not need to evaluate technical details of resources. This model allows users to focus on issues that are the most important and understandable for them, *e.g.* job completion time and cost. In this case they don't need to deal too much with low level technical details such as local systems queues, CPU speed *etc*.
- **Providing users with a priori information about start, wait and completion times**. Users can be informed by a Grid resource broker about job start and completion times. This is essential if users must be present during job execution, *e.g.* for interactive applications.
- **Realization of quality of service (QoS).** Using this model realization of QoS becomes possible. For instance, jobs with deadlines can be handled. Grid resource broker can schedule a job so that deadline imposed by a user is met.
- **Efficient (synchronized) co-allocation of resources.** If distributed jobs (*e.g.* MPI applications) are being scheduled in a common Grid environments (in which times are unknown a priori) it might happen that some processes are being executed while others stuck in a queue of other local resource management system. This is highly inefficient and is hard to overcome. When job start times are known then a Grid resource broker can schedule jobs so that all processes starts approximately at the same time.
- **Reliable calculation of resource utilization costs.** Users can be aware what they are charged for. If they know when their job will start and how long resources will be used it is easier to convince him/her to pay for this service.

Although most of available Grid environments do not provide knowledge about job start and completion times there are attempts in a Grid community to realize Grid scheduling based on QoS and agreements instead of "best effort" approaches. To overcome these limitations many efforts have been made on issues such as service level agreements (SLA) and negotiations [6], advanced resource reservations [34] and performance prediction [33] as described in Section 2. Aforementioned trends and needs in a Grid community show that efficient methods of Grid scheduling for such a model are gaining more and more importance. In this section we present a multicriteria approach to this problem.

5.1. Issues

The model which assumes knowledge about job start and completion times provides better quality of service and reliability, however, has certain drawbacks, too. First of all, such environments are difficult to build and sometimes may even lead to worse total efficiency of scheduling. They require additional mechanisms to support a model such as proposed in this section.

First, to guarantee specific job start time, an advance resource reservation mechanism, which is an exclusive assignment of resources for a certain user over a defined time interval, is needed. This mechanism must be provided by resource providers in a Grid environment. Nevertheless it must be introduced carefully since too big number of reservations may decrease the overall throughput due to their negative influence on start times of other jobs [34].

Another issue is that in order to obtain knowledge about job execution times, performance prediction system(s) must be available. This knowledge may be very imprecise due to heterogeneity and size of Grids so a Grid resource broker has to be aware of this possible imprecision and take it into account while scheduling. Otherwise its scheduling decisions may be wrong and lead to inefficient schedules.

If cost is one of criteria then accounting and billing systems are needed as well. In general they should allow gathering information about resource utilization and setting limits (quotas). There are many possible charging scenarios and policies based on specific business cases and computing centers policies. The use of them imposes the need of QoS guarantees.

5.2. Scheduling Procedure

Apart from differences in applied criteria and definition of a solution presented in a subsequent section, model (ii) requires a modified job scheduling process compared to model (i). Instead of making decisions based on all information about available resources it queries resource providers about their offers. They construct offers using their own local knowledge and policies. A Grid resource broker asks all candidate resource providers to make preliminary reservations and, finally, confirms reservations for those selected. A sequence of steps that must be performed during job scheduling for the model (ii) is illustrated in Fig. 1. Diagram (ii) contains elements responsible for providing resource reservation and time prediction mechanisms relevant to model (ii). As you can see, a resource broker after receiving requests from users (step 1) asks resource providers about their offers. Offers are returned in a form of lists of amounts of available resource units in various time slots (step 2). Providing an offer a resource provider agrees to initially reserve resources for a certain period. If a reservation is not confirmed by the end of this period the reservation is cancelled. This approach guarantees that resources will not be reserved by other consumers. In the next step a performance prediction system provides knowledge about estimated job start and completion times (3). The prediction system calculates estimations based on historical information that contain traces of previous job submissions. Then a resource broker filters offers according to constraints defined by users, makes a decision about a schedule of jobs (4) and returns this information to users or software acting on behalf of them (5). For those users that have accepted the choice of a resource broker reservations are confirmed with appropriate resource providers (6).

5.3. Problem Definition

The problem for this model can be defined as finding a schedule that satisfies users' preferences and requirements in the best possible way. Nevertheless, preferences and requirements are related mostly to application completion time and execution cost. The general formulation of the problem is similar to that for the model (i) but in this case a schedule means an assignment of jobs to resource providers over a specific time interval. In other words the problem is to find a compromise (maximizing users' satisfaction) schedule of jobs J of users U to resource providers RP according to the notation given in previous sections. By users' satisfaction we mean to which extent a schedule satisfies their preferences and requirements. We assume that each user may submit more than one job.

First of all the model includes a set of hard constraints that must be satisfied for a given solution in order to take this solution into consideration during evaluation and selection of the best schedule. For model (ii) hard constraints consist of two major types of constraints. The first one includes resource requirements concerning parameters such as, for instance, required operating system, CPU architecture, CPU speed, minimal amount of memory *etc.* The second type of constraints consists of time requirements, *e.g.* time slots, deadlines, days of week *etc.*

More formally, a model adopted for the described problem is as follows. Users from a finite set $U = u_1, u_2, ..., u_{|U|}$ need to run their jobs $J = j_1, j_2, ..., j_{|J|}$ on resources managed by resource providers from the set $RP = rp_1, rp_2, ..., rp_{|RP|}$. For each job resource requirements are defined. They are modeled as a set of hard constraints that must be met as explained above. They consist of amounts of resource units RU^{req} that must be reserved for a given job (*e.g.* 3 CPUs, 1 GB of disc space, *etc.*) and required resource attributes Q^{req} (*e.g.* CPU speed at least 1TFlops).

In this model we assume that a scheduler has knowledge about job start times. Thus, each resource provider must provide information about its offers in a form of lists containing available resource units in certain time slots in a given time period (t_0, t_f): $RT_i(t_0, t_f) = rt_{i1}, rt_{i2}, ..., rt_{i|RT_i|}$, where $i = 1, ..., |RP|$, $j = 1, ..., |RT_i|$, $rt_{ij} = (t_i^{\text{start}}, t_i^{\text{end}}, RU_{ij}, Q_{ij})$, where $RU_{ij} = (ru_{ij1}, ru_{ij2}, ..., ru_{ij|RU_{ij}|})$ and ru_{ijl} is an amount of the available resource unit l for resource provider i and time slot j that can be reserved for a user (*e.g.* 100 MB of free memory). Q_{ij} is a set of resource attributes as described in the previous sections (*e.g.* CPU speed, operating system, *etc.*). A single offer rt_{ij} consists of start and finish time, available resource units, and resource attributes. An offer is also called a *candidate reservation* or a *time slot* in the subsequent sections of the paper.

In addition to knowledge of deterministic (guaranteed) job start times, information about estimated job execution times is assumed to be available. To this end, a Grid scheduler can take the advantage of the list of estimated job execution times, which can be calculated by the prediction system on the basis of resource attributes provided by each resource provider for a certain reservation: et_{ijk}^{exec} where $i = 1,\dots,|RP|$, $j = 1,\dots,|RT_i|$, $k = 1,\dots,|J|$. Estimations are calculated on the basis of Q_{ij} – a specification of parameters describing a resource belonging to a given resource provider. Since job execution times are available and we assume that reserved jobs can start earlier if there is such a possibility, real job start times can also be estimated. These times, denoted as et_{ijk}^{start} where $i = 1,\dots,|RP|$, $j = 1,\dots,|RT_i|$, $k = 1,\dots,|J|$, may be significantly shorter than the guaranteed ones. They can be provided either by a prediction system if this information cannot be taken from resource providers or by resource providers themselves (in the latter a broker or prediction system should estimate possible errors of predictions delivered by resource providers).

The problem is to find the best time slot (resource providers' offer) for each job according to user's requirements. Each assignment of a job to a time slot (denoted as: $j \rightarrow rt$) is a candidate solution (also called action using a decision support terminology) and denoted as $a \in A$, where A is a set of all candidate solutions. Requirements consist of constraints that must be satisfied (hard constraints) and preferences concerning a choice of the best solution (soft constraints).

In the first step offers of resource providers must be filtered according to hard constraints defined by an user in a similar way as in model (i). This step can be performed by resource providers themselves since they retrieve from a resource broker information about job requirements in order to decide according to their local policies if there are any offers for this job. To this end two issues are checked. For each job k and offer $rt_{ij}, i = 1,\dots,|RP|, j = 1,\dots,|RT_i|$ a resource broker (or provider) checks if requirements $Q_k^{req}, k = 1,\dots,|J|$ concerning resource attributes Q_{ij} are met, *i.e.* whether $\forall_{q_{ijl} \in Q_{ij}} (q_{ijl} \propto q_{kl}^{req})$. $\propto$ denotes a relation between resource attribute (q_{ijl}) and a job requirement concerning this attribute (q_{kl}^{req}). This relation occurs if and only if q_{ijl} matches q_{kl}^{req}, *e.g.* is less, equal or greater than required values depending on particular attributes. In the second step it is checked if a sufficient amount of resource units can be reserved, *i.e.* whether $\forall_{ru_{ijm} \in RU_{ij}} (ru_{ijm} \geq ru_{km}^{req})$.

In addition to hard constraints that must be satisfied, a Grid resource broker needs criteria (soft constraints) that define how the best resources should be selected. User can specify more than one soft constraint, concerning *e.g.* time and cost. To handle such requests modeling and exploitation of multicriteria users' preferences is needed. We consider the following soft constraints (criteria) used to evaluate solutions (schedules) by every single user:

- **Guaranteed job start time**, t_s. This is a start time of reservation for this job, which is known and guaranteed by a resource provider in advance.
- **Estimated job completion time**, t_c^{est}. This is a sum of guaranteed job start time (t_s) and estimated execution time retrieved from a prediction system. Estimated execution time is a mean execution time for a given job profile based on historical information about job executions [33].
- **Estimated job start time**, t_s^{est}. This is an estimated mean job start time based on estimated execution times of previously scheduled jobs.
- **Cost of job execution**, c.
- **Variance measures of time predictions**:
  - $st\_dev(t_c^{est}/t_s^{est})$ – standard deviation of an estimated mean job execution/start time
  - $\max(t_c^{est}/t_s^{est})$ – maximal value of an estimated mean job execution/start time
  - $\mathrm{error}(t_c^{est}/t_s^{est})$ – mean error based on comparison of estimated and actual mean job execution/start times.

Please note that particular users can certainly have different preferences concerning these criteria. In particular, they can even use different sets of criteria (some of these criteria may be irrelevant) to evaluate schedules. In general, we introduced a distinction between guaranteed and estimated times since sometimes jobs can finish earlier than the end of reservation period. There are different criteria since users may have different preferences depending on whether they run interactive applications, have deadlines *etc*.

As we already mentioned several methods of preference modeling can be applied, however, here we present a procedure of resource selection using a utility function. For each pair: a job and time slot a utility function F is calculated, similarly as for model (i), using formula (1).

When multiple jobs are being scheduled, a resource broker must get resource providers' offers not only for a single job. Therefore, resource provider should specify jobs that can be run in given time slots. Thus, for each time slot the following list must be provided: $JT(rt_{ij}) = \{j_1, j_2, \dots, j_{|J|}\}$. Note that one time slot can be reserved for multiple jobs if there are enough resource units available in this slot.

A consequence of scheduling sets of jobs, which have come in a certain time interval, is a need for solutions that satisfy in the best possible way objectives of multiple users. Therefore, a total users' satisfaction must be evaluated. To this end, preferences of all users have to be aggregated into a measure that allows a resource broker to select the best schedule. A method of aggregation depends on an approach used for modeling user's preferences. Again a utility function can be applied for criteria aggregation. In our solution we perform an evaluation of the whole schedules is performed using the same metric as for criteria aggregation for every single users (see Section 2 for more details).

Based on the definitions, notations, and considerations described above the problem can be generally defined in the form of a multi-criteria problem as follows:

$$\min\left\{f_1(a), f_2(a), \ldots, f_{|CR|}(a)\right\}, \tag{4}$$

$$s.t.$$

$$\forall_{i,j} \forall_{k:(j_k \to rt_{ij}) \in a} \left(q_{ijl} \propto q_{kl}^{\mathrm{req}}\right),$$

where:

$$q_{ijl} \in Q_{ij},\ q_{kl}^{req} \in Q_k^{req},\ l = 1,\ldots,|Q|$$

$$\forall_{i,j}\left(\sum\nolimits_{k:\left(j_k \to rt_{ij}\right) \in a} ru_{km}^{req}\right) \le ru_{ijm},$$

where:

$$ru_{ijm} \in RU_{ij},\ ru_{km}^{req} \in RU_k^{req},\ m = 1, \ldots, |RU|$$

$$\forall_{\left(j_k \to rt_{ij}\right) \in a} j_k \in JT(rt_{ij})$$

$$a = \left\{j_1 \to rt_{ij}, j_2 \to rt_{ij}, \ldots, j_{|J|} \to rt_{ij}\right\},$$

$$i \in \left\{1,\ldots,|RP|\right\},\ j \in \left\{1,\ldots,|RT|\right\},\ k \in \left\{1,\ldots,|J|\right\}.$$

The set a is a candidate solution (decision action). It consists of an ordered list of time slots assigned to every single job that belongs to the set J. The first constraint ensures that all time slots meet requirements of assigned jobs in terms of resource attributes. The goal of the second constraint is to guarantee that sums of resource units that have to be allocated to assigned jobs do not exceed those offered by resource providers. As explained earlier Q_{ij} and RU_{ij} mean attributes of resources and amounts of resource units offered by resource providers respectively. Q_k^{req} and RU_k^{req} are corresponding job requirements concerning these values.

In this case the solution search space becomes large, when compared to the scheduling of single jobs. A number of possible solutions increases exponentially depending on a number of jobs. More precisely the order of magnitude of a complexity of this problem is $O(|RT|^{|J|})$. Thus, optimization methods must be used to search for the best solutions. The selection procedure must be relatively fast, therefore we used heuristic methods (local search) to provide approximate solutions.

5.4. Model (ii) – an example

In this section an example of job scheduling for the model (ii) is presented. Our multicriteria approach is compared with the well-known method – Minimum Start Time (referred as MST) strategy. In MST jobs are allocated to the resource providers that offer the earliest reservation start times. In our example there are 3 users (A, B, C) with their jobs J_A, J_B, J_C respectively (see Table 4). Each user specifies importance of criteria. For simplicity reason this is a subset of criteria defined in Section 5.3. Nevertheless, we skipped only measures related to a precision of prediction and estimated job start time, and kept the major criteria:

t_s – guaranteed start time (start time of reservation for this job)

t_c – estimated completion times (start time + estimated execution time)

c – cost of resource utilization.

All of these criteria are cost criteria (their values should me minimized). Weights of criteria used in the example are presented in Table 4. You can see that each of the users has different preferences based on a type of application he/she executes and other factors such as, for instance, imposed deadlines. Just as in the model (i) criteria weights are relative, and incomparable between users. Therefore these weights are first scaled so that their sum equals 1.

We assume that for this set of jobs values of considered parameters offered by resource providers are as those presented in Table 5. In this example we assumed, without loss of generality, that each resource providers returns one offer per job.

Table 4. Jobs and users' preferences

Jobs	Job profiles	Criteria weights		
		t_s	t_c	c
J_A	Batch job	1	1	5
J_B	Batch job with deadline	3	4	1
J_C	Interactive application	3	0	1

Table 5. Criteria values

RP	rp_1			rp_2			rp_3			rp_4		
Jobs	t_s	t_c	c	t_s	t_c	c	t_s	t_c	c	t_s	t_c	c
J_A	100	125	15	65	75	25	55	75	35	60	80	30
J_B	100	130	15	65	75	25	55	85	35	60	90	30
J_C	100	145	25	65	100	25	55	95	40	60	100	30

For these values we obtained, using the MCT policy, the schedule illustrated in Fig. 4 in Gantt chart a). We can see that although the MCT gives better makespan (C_{MAX}) the multicriteria method allows a Grid resource broker to make allocations more adequate to users' preferences (Gantt chart b)). For example, user A gets the cheapest resource, that was the most important requirement (criterion weight = 5). Estimated completion time of job B is earlier than for others providers. A job of user C starts the earliest which is very important since this is an interactive application and a user wants to start working with it as

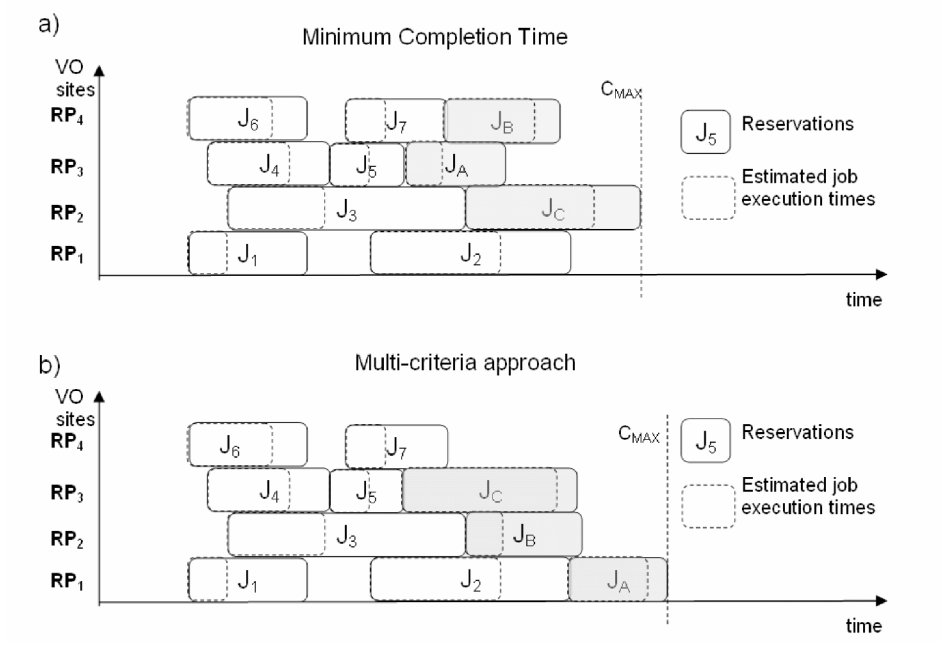


Fig. 4. Illustration of model (ii) example using: a) MCT and b) Multicriteria approach

soon as possible. In Table 5 the best solution obtained using multicriteria approach is marked using bold while the best using MCT italics.

In Table 6 detailed values of evaluation of solutions are presented. As an example six solutions are shown including those chosen as the best using the MCT policy and multicriteria approach. Notation $A1B2C3$ denotes an allocation of the job J_A to resource provider $rp1$, J_B to rp_2, and J_C to rp_3. Values for every single user have been calculated using the modified Chebyshev norm (see Section 2). These values were first normalized so that they are in the range <0, 1> and are maximized, *i.e.* the greater value the better satisfaction of a user (or total satisfaction of all users). To this end minimal and maximal available values have been used. For better modeling of user preferences this values should be given by a user (or learned from user choices). In addition to partial multicriteria evaluations of solutions based on preferences of particular users and a total evaluation, two global metrics that are commonly used in traditional scheduling were added: C_{MAX} and mean start time for a comparison purpose.

As mentioned above a total evaluation of solutions (schedules) is obtained through aggregation of evaluations of all users (also using the modified Chebyshev norm). It is easy to see that these global metrics do not reflect satisfaction of users. For instance, the solution $A1B2C3$ although the worst in terms of C_{MAX} and mean start time satisfies users' preferences in the best way. On the other hand, using MCT method, users may be often not pleased with the obtained schedule in spite of small C_{MAX} values. For instance, user A may get access to an expensive machine with a very good performance but this is not his/her goal.

Table 6. Evaluation of solutions

Solutions	Evaluation					
	User A	User B	User C	Total satisfaction	C_{MAX}	Mean start time
$A1B2C3$	0.721	0.509	0.758	1.1	125	73.333
$A2B3C4$	0.363	0.417	0.675	0.375	100	60
$A4B2C3$	0.183	0.509	0.758	0.133	95	60
$A2B4C3$	0.363	0.371	0.758	0.046	95	60
$A4B3C2$	0.183	0.417	0.590	0.013	100	60
$A3B4C2$	0.146	0.371	0.590	0.000	100	60

In general the multicriteria approach seems to be a better method for this model and presented assumptions than scheduling using global metrics or simple on-line strategies such as MCT. Use of multicriteria methods may increase a total satisfaction of stakeholders taking part in Grid resource management.

6. SUMMARY AND FUTURE WORK

In this paper we presented a multicriteria approach for Grid scheduling. In our models we assumed that each stakeholder can flexibly express, apart from hard constraints, their own preferences using multiple criteria. To model and aggregate stakeholders' preferences the functional model was adopted. More precisely we applied a combination of Czebyshev and L_1 norms that has certain favorable properties.

This general and flexible model was applied to two different environments very important in Grids. We shown how to express stakeholders' preferences in terms of hard and soft constraints. Additionally, benefits that this approach may bring for both models were illustrated by examples. In this examples we compared our results with common scheduling methods.

This study provides a solid basis for further work. In particular we would like to investigate other methods of performance modeling and criteria aggregation to make expressing preferences easier for stakeholders on one hand but on the other hand to make it more accurate. To this end, we would like to study other aggregation functions, automatic preference learning etc. Another issue to do is to compare scheduling strategies for the whole set of jobs with on-line scheduling (commonly used in current approaches). Promising preliminary results discussed in [8] indicate that also advanced sensor oriented Grid monitoring infrastructures can be used to develop more dynamic and adaptive multi-criteria strategies in the future.

We should also more deeply test various criteria and evaluation metrics to determine which are the most favorable and convenient. Another interesting experiment would be to compare our approach with totally decentralized model in which resource providers and consumers (users) negotiate and make agreements without additional entities such as Grid scheduler. Evaluation of benefits and drawbacks of use of advance reservation techniques would be also interesting as well as studying influence of resource providers policies on efficiency of Grid scheduling. At the end we can attempt to combine two models and investigate Grid scheduling for environments containing various types of resource providers.

Obviously, since we can use in addition to a simulation environment a real Grid environment based on the GRMS system, we plan to implement the described methods and after simulation experiments put them into practice. To this end, additional technical work is needed to overcome problems concerning availability of advance reservation methods, performance prediction systems, and interfaces to local resource management systems.

References

[1] http://www.coregrid.net/

[2] J. Blazewicz, K. H. Ecker, G. Schmidt and J. Węglarz, *Scheduling in computer and Manufactoring Systems*, Springer-Verlag (Second Edition) 9, 265-297 (1994).

[3] J. R. Boisseau, *UT Grid: A Comprehensive Campus Cyber-infrastructure*, hpdc, pp. 274-275, 13th IEEE International Symposium on High Performance Distributed Computing (HPDC-13 '04) 2004.

[4] Community scheduler framework (CSF) http://www.globus.org/toolkit/docs/4.0/contributions/csf.

[5] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith and S. Tuecke, *A resource management architecture for metacomputing systems*, In: D. Feitelson and L. Rudolph (eds.) Job Scheduling Strategies for Parallel Processing (Proceedings of the Fourth International JSSPP Workshop; LNCS #1459), pp. 62-68, Springer-Verlag (1998).

[6] K. Czajkowski, I. Foster, C. Kesselman, V. Sander and S. Tuecke, *SNAP: A protocol for negotiating service level agreements and coordinating resource management in distributed systems,* In: D. Feitelson and L. Rudolph (eds.), Job Scheduling Strategies for Parallel Processing (Proceedings of the Eighth International JSSPP Workshop; LNCS #2357), Springer-Verlag, pp. 153-183 (2002).

[7] P. Dinda, *Online prediction of the running time of tasks*, In: Proc. 10th IEEE Symp. on High Performance Distributed Computing (2001).

[8] P. Domagalski, K. Kurowski, A. Oleksiak, J. Nabrzyski, Z. Balaton, G. Gombás and P. Kacsuk, *Sensor Oriented Grid Monitoring Infrastructures for Adaptive Multi-Criteria Resource Management Strategies*, In: Proceedings of the 1st CoreGrid Workshop 2005, Pisa, Italy (2005).

[9] P. F. Dutot, L. Eyraud, G. Mounié and D. Trystram, *Bi-criteria algorithm for scheduling jobs on cluster platforms,* In: Symposium on Parallel Algorithm and Architectures, Barcelona, pp. 125–132 (2004).

[10] Enabling grids for e-science. http://public.eu-egee.org

[11] P. C. Fishburn, *Utility Theory for Decision Making*, Wiley, New York (1970).

[12] I. Foster and C. Kesselman, *Computational Grids*, In: I. Foster and C. Kesselman (eds.) The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, San Francisco, California, pp. 15-52. (1999).

[13] I. Foster, C. Kesselman and S. Tuecke, *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*, International Journal of High Performance Computing Applications, **15(3)**, 200-222 (2001).

[14] Global Grid Forum (GGF), http://www.ggf.org

[15] Grid Resource Allocation Agreement Protocol Working Group (GRAAP-WG), http://forge.gridforum.org/projects/graap-wg

[16] S. Greco, B. Matarazzo, R. Słowinski and A. Tsoukias, *Exploitation of a rough approximation of the outranking relation in multicriteria choice and ranking*, In: T. Stewart (ed): Multiple Criteria Decision Making. Proceedings of the Thirteenth International Conference, Cape Town (South Africa), January 1997; Springer- Verlag, Berlin (1988).

[17] E. Huedo, R. S. Montero and I. M. Llorente, *A Framework for Adaptive Execution on Grids*, Journal of Software – Practice and Experience, **34**, 631-651 (2004).

[18] E. Jacquet-Lagr'eze and Y. Siskos, *Assessing a set of additive utility functions for multicriteria decision-making, the UTA method*, European Journal of Operational Research **10**, 151-164 (1982).

[19] Job Submission Description Language (JSDL) Specification, http://forge.gridforum.org/projects/jsdl-wg

[20] K. Kurowski, J. Nabrzyski, A. Oleksiak and J. Węglarz, *Multicriteria Aspects of Grid Resource Management*, In:

Grid Resource Management, J. Nabrzyski, J. Schopf and J. Węglarz (eds.), Kluwer Academic Publishers, Boston/Dordrecht/London (2003).

[21] Kurowski K., Nabrzyski J., Oleksiak, J. Pukacki *et al.*, *Programming Grid Applications with Gridge*, In: Computational Methods for Science and Technology, by J. Nabrzyski, M. Stroinski, (eds.): Grid Applications – New Challenges for Computational Methods, OWN 2006.

[22] K. Kurowski, A. Oleksiak, J. Nabrzyski, A. Kwiecień, M. Wojtkiewicz, M. Dyczkowski, F. Guim, J. Corbalan and J. Labarta, *Multi-criteria Grid Resource Management using Performance Prediction Techniques*, In: Proceeding of the CoreGrid Integration Workshop, Pisa (2005).

[23] Oh-Kyoung Kwon, Jaegyoon Hahm, Sangwan Kim and Jongsuk Lee, *A Grid Resource Allocation System for Scientific Application: Grid Resource Allocation Services Package (GRASP)*, Mardi Gras Conference (2005).

[24] M. Litzkow and M. Livny, *Experiences with the Condor distributed batch system,* In: Proceedings of the IEEE Workshop on Experimental Distributed Systems (1990).

[25] C. Liu, L. Yang, I. Foster and D. Angulo, *Design and evaluation of a resource selection framework for Grid applications,* In: Proceedings of the Eleventh IEEE International Symposium on High-Performance Distributed Computing (HPDC-11) (2002).

[26] E. Medernach, *Workload analysis of a cluster in a grid environment*. In: D. G. Feitelson, E. Frachtenberg, L. Rudolph and U. Schwiegelshohn (eds.), Job Scheduling Strategiesfor Parallel Processing, pp. 36-61. Springer Verlag, 2005.

[27] J. Nabrzyski., J. Schopf and J. Węglarz (eds.), Grid Resource Management – State of the Art and Future Trends, Kluwer Academic Publishers (2003).

[28] B. Roy, *Multicriteria Methodology for Decision Aiding*, Kluwer, Dordrecht (1996).

[29] J. Schopf and F. Berman, *Performance prediction in production environments*. In: Proceedings of IPPS/SPDP (1998).

[30] B. A. Shirazi, A. R. Husson and K. M. Kavi, *Scheduling and Load Balancing in Parallel and Distributed Systems*, IEEE Computer Society Press (1995).

[31] R. Słowinski, S. Greco and B. Matarazzo, *Axiomatization of utility, outranking and decision – rule preference models for multiple-criteria classification problems under partial incosistency with the dominance principle.* Control and Cybernetics, **31(4)** (2002).

[32] R. Slowinski, *Rough set theory for multicriteria decision analysis*, European Journal of Operational Research **129**, 1-47 (2001).

[33] W. Smith, V. Taylor and I. Foster, P*redicting Application Run-times Using Historical Information*, In: Proceedings IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing (1988).

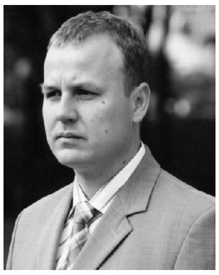
[34] W. Smith, I. Foster and V. Taylor, *Scheduling with Advanced Reservations,* Proceedings of the 2000 International Parallel and Distributed Processing Symposium. May (2000).

[35] W. Smith, V. Taylor and I. Foster, *Using Run-Time Predictions to Estimate Queue Wait Times and Improve Scheduler Performance,* In: Proceedings of the IPPS/SPDP '99 Workshop on Job Scheduling Strategies for Parallel Processing (1999).

[36] R. Wolski, N. Spring and J. Hayes, *Predicting the CPU availability of time-shared unix systems*, In:submitted to SIGMETRICS '99 (also available as UCSD Technical Report Number CS98-602) (1998).

[37] R. Wolski, N. Spring and J. Hayes, *The network weather service: A distributed resource performance forecasting service for metacomputing*, Future Generation Computer Systems, **15(5-6)**, 757-768 (1999).

Krzysztof Kurowski received the Bachelor of Science Degree from Computer Science Institute of the Poznan University of Technology in 1999. He continued his education at the Laboratory of Intelligent Decision Support System and did a second faculty at the Management and Logistic Institute. His Master Degree Diploma has been approved and honoured by the University of Paris-Dauphine in France. He joined in research and development activities conducted at Poznan Supercomputing and Networking Centre in 1999. He has taken an active interest especially in the following research fields: multi-objective optimisation, scheduling and resource management in grid environments. He has been involved in many national grid-related projects, e.g. PROGRESS, SGIGrid and Clusterix. He has taken also an active part in some EU Projects within 5 and 6 Framework Programme, e.g. Enacts, GridLab, inteliGrid and HPC-Europa. Results of his research efforts have been successfully presented at many international workshops and conferences, including IFORS, CCGrid, HPDC, Supercomputing and GGF.



Dr Jarek Nabrzyski received his M.Sc. and Ph.D. degrees in computer science from Poznań University of Technology in POLAND. Currently he is a researcher at the Poznań Supercomputing and Networking Center (PSNC), where he heads the Applications Department. His research interests over the last 10 years have focused on knowledge-based multiobjective project scheduling, and resource management for parallel and distributed computing. For the last couple of years he has been working on tools and middleware technologies for computational grids. Jarek Nabrzyski is a co-founder of the European Grid Forum and the Global Grid Forum. In 2001-2002 he was a member of the Global Grid Forum Steering Group where he was the Area Director of the Applications, Programming Models and Environments Area. In 2002-2005 he managed the European GridLab project, in which he was one of the Principal Investigators, responsible for such areas as Resource Management, Security and Mobile User Support. He is also involved in a number of 6FP projects, including *e.g.* ACGT, InteliGrid, GridCoord, BREIN, QosCosGrid, Challengers, BeInGrid, OMII-Europe. Jarek Nabrzyski is a member of several advisory boards, including projects such as Akogrimo, CoreGrid and UCoMs (USA). He is also a member of the KISTI Supercomputing Center Advisory Board (Korea).

Ariel Oleksiak received his diploma in Computer Science at the Poznań University of Technology (Laboratory of Intelligent Decision Support Systems) in 2001. Since 2002 he has been working at the Application Department of the Poznań Supercomputing and Networking Center. His research interests include mainly resource management in Grids, scheduling, multi-criteria decision support, optimization, and knowledge discovery. He has participated in many Polish as well as international research projects related to Grids such as GridLab, SGigrid, Clusterix, HPC-Europa, and BREIN. He presented his results in many international conferences, workshops, and scientific journals.

Jan Weglarz, Academician, Professor (Ph.D. 1974, Dr. Habil. 1977), in years 1978-83 Associate Professor and then Professor in the Institute of Computing Science, Poznań University of Technology, member of the Polish Academy of Sciences (PAS), Director of the Institute of Computing Science, Poznań University of Technology and its predecessors since 1987, Director of Poznań Supercomputing and Networking Center, President and Scientific Secretary of the Poznań Branch of the PAS, v-ce President of the Committee for Computer Science of the PAS, member of the State Committee for Scientific Research, Principal Editor of the Foundations of Computing and Decision Sciences, member of several editorial boards, among others Internat. Trans. Opnl. Res. and European J. Opnl. Res. Representative of Poland in the Board of Representatives of IFORS and in EURO Council (President of EURO in years 1997-98). Member of several professional and scientific societies, among others the American Mathematical Society and the Operations Research Society of America. Author and co-author of 11 monographs, 3 textbooks (3 editions each) and over 200 papers in major professional journals and conference proceedings. Frequent visitor in major research centers in Europe and in the USA Co-laureate of the State Award (1988) and the EURO Gold Medal (1991), laureate of the Foundation for Polish Science Award (2000).