International Journal of Approximate Reasoning 53 (2012) 228–247

Contents lists available at SciVerse ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

On providing quality of service in grid computing through multi-objective swarm-based knowledge acquisition in fuzzy schedulers

R.P. Prado [a], F. Hoffmann [b], S. García-Galán [a,*], J.E. Muñoz Expósito [a], T. Bertram [b]

[a] Telecommunication Engineering Department, University of Jaén, Alfonso X el Sabio, 28 Linares, Jaén, Spain
[b] Control System Engineering, University Dortmund, D-44221 Dortmund, Germany

ARTICLE INFO

Article history:
Received 2 February 2011
Received in revised form 18 October 2011
Accepted 19 October 2011
Available online 6 November 2011

Keywords:
Soft computing
Grid scheduling
Fuzzy rule-based systems
Automatic learning
Multi-objective evolutionary algorithms

ABSTRACT

Nowadays, grid computing is increasingly showing a service-oriented tendency and as a result, providing quality of service (QoS) has raised as a relevant issue in such highly dynamic and non-dedicated systems. In this sense, the role of scheduling strategies is critical and new proposals able to deal with the inherent uncertainty of the grid state are needed in a way that QoS can be offered. Fuzzy rule-based schedulers are emerging scheduling schemas in grid computing based on the efficient management of grid resources imprecise state and expert knowledge application to achieve an efficient workload distribution. Given the diverse and usually conflicting nature of the scheduling optimization objectives in grids considering both users and administrators requirements, these strategies can benefit from multi-objective strategies in their knowledge acquisition process greatly. This work suggests the QoS provision in the grid scheduling level with fuzzy rule-based schedulers through multi-objective knowledge acquisition considering multiple optimization criteria. With this aim, a novel learning strategy for the evolution of fuzzy rules based on swarm intelligence, Knowledge Acquisition with a Swarm Intelligence Approach (KASIA) is adapted to the multi-objective evolution of an expert grid meta-scheduler founded on Pareto general optimization theory and its performance with respect to a well-known genetic strategy is analyzed. In addition, the fuzzy scheduler with multi-objective learning results are compared to those of classical scheduling strategies in grid computing.

1. Introduction

Grid computing is a hardware and software platform that provides high-end computational capabilities by the cooperation of geographically distributed resources interconnected through high-speed networks [1]. It essentially consists of a shared environment supported by a standard-based service infrastructure that enables the coordination of administrative domains or communities computing resources [2]. Resources may differ in terms of scope, competence and structure and they are subject to the local access and sharing policies of their associated administrative organization. Moreover, as corresponds to non-dedicated systems, the capabilities of the available resources and reservation behaviour may change with time. These facts together with the active arrival of heterogeneous jobs (i.e., jobs are diverse in computational needs, they can be classified into computing or data intensive; some of the jobs can be full applications with diverse specifications and others can be just atomic tasks [3]) and failures occurrence, make the grid a "*fully dynamic environment with uncertainties*" [4]. Thus, unlike other traditional distributed systems, one of the main problems is related to the efficient coordination of resources or scheduling in an inherent dynamic environment, which is known to be a NP-complete problem [5].

The scheduling problem in grid computing is multi-objective in its general formulation [3,4,6]. A grid performance can be rated in terms of many different criteria which must be simultaneously considered in the scheduling process to

* Corresponding author. Tel.: +34 953648556.
E-mail addresses: rperez@ujaen.es (R.P. Prado), frank.hoffmann@tu-dortmund.de (F. Hoffmann), sgalan@ujaen.es (S. García-Galán), jemunoz@ujaen.es (J.E. Muñoz Expósito), torsten.bertram@tu-dortmund.de (T. Bertram).

0888-613X/$ - see front matter
doi:10.1016/j.ijar.2011.10.005

achieve the objectives of the grid. Essentially, the objectives of a scheduling system in grid computing are to manage and offer resources in a way that users demands are satisfied, what may involve the consideration of different levels and sorts of quality of service (QoS) [7]. In fact, grid computing is increasingly showing a service-oriented tendency [8] and as a result, providing QoS has raised as a relevant issue in such highly dynamic and non-dedicated systems. The definition of QoS depends on the different applications and it may concern metrics regarding hardware capacity, available software and many other resources specifications. Further, as studied in [2], QoS is usually a constraint that must be imposed on the scheduling process. The consideration of QoS has effect on the resource selection step and then affects all the final optimization process. Therefore, resources assignment in the scheduling process is highly influenced by the consideration of QoS and thus, it also affects the optimization of the final scheduling objectives. Furthermore, as grid systems become more complex and used, the specification of QoS also becomes tougher and more demanding and requires the consideration of multiple objectives in the scheduling. However, given the diverse nature and the large number of optimization criteria, the simultaneous consideration of multiple optimization criteria in the scheduling process is a major challenge. In order to optimize the behavior of a scheduler regarding multiple QoS objectives, one of the existing solutions is based on the definition of multi-objective indexes that add different optimization objectives to obtain scalars that show the quality of the scheduling. One of the main problems in the specification of multi-objectives indexes is given by the normalization of the different optimization criteria. The different optimization objectives are often measured in different units and scales what makes the specification of multi-objective indexes complex. Further, these objectives can be conflicting in such a way that improving one objective may be to the detriment of other objectives. Nevertheless, schedulers should be able to work simultaneously with diverse optimization objectives. In addition, it must be noted that grid performance fluctuation makes it difficult to evaluate the scheduling strategy performance and it is claimed to be one of the main differences with conventional distributed systems [2].

Besides, the dynamism in the grid environment makes the adaptation of schedulers to changing conditions necessary. Adaptive scheduling strategies are those in which the grid past, current and future states are considered in every step of the decision making process in a way that system performance is not deteriorated and thus, they are able to provide certain levels of QoS [3,4]. Actually, it is stated that any scheduling strategy aiming to provide certain levels of QoS must consider the grid state. In this sense, the role of fuzzy rule-based scheduling strategies have recently attracted the attention for the scheduling problem in massive parallel machines and grids [9–11]. Fuzzy rule-based schedulers are knowledge-based systems [12] derived from Fuzzy Rule-Based Systems (FRBSs) that incorporate grid expert knowledge and deal with the grid state information uncertainty through the application of fuzzy logic [13–15] to provide efficient solutions. Their main strength resides in their ability to adapt to variations in the grid conditions through the flexible featuring of the system state and associated inference of the most suitable response. However, the successful performance of fuzzy rule-based schedulers highly depends on the quality of their knowledge bases and thus, with the learning strategy. In spite of the existence of some approaches facing the adaptation of fuzzy rule-based schedulers to the scheduling process [9–11], there exists a lack in the consideration of multiple optimization parameters simultaneously in their knowledge acquisition processes. However, a multi-objective learning for fuzzy rule-based schedulers is necessary to offer complex QoS requirements.

This work suggests the QoS provision in grid computing with fuzzy rule-based schedulers. With this aim, a multi-objective learning strategy for fuzzy rules is incorporated to the fuzzy schedulers. Specifically, the learning strategy is based on the adaptation of KASIA (Knowledge Acquisition with a Swarm Intelligence Approach) to the multi-objective evolution of fuzzy rules for grid schedulers and it is founded on the Pareto general optimization theory [16,17]. The performance of the proposed learning strategy is compared to the well-known genetic learning strategy, Pittsburgh approach and results are analyzed in terms of convergence behaviour and accuracy considering non-parametric statistical tests. It is shown that incorporating multi-objective techniques to the knowledge acquisition of expert fuzzy schedulers allows obtaining a set of high quality solutions, including trade-off solutions, among contradictory optimization criteria simultaneously. Moreover, the performance of fuzzy schedulers (incorporating the acquired knowledge with the proposed multi-objective learning) is compared to that of widely used scheduling strategies in grid computing, *EASY-Backfilling* (*EASY-BF*) [18], *First Come First Served* (*FCFS*) [19], *Earliest Suitable Gap* (*ESG*), *ESG + Local Search*(*ESG + LS*) and *ESG + LS periodical* [20–22], considering diverse grid performance criteria through the whole scheduling process. Therefore, this work represents a new effort towards the improvement of fuzzy rule-based schedulers and QoS provisioning in grid computing.

The rest of the paper is organized as follows. In Section 2 an overview of optimization criteria and multi-objective scheduling in grid computing is provided. Section 3 introduces the general formulation of the scheduling problem and presents the fuzzy scheduler organization within a grid environment. The proposed knowledge acquisition process for fuzzy meta-schedulers based on KASIA approach for multi-objective learning in grid computing is described in Section 4. In Section 5 the performance of the learning strategy is evaluated through simulations and results of the expert fuzzy scheduler are compared to those of classical grid scheduling techniques. Finally, Section 6 concludes the paper.

2. Background

Scheduling in grid computing is described as a multi-objective problem in its general formulation [3,4,6]. An efficient scheduling strategy must satisfy both users and administrators demands what generally involves the simultaneous optimization of several scheduling optimization objectives. One challenge in the simultaneous optimization is related to the conflicting or contradictory nature of the diverse optimization objectives. On the one hand, it is said that an objective conflicts with another objective, when optimizing one of these objectives at least causes the detriment of the other one [3,4]. On the other hand, an optimization objective is considered contradictory respect to another objective when its optimization does not only involves the deterioration of the other objective but the optimization of this one in a opposite sense (i.e., minimizing the first objective leads to maximizing the second one or vice versa). Since the optimization of the grid performance is usually associated to providing a balance between an acceptable QoS for users and an efficient harnessing of the distributed system or throughput, which are conflicting objectives, finding trade-off solutions often arises as the goal of the optimization process [3]. Another challenge is associated to the measurement of the different optimization objectives. The optimization objectives are generally measured in various scales and units and thus, a direct combination of these objectives to define performance indexes is not a feasible option in most of the multi-objective scheduling processes. Hence, a relevant problem is how to combine and compare different and typically opposed scheduling optimization criteria to obtain efficient schedules and how to incorporate these strategies to the scheduling system. Note that in this work, the terms criteria and objective are used interchangeably, where optimization objectives/criteria are used to express the goal of the optimization and performance objectives/criteria are used to characterize the grid system performance [4].

Different approaches can be found in the multi-objective optimization theory that consider the integration of multiple optimization criteria. These strategies are generally classified into hierarchical and simultaneous ones [3]. On the one hand, hierarchical approaches suggest the consideration of priority levels among the diverse optimization criteria and these priorities are established based on the final purpose of the grid, e.g., in high-performance grids, it can be commonly beneficial to further improve *makespan* compared to *response time* and an analog reasoning can be followed in case users demands are to be prioritized [3]. Therefore, in the hierarchical approach, the scheduling optimization objectives are ranked and an optimization process is conducted in such a way that higher priority criteria cannot be deteriorated when improving less significant ones. This approach is mainly suitable for those cases in which the optimization objectives are quantified in different units and scales and their combination in a single performance index is not feasible, e.g., aggregation of *makespan* and *resource usage*. Some instances of the application of the hierarchical approach in jobs scheduling can be found in [23,24]. The main drawback of the hierarchical approach is given by the specification of priorities among objectives which is not feasible in many situations. On the other hand, in the simultaneous approach, several optimization objectives are to be optimized at the same time. A critical aspect of this approach is given by the high computational costs associated to the optimization of multiple conflicting or contradictory optimization objectives. To deal with the problem, the Pareto optimization theory [17] is studied as an efficient solution. Within the Pareto optimization theory several strategies can be mentioned. However, two main strategies can be found for grid scheduling: the weighted sum approach and the general approach [3,4].

In the weighted sum approach the diverse optimization objectives are aggregated in a single multi-objective index. This index indicates the quality of the schedule in such a way that the problem can be addressed as single-objective. The weighted sum approach presents a major problem; the aggregation of several optimization objectives into a single index requires the specification of weighted factors to compensate the different scales and units. This increase the complexity of the search process since it involves adding new variables to the problem. This way, in many practical situations it is generally necessary to set these parameters with prior tuning processes and users or applications optimization criteria predefined priorities. Xhafa et al. suggest the selection of *makespan* and *flowtime* as optimization criteria for the scheduling problems in grid computing [3,4,25]. Specifically, the aggregation of these two indexes into a single index is considered. However, in spite of the fact that both *makespan* and *flowtime* are measured in the same units, they are associated to incomparable ranges and thus, an alternative formulation for these optimization criteria such as *normalized flowtime* must be used. Also, suitable weight factors to compensate the diverse optimization criteria must be found and a prior tuning process is suggested. This way, authors state that the combination of these optimization criteria into a single index allows the application of single-objective metaheuristics. Also, Carretero and Xhafa [26] present a scheduling implementation based on genetic algorithms for independent job scheduling in grid computing and both hierarchical and weighted approaches are considered for *makespan* and *flowtime*. A similar approach can be found in [6] where a modification of AFSA (Artificial Fish Swarm Algorithm) is proposed and different optimization criteria are optimized on the basis of predefined priorities. On the other hand, Li and Li [27] apply the concept of utility function to combine and optimize three optimization criteria in computational grids, i.e., payment, deadline and reliability, and these multi-dimensional requirements are combined as an overall utility function by the weighted sum of the various QoS utility functions. Also, Izakian et al. [28] suggest the composition of conflicting temporal optimization criteria into a single weighted performance index for scheduling in grid computing. However, in the weighted simultaneous approach, it is discussed whether it is always feasible to determine efficient values for the weight factors given their high dependence with the specific grid state and dynamism of the grid. Furthermore, it is argued whether this approach can be considered as multi-objective in a strict sense.

The second simultaneous approach is the general approach. The general approach suggests to compute the Pareto optimal solutions [29]. In contrast to the weighted-based simultaneous approach, the search for Pareto optimal solutions does not need prior tuning processes to find suitable weight factors to combine different optimization objectives measured in diverse scales and units. Further, no priorities have to be established among the diverse optimization objectives. Hence, this approach provides a general strategy that cannot only be suitable but necessary when objectives cannot be prioritized or appropriated scaling factors cannot be found. The concept of Pareto optimality has been used for multi-objective optimization in several scheduling strategies. In [30] a scheduling strategy for computational grids based on resources providers bids is introduced by Sample et al. In this strategy a bid or index that considers the needed service, expected start time for the service and information related to size and complexity of the service input parameters is defined and the scheduling strategy decision is taken founded on the Pareto optimality of the best schedules founded on this bid. Specifically, a bid is said to be Pareto optimal in case it outperforms it in terms of time, cost and performance certainty the rest of bids. Also, in [31] some preliminaries studies about the simultaneous combination of bi-criteria for scheduling in grid computing and evolutionary multi-objective (EMO) using the Pareto dominance are presented. In addition, Benedict and Vasudevan [32] study the problem of scheduling scientific workflows in grids by a niched Pareto-based strategy considering the workflow completion subject to deadlines. Talukder et al. [33] introduce Multi-objective Differential Evolution (MODE) with the aim of generating trade-off in schedules with respect to users QoS demands in terms of time and cost and Pareto fronts are obtained for different workflows. Moreover, regarding adaptive scheduling, Franke et al. [9,34] presented a fuzzy schema for scheduling independent jobs on parallel machines which are increasingly part of computational grids. They formulate an objective function that prioritizes different users average weighed response time and results are analyzed regarding the Pareto front of all possible scheduling decisions for the workload. In this work, the simultaneous optimization of diverse objectives though the use of Pareto general theory is suggested for the provision of QoS in grid scheduling. Specifically, the aim is to find a set of non-dominated solutions or rule bases (RBs) that can be used by the fuzzy rule-based scheduler depending on the conditions of the grid. Non-dominated solutions are obtained through the general theory of Pareto to avoid the problems associated with simultaneous weighted and hierarchical approaches in the multi-objective optimization of schedulers as studied in this section.

3. Problem formulation and fuzzy rule-based scheduler specification

The general formulation of the scheduling problem in a hierarchical grid can be summarized as follows [35]. The grid system, GS, is made up of a set of G geographically distributed sites or resources domains RD_j, $GS = \{RD_1, RD_2, \ldots, RD_G\}$ that aggregate H_j heterogeneous computational resources, $RD_j = \left\{r_{j,1}, r_{j,2}, \ldots, r_{j,H_j}\right\}$ and share capabilities in order to satisfy users and applications demands on the basis of own local access and availability policies that may change with time. In a grid environment jobs, $J = \{J_1, J_2, \ldots, J_L\}$, dynamically arrive and specify the required properties for the target resource in order to satisfy compatibility such as the time limit for the execution (given by the queue or user), the required machine architecture, the demanded software licenses, the operating system, the network type or the file system. Thus, a job describes a user's application that can demand a single (i.e., sequential) or more CPUs (i.e, parallel). Several machines within the same cluster or site can be co-allocated to execute a given parallel job but machines belonging to different clusters cannot be co-allocated to process the same parallel job [36].

Regarding scheduling of users jobs, two scheduling levels can be differentiated in a hierarchical grid. On the one hand, Local Resource Managers (LRMs) are associated to resources domains and they are responsible for the allocation of workload among machines within their sites. It must be highlighted that workload corresponds both to grid and local users. Some examples of LRM are OpenPBS [37] and Condor [38]. On the other hand, the cooperation of LRM is addressed by a second-level scheduling system or meta-scheduler. Thus, the goal of a meta-scheduler is to manage the grid resources efficiently through the local scheduling systems coordination. Also, the role of Grid Information Systems (GISs) must be pointed out in the scheduling structure. Considering the heterogeneous and dynamic nature of grids, the information related to resources state must be considered to achieve a proper schedule [2]. In this sense, GIS are in charge of providing resources domains state (i.e., gathering and predicting site dynamic information, such as CPU capabilities and memory size, software resources, network bandwidth and domains background load) to meta-schedulers and they are supported by associated LRM reports. To this end, LRM use tools such as Network Weather Service [39], Hawkeye [38] and Ganglia [40]. Furthermore, an example of GIS can be found in The Globus Monitoring and Discovery System (MDS) [41]. Hence, within this grid system structure, our work is focused on the improvement of meta-schedulers.

Scheduling in grid computing is a NP-complete problem in its general formulation and many strategy have been suggested to obtain an efficient planning [5,21]. Scheduling strategies can be classified into queue-based and schedule-based strategies. On the one hand, queue-based strategies such as *EASY-Backfilling* (*EASY-BF*) or *Earliest Deadline First* (*EDF*), are characterized by their simplicity and short algorithms runtimes [22] and can be found of today's production systems (e.g., Condor [42] or Grid Service Broker [43]). A relevant drawback of queue-based strategies is related to the provision of QoS. These strategies cannot offer many QoS guarantees since they do not consider the current conditions of the grid to compute the schedule. On the other hand, scheduled-based methods such as *EGS* (*Earliest Gap*), are founded on up-to-date grid state information [44]. However, the state of the grid system, in contrast to classical distributed systems, is uncertain due to the high dynamism and thus, the grid available information is mostly imprecise and can deteriorate the efficiency of the schedule. This way, recent works are focused on the design of adaptive scheduling strategies [3,4] able to work in a system subject to uncertainty. In this sense, it is important to underline the consideration of FRBSs.

Fuzzy rule-based meta-schedulers are expert systems that provide scheduling decisions on the basis of the reasoning applied over the sites state information and their own knowledge of the environment. The state of the grid sites is given by a set of features, so-called grid input features, that characterize their current conditions. The general structure of a fuzzy rule-based scheduler within the grid environment is shown in Fig. 1. Essentially, the aim of the expert meta-scheduler is to provide an indicator of suitability for each available site to be used in the current planning. To do this, in every scheduling step, the state of each site is obtained considering a number of grid input features and this state in numerical format is transformed into a linguistic format that considers the fuzzy uncertainty of this information. Next, the knowledge of the

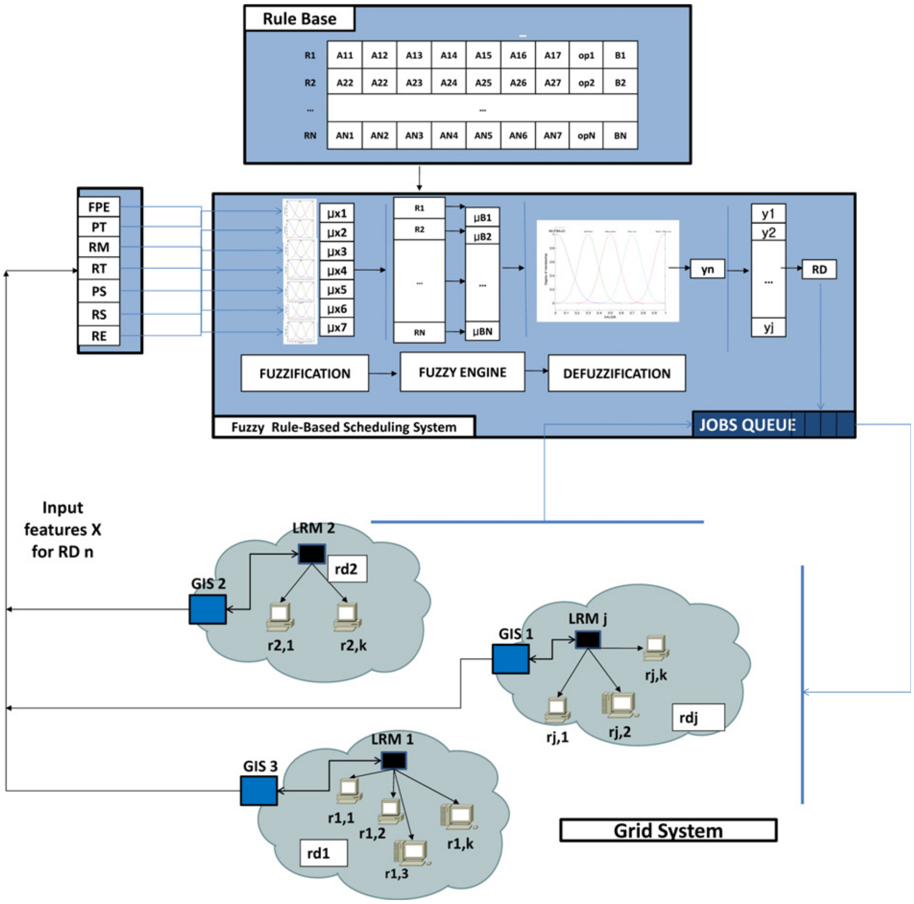


Fig. 1. Fuzzy rule-based meta-scheduling system structure.

expert system given in the form of fuzzy rules is applied to this fuzzy to obtain the indicator of suitability for the considered site. Finally, once the meta-scheduler has all the indexes for all the sites, the site with a higher rate is selected for the current schedule.

Specifically, the basic operation of a fuzzy rule-based meta-scheduler can be summarized as the combined operation of three main systems for every site, namely, *fuzzification*, *inference* and *defuzzification* systems:

Fuzzification system. First, in the fuzzification stage, the expert scheduler selects the finite set of features to describe the grid sites state and makes a fuzzy characterization of these features crisp values in a way that the inherent uncertainty of the information provided by GIS systems is considered. This selection of sites features must be done in a way that a compromise is reached among the accuracy in the state description and the simplicity of the expert system associated knowledge, i.e., knowledge base (KB). The scheduler knowledge is given in the form of associations or "IF-THEN" rules relating the grid input features and the expert system output. Specifically, in this work, a Mamdani encoding [45,46] is considered for the rules where a rule R_i can be formulated as

$$R_i = IF \quad x_1 \, is \, A_{i,1} \;\; and/or \ldots x_n \;\; is \;\; A_{i,n} \;\; THEN \;\; y \; is \;\; B_i \tag{1}$$

where $(x_1, \ldots, x_n)$ represents the n considered grid input features to describe the grid state and $A_{i,m}$ and B_i denote the associated fuzzy sets for the grid input feature x_n and output for rule i, respectively. Hence, a large number of input features can significantly increase the search space and enlarge the learning processes. In this work, grid features are selected following previous works in the design of fuzzy rule-based meta-schedulers [11,47]. Specifically, the following grid input features are considered which concern both actual resources state and performance evolution,

- Number of free processing elements (FPE): Number of free processing elements within a participating resource domain, RD_i.
- Previous tardiness (PT): Sum of tardiness [48] of all finished jobs in resource domain RD_i.
- Resource makespan (RM): Current makespan [4] or finalization time of the last considered job in the RD_i.
- Resource tardiness (RT): Current tardiness of jobs assigned to the RD_i.

- Previous score in deadline evaluation (PS): Previous deadline score of already finished jobs in the RD_i.
- Resource score or number of delayed jobs (RS): Number of non delayed jobs so far in the RD_i.
- Resources in execution (RE): Number of resources executing jobs within the RD_i currently.

Hence, the fuzzification interface addresses the task of converting the crisp input values characterizing the site state into fuzzy values to be used in the inference phase. Specifically, in the fuzzification interface, the aim is to retrieve the current values for the grid input features and to obtain the degree to which these inputs belong to each rule for every site.

Inference system. Next, in the inference stage, for every site the scheduler applies its knowledge, i.e., RB, to obtain a fuzzy set showing the suitability of the site to be selected in the next schedule. Once the inputs are fuzzified, the degree of membership to which every input is satisfied for every rule, is known. In the inference stage, inference operators are applied to determine a fuzzy set that indicates the overall result for every rule for every site.

Defuzzification system. Finally, the fuzzy output for every site is translated into a crisp index in the defuzzification stage in a way that a quantifiable value can be associated to the site suitability to be selected. As described above, decisions are founded on the analysis of every rule of the expert system knowledge base as individual. However, in order to obtain a final crisp decision, rules contribution must be first joined for every site. The combination of output fuzzy sets of every rule into an overall output fuzzy set, is done in the aggregation process. Finally, a defuzzification method is applied in order to obtain a crisp output index for every site from the resulting output fuzzy set obtained in the aggregation process.

These processes, *fuzzification*, *inference* and *defuzzification* stages are repeated for every participating site in every schedule. This way, the expert scheduler founds its decision on the acquisition of several (i.e., number of sites) resource domain selectors or indexes in every schedule. The higher the site index the more suitability of the site to be selected for the current schedule. It must be noted here that the execution cost of the proposed meta-scheduler corresponds to that of fuzzy logic systems where a fuzzification stage is required for every site in the grid. The implementation of these systems is in fact a very easy and quick process and what it is more important to allow scalability, their software cost are low [45].

As studied in this section, the fuzzy rule-based meta-schedulers reasoning strategy is mainly driven by the application of the system knowledge to the system state and this way the quality of this knowledge is decisive for the their performance where quality can be rated in terms of diverse optimization criteria. Therefore, this knowledge must be robust to withstand changes in conditions on the grid due to dynamism. This is why the expert system must employ knowledge gained from an extended learning of the environment in which it is located. Thus, in this work real traces provided by a current operating environment and including a wide period of operation are used in a way that the scheduler can be learnt considering the dynamic behavior of the network due to various causes such as the absence or addition of domains or resources, changes in access policies or failures. On the other hand, a significant modification of environmental conditions with respect to those the system had been trained with (due to a high dynamism) can lead to a loss of quality of the expert system knowledge. This is the reason why the proposed scheduler must be provided with the ability to adapt to this high dynamism by considering learning processes that are able to improve the quality of its scheduling. The incorporation of an efficient learning system for the meta-scheduler prevents the loss of quality of the scheduling strategy at the advent of major changes in the conditions of the grid system. Hence, the knowledge acquisition process is critical for the efficient operation of fuzzy rule-based scheduling systems. In the next section a multi-objective learning approach based on swarm intelligence is presented.

4. Multi-objective swarm-based knowledge acquisition strategy

In this work, expert meta-schedulers RBs are suggested to be acquired through the adaptation of KASIA strategy [11] to multi-objective learning. KASIA is a swarm intelligence-based strategy for the acquisition of fuzzy RBs inspired by the stochastic evolutionary algorithm Particle Swarm Optimization (PSO). Although the original aim of PSO was to simulate the choreography of birds within a flock graphically, it has derived in an optimizer which has proved effective in complex multidimensional problems in a wide range of research areas including renewable energies [49] and electromagnetics [50]. Further, PSO has been adapted to knowledge acquisition in FRBSs [11]. The major advantages of KASIA over other classical learning strategies in FRBSs such as Pittsburgh [51] and Michigan [52] approaches are related to its accuracy and convergence velocity. Furthermore, it is to be mentioned its simple implementation and reduced number of fixing parameters. In KASIA, each individual is known as a "particle" P_i of a swarm which moves within a multidimensional search space and represents a whole fuzzy RB,

$$P_i = \begin{bmatrix} a^i_{1,1} & a^i_{1,2} & \dots & a^i_{1,n} & b^i_1 & c^i_1 \\ a^i_{2,1} & a^i_{2,2} & \dots & a^i_{2,n} & b^i_2 & c^i_2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a^i_{m,1} & a^i_{m,2} & \dots & a^i_{m,n} & b^i_m & c^i_m \end{bmatrix}, \quad i = 1, 2, \dots, NP \tag{2}$$

where every row denotes the encoding of a Mamdani type fuzzy rule [45], n, m describe the number of input variables and rules, respectively and NP represents the number of particles in the swarm. Thus, antecedents $a^i_{j,k}$ are represented as an integer in the interval

Algorithm 1 Multi-objective knowledge acquisition with a swarm intelligence approach, KASIA.

Initialization

1. Swarm: the size of the swarm (Num_particles) and the number of rules of each individual (m), number of iterations of the algorithm (Num_iter), initial value for inertia weight (ω_0), weight factors d_1 and d_2.
2. Random initialization of RB-swarm position: rules initialization
   for every particle/RB i and every rule j,
   (a) Check no zero antecedents are considered to keep rules coherence:
   while($\sum_{k=1}^{n} a_{j,k}^{i} == 0$)
   Random setting of $a_{j,k}^{i}$ *for random k* (rule initialization if antecedents are zero)
   end
   (b) *if* $\mid a_{j,k}^{i} \mid > NF_{in}$ *then Eq.* (11)
   (c) *if* $\mid b_{j,k}^{i} \mid > NF_{out}$ *then Eq.* (12)
   (d) *if* $c_j^i \notin \{1, 2\}$ *then Eq.* (13)
   end
3. Random initialization of velocity of particles in the swarm.
4. Check velocity satisfy maximum and minimum values corresponding to V_{max} and V_{min} after initialization.
5. Initialize best local non-dominated solution ($P_{ND}^{\#}$), where this solution corresponds to:

$$f_i(P_{ND}^{\#}(t)) \leq f_i(P_{T_i}) \forall i = 1, \ldots, q$$
$$\textit{and } \exists i \in \{1, \ldots, q\} \textit{ with } f_i(P_{ND}^{\#}(t)) < f_i(P_{T_i}))$$

6. Initialize best global non-dominated solution (P_{ND}^{*}), where this solution corresponds to:

$$f_i(P_{ND}^{*}(t)) \leq f_i(P_T)\ \forall i = 1, \ldots, q$$
$$\textit{and } \exists i \in \{1, \ldots, q\} \textit{ with } f_i(P_{ND}^{*}(t)) < f_i(P_T))$$

Swarm search for best locations loop

For (Num_iter),
For (Num_particles)
1. Update position of a particle in the swarm. Eq. (9).
2. Check if constraints are satisfied in RB-swarm position.
   (a) Check no zero antecedents are considered to keep rules coherence:
   while $\sum_{k=1}^{n} a_{j,k}^{i} == 0$
   Random setting of $a_{j,k}^{i}$ *for random k* (rule initialization if antecedents are zero)
   end
   Check antecedents, consequents and connectives restrictions
   (b) *if* $\mid a_{j,k}^{i} \mid$ NF_{in} *then Eq.* (11)
   (c) *if*
   $midb_{j,k}^{i} \mid > NF_{out}$ *then Eq.* (12)
   (d) *if* $c_j^i \notin \{1, 2\}$ *then Eq.* (13)
3. Evaluate the quality found by the particle.
   Particles ++, repeat the process above for the following particle in the swarm
end
Update best global non-dominated solution (P_{ND}^{*}).
For (Num_particles)
1. Update best local non-dominated solution ($P_{ND}^{\#}$).

$$f_i(P_{ND}^{\#}(t)) \leq f_i(P_{T_i}) \forall i = 1, \ldots, q$$
$$\textit{and}$$
$$\textit{existsi} \in \{1, \ldots, q\} \textit{ with } f_i(P_{ND}^{\#}(t)) < f_i(P_{T_i}))$$

2. Update velocity. Eq. (18).
3. Check velocity satisfy maximum and minimum values corresponding to V_{max} and V_{min} after initialization.
Particles ++
end, repeat the process above for the following particle in the swarm
iter++
end
Return Solution: best global non-dominated solutions (P_{ND}^{*}).

$$a_{j,k}^{i} \in [-NF_{in}, \quad NF_{in}], \quad j \in \{1, 2, \ldots, m\}, \quad k \in \{1, 2, \ldots, n\} \tag{3}$$

where NF_{in} is the number of fuzzy sets for input j. Also, consequents b_j^i are bounded to

$$b_j^i \in [-NF_{out}, \quad NF_{out}], \quad j \in \{1, 2, \ldots, m\} \tag{4}$$

$$NF_{in}, \quad NF_{out} \in \mathbb{N} \tag{5}$$

with NF_{out}, the number of output sets.

In addition, two values for the connectives are considered: AND and OR connectives are represented by "1" and "2", respectively,

$$c_j^i \in \{1, 2\} \tag{6}$$

In a first stage of the algorithm, the swarm is initialized with a set of *NP* particles randomly distributed in the search space and a single real function *f* is defined making up the objective function or fitness to be improved. The fitness indicates the quality of a given position for particles. Hence, each particle position is updated through iterations with the aim that a better value for the objective function *f* is achieved. To be precise, the position updating is done on the basis of particle velocity matrix modification where velocity at every iteration is presented as a matrix for each particle P_i:

$$V_i = \begin{bmatrix} v_{1,1}^i & v_{1,2}^i & \dots & v_{1,n}^i & v_{1,n+1}^i & v_{1,n+2}^i \\ v_{2,1}^i & v_{2,2}^i & \dots & v_{2,n}^i & v_{2,n+1}^i & v_{2,n+2}^i \\ \dots & \dots & \dots & \dots & \dots & \dots \\ v_{m,1}^i & v_{m,2}^i & \dots & v_{m,n}^i & v_{m,n+1}^i & v_{m,n+2}^i \end{bmatrix} \tag{7}$$

where $v_{j,k}^i \in [V_{min}, V_{max}]$, $j \in \{1, 2, \dots, m\}$, $k \in \{1, 2, \dots, n+2\}$. V_{max} and V_{min} denote the maximum and the minimum values allowed for the velocity, respectively. Moreover, three different velocity components drive the whole optimization: *the inertia component* or particles leaning to keep their current velocity, *the self-recognition component* or inner tendency to return to their best position, $P^{\#}(t)$ and *the social component* that represents the particles inclination to move towards the best position found by the whole swarm, $P^*(t)$. Specifically, particles velocity updating is formulated as

$$V_i(t+1) = \omega \otimes V_i(t) \oplus (d_1 * r_1) \otimes (P^{\#}(t) \ominus P_i(t)) \oplus (d_2 * r_2) \otimes (P^*(t) \ominus P_i(t)) \tag{8}$$

where d_1, d_2 are constant weight factors, r_1, r_2 are random factors in the [0,1] interval, ω represents inertia weight, $\otimes$ indicates multiplication and $\oplus$ and $\ominus$ denote regular addition and subtraction of matrices, respectively. Finally, particles change their location according to the following expression:

$$P_i(t+1) = P_i(t) \oplus V_i(t+1) \tag{9}$$

and this updating process is repeated until the stopping condition is satisfied (i.e., a number of fixed iterations or a minimum required fitness quality). As found in literature [34] about FRBSs knowledge acquisition in massive parallel processing environments, the specification of the stopping condition in bio-inspired optimization algorithms is generally founded on the statistical study of the system behaviour within the particular environment.

A relevant factor for convergence in this learning strategy is related to the suitable setting of inertia weight ω, whereas high values for ω advantages global searches within the considered space, low values fosters local searches. Hence, ω can be fixed to balance both types of searches which may be suitable according to the stage of the optimization and, in this sense, to decrease the necessary iterations to reach an optimum position. To be precise, ω is generally set to decrease its value through iterations in order to intensify local searches around final positions once the global space has been explored. A starting value of 1.2 and progressive decrease until 0 is generally accepted as a good selection for ω [25]. Also, other works [53,54] suggest the consideration of adaptive strategies such as the usage of fuzzy controllers, where control parameters are fixed depending on the considered problem. Specifically, the following expression is used for the inertia weight,

$$\omega(iter) = \omega_0 \cdot e^{(-iter/Num_{iter})c} \tag{10}$$

where ω_0 denotes the initial inertia weight, *iter* represents the current step of the learning process, Num_{iter} is the total number of steps in the learning process and *c* is a constant that allows establishing the convergence speed for ω in a way that a small value for ω just is achieved in the last iterations to prevent the early stagnation of particles in the global search at the same time a fine search of good solutions around final positions or local search in the last iterations is done. On the other hand, given that antecedents, consequents and connectives of rules making up a RB may exceed the search space through the updating process, Eq. (9), some constrains are imposed in every iteration in a way that RBs coherence is kept:

$$a_{j,k}^i = \begin{cases} NF_{in} & if\ a_{j,k}^i > NF_{in} \\ -NF_{in} & if\ a_{j,k}^i < -NF_{in} \end{cases} \tag{11}$$

$$b_{j,k}^i = \begin{cases} NF_{out} & if\ b_{j,k}^i > NF_{out} \\ -NF_{out} & if\ b_{j,k}^i < -NF_{out} \end{cases} \tag{12}$$

$$c_j^i = \begin{cases} 1 & if\ c_j^i < 1 \\ 2 & if\ c_j^i > 2 \end{cases} \tag{13}$$

In addition, in order to keep the knowledge coherence, every rule j of P_i is initialized in KASIA in the advent of a simultaneous null value for its antecedents ($if \ \sum_{k=1}^{n} a^i_{j,k} = 0 \Rightarrow init \ a^i_j$).

It is to be underlined that KASIA is not supported by genetic operators, such as crossover or mutation, but particles update their location just taking into account their own internal velocities. This reduction in the number of control parameters implies a smaller computational effort generally and it also simplifies the setting of the optimization strategy in each problem. On the other hand, particles have memory and take into account unidirectional information exchanges, what significantly reduces the number of communications among individuals needed. Moreover, in contrast to other classical genetic learning strategies, a higher control over particle convergence behaviour can be achieved and as a consequence, the required number of particle movements or RB evaluations can be reduced significantly. Thereby, KASIA strategy is selected for the evolution of RBs in this work.

As introduced above, particles in KASIA learning strategy are moved through the search space in every iteration under the consideration of their own best reached location or best local knowledge, and the whole swarm best reached position or best global knowledge (i.e., $P^{\#}(t)$ and $P^{*}(t)$, respectively). Hence, a relevant aspect in KASIA is the selection of those locations or particles selection mechanism. In this process it is determined which individual of the swarm has reached a better position and also, the best position found for every particle along the whole optimization process. This selection is addressed on the basis of how good or bad a position is. Thereby, some criteria must be established to decide whether a particle location is better than another one. In the scalar case, a direct reasoning can be followed: the better objective fitness f a position has, the better this position is. Nevertheless, in the multi-objective case, not only one criterion is considered to determine if one location is better than another. Thus, since no order of solutions can be set in a natural way founded of the values of fitness and since no priorities have to exist in the set of optimization criteria, positions quality must be ranked considering a non-scalar approach. The multi-objective problem can be formulated as [29]

$$min\, F(P_i) = (f_1(P_i), \ \ldots, f_q(P_i)) \tag{14}$$

$$s.a: \quad P_i \in P_T \tag{15}$$

where P_T is the set of all possible particles or feasible locations of the problem and q is the number of optimization criteria to be considered in the optimization process. In general, the optimization criteria or objective fitness involved f_i can present conflicting or even contradictory interests in a way that optimizing an objective may cause at least the deterioration of another objective. Hence, situations where there exist no solution able to minimize all criteria in a simultaneous way are possible. In this sense, multi-objective evolutionary strategies can offer a set of possible solutions that are considered to be optimal in some criteria. Let two particles of the swarm be denoted as P_1 and P_2, P_1 is said to dominate P_2 if and only if $F(P_1)$ is partially less than $F(P_2)$,

$$f_i(P_1) \le f_i(P_2) \quad \forall i = 1, \ldots, q \tag{16}$$

$$and \quad \exists i \in \{1, \ldots, q\} \quad with \quad f_i(P_1) < f_i(P_2) \tag{17}$$

In this work, the concept of Pareto dominance above is used to select the best positions found by every particle $P^{\#}_{ND}(t)$ and by the whole swarm $P^{*}_{ND}(t)$, i.e., a particle P is said to be non-dominated if no other particle P' found by the algorithm can improve some optimization criteria without worsen simultaneously other solution in at least other optimization criteria. Hence, at every iteration, KASIA strategy is modified in such a way that best solutions are found among the non-dominated ones. To be precise, $P^{*}(t)$ is selected within all the non-dominated solutions within the swarm and $P^{\#}(t)$ is selected among all the non-dominated solutions found by the associated particle during the search. With this aim, Goldberg's ranking method [29] is adapted to KASIA. This approach directly uses the concept of Pareto dominance to define the selection process. It establishes a ranking among the solutions so that the non-dominated solutions of the population have a lower ranking value than the dominated ones and therefore a greater probability of being selected. Also, it assigns equal probability of reproduction to all non-dominated individuals in the population. Thereby, the updating of every particle velocity can be formulated as

$$V_i(t+1) = \omega \otimes V_i(t) \oplus (d_1 * r_1) \otimes (P^{\#}_{ND}(t) \ominus P_i(t)) \oplus (d_2 * r_2) \otimes (P^{*}_{ND}(t) \ominus P_i(t)) \tag{18}$$

where $P^{\#}_{ND}(t)$ denotes the best position or non-dominated position found by particle P_i during the search process up to iteration t

$$f_i(P^{\#}_{ND}(t)) \le f_i(P_{T_i}) \quad \forall i = 1, \ldots, q \tag{19}$$

$$and \quad \exists i \in \{1, \ldots, q\} \quad with \quad f_i(P^{\#}_{ND}(t)) < f_i(P_{T_i})) \tag{20}$$

with P_{T_i} all the solutions found by particle P_i. On the other hand, $P^{*}_{ND}(t)$ indicates the best position or non-dominated position found the whole swarm during the optimization until iteration t

$$f_i(P^{*}_{PO}(t)) \le f_i(P_T) \forall i = 1, \ldots, q \tag{21}$$

$$and \quad \exists i \in \{1, \ldots, q\} \quad with \quad f_i(P^{*}_{ND}(t)) < f_i(P_T)) \tag{22}$$

with P_T representing all the solutions found by the swarm. It must be mentioned that in case of obtaining multiple non-dominated solutions as a result of the learning strategy, the non-dominated base to be used by the scheduler, is selected according to the established preferences for the system that are associated to the purpose and the state of the grid system that must be obtained through a later automatic process. Hence, as a result of the multi-objective learning strategy, the scheduler can obtain a set of non-dominated solutions according to the different optimization objectives. This provides a set of knowledge bases that can offer different levels of quality in terms of the diverse optimization objectives. In this sense, to allow the scheduling strategy to offer guarantees in its performance, the learning process must be enough exhaustive to acquire knowledge bases that are robust to address dynamic environmental conditions. Thereby, in this work the theory of Pareto optimization is used to let expert schedulers learn on multiple objectives simultaneously and find non-dominated solutions. Following this strategy the system is able to obtain quality RBs on multiple objectives which allow an efficient management of the grid in different perspectives either by the optimization of an objective or offer of a balanced solution between different objectives. On the other hand, the convenience of using one of the different bases obtained in the simultaneous optimization is to be determined through a later automatic learning process. Through this automatic process, the conditions of the grid in which is favorable to use these bases are learnt. The suggested multi-objective KASIA strategy is summarized in Algorithm 1.

Among all the optimization objectives in grid computing, the importance of *makespan* and *flowtime* is to be underlined. *Makespan* and *flowtime* are contradictory optimization objectives, i.e., the minimization of *makespan* leads to the maximization of *flowtime* and vice versa [31,55], whose simultaneous minimization is desired. The minimization of the latest job finalization time or *makespan* is formulated as [3,4],

$$min_{S_i \in Sched}\{max_{j \in J} T_j\} \tag{23}$$

where T_j denotes the time when job J_j finishes, *Sched* indicates all the possible schedules and J represents the set of considered jobs, and the minimization of *flowtime*:

$$min_{S_i \in Sched}\left\{\sum_{j \in J} T_j\right\} \tag{24}$$

On the one hand, *makespan* is generally presented as a general grid productivity index [3,4] and good scheduling results are associated to small values of this indicator. On the other hand, *flowtime* is considered to be a critic criteria for interactive applications in grid computing and it is taken into account to provide a grid performance index from users perspective. The combined consideration of these criteria can be extensively found in literature [3,4,25] as presented in Section 2 to optimize grid schedulers in terms of productivity or throughput and jobs response time simultaneously. Hence, they are also considered in this work to test the proposed strategy.

5. Simulation results and discussion

Considering the difficulties associated to tests in real settings, the fuzzy meta-scheduler with multi-objective KASIA learning is analyzed through simulations. To be precise, a Gridsim-based toolkit, Alea, is considered in its last version 2.1 [56]. Alea software allows the study of scheduling strategies in settings and workload conditions founded on traces obtained from real world. Particularly, in this work, a scenario based on the Czech National Grid Infrastructure Metacentrum project [57] is proposed. Metacentrum is associated to CESNET (operator of academic network of the Czech Republic -National Research and Education Network, NREN) whose final goal is to contribute towards the development of a large virtual computational infrastructure by the cooperation of multi-institutional resources worldwide. Also, Metacentrum has contributed to several international grid projects such as EGEE III, EuAsiaGrid and EGI_DS Metacentrum. The suggested scenario is based on 14 Metacentrum sites integrating 210 machines and a set of 806 of heterogeneous CPUs. Table 1 summarizes sites features.

Table 1
Metacentrum-based grid resources.

Cluster	CPU speed (MHz)	Main memory size (KB)	CPU type	Operating system	Number of machines	Total number of CPUs
cluster_0	1500	48,000,000	Itanium2	Linux	1	8
cluster_1	2200	32,000,000	Opteron	Linux	1	16
cluster_2	3200	1,009,000	Xeon	Linux	10	10
cluster_3	2600	131,182,840	Opteron	Linux	5	80
cluster_4	1600	1,005,000	AthlonMP	Linux	16	32
cluster_5	2400	1,048,576	Xeon	Linux	32	64
cluster_6	2659	15,565,060	Xeon	Linux	36	148
cluster_7	3056	2,021,000	Xeon	Linux	35	70
cluster_8	1600	1,024,000	Opteron	Linux	10	20
cluster_9	2400	4,000,000	Opteron	Linux	3	6
cluster_10	2000	4,000,000	Opteron	Linux	23	92
cluster_11	3000	4,546,800	Xeon	Linux	19	152
cluster_12	2660	27,343,000	Xeon	Linux	8	64
cluster_13	2360	15,200,000	Xeon	Linux	11	44

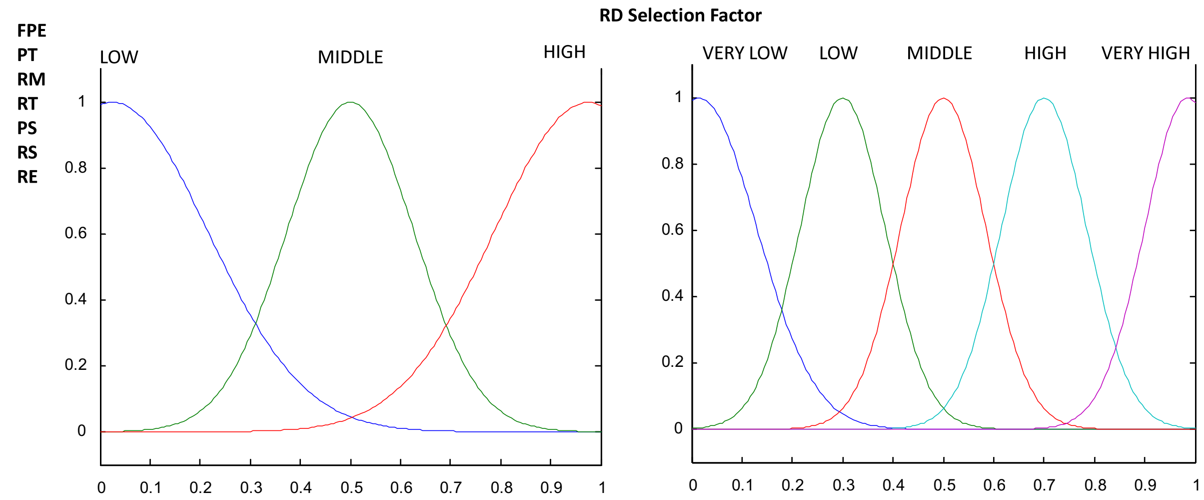


Fig. 2. Fuzzy sets for grid features and resource domain selector.

Moreover, resources maintenance and reservation behaviour (i.e., initial time a machine fails, becomes reserved, dedicated or unavailable and associated duration), resources queues setting parameters, and workload characterization are obtained from Metacentrum facilities traces retrieved in January 2009, available at [58]. Workload in our scenario is made up by a set of jobs that arrive dynamically and specify the required properties for the target resource in order to satisfy compatibility. A job represents a user's application that is associated to an arrival time and computational need which can require sequential or parallel usage of processing units. The used workload traces follow the MWF format (Metacentrum Workloads Format) [36,58] where the job identifier (job ID), associated job user or owner priority, list of properties to be met in the target machine (i.e., number of CPU, CPU type, operating system, etc.) and arrival time at the scheduler are indicated for every considered job [36]. Specifically, in our simulations we consider traces of 2000/2400 jobs (for training and validation, respectively) to show the performance of the proposed learning strategy.

In this scenario, the fuzzy rule-based meta-scheduler is learned through the proposed multi-objective knowledge acquisition strategy based on KASIA. Fig. 2 illustrates these features associated fuzzy sets for the input features of the scheduler as presented in Section 3 and output. As shown, three gaussian shaped fuzzy sets are considered to characterize input features, corresponding to *low*, *medium* and *high* levels. Also, Fig. 2 shows the scheduler single output or resource domain selector fuzzy sets representing *very low*, *low*, *medium*, *high and very high* levels. These sets can mathematically expressed as

$$\mu_i^{(x_m)}(z) = \frac{1}{\sigma_i^{(x_m)}\sqrt{2\pi}} \exp\left\{\frac{-(z-\tau_i^{(x_m)})^2}{2\sigma_i^{(x_m)2}}\right\} \quad \left\{z \in \mathbb{R}^+ \mid z \leq 1\right\} \tag{25}$$

where $\tau_i^{(x_m)}$ and $\sigma_i^{(x_m)}$ represent the mean and standard deviation.

Furthermore, a classical learning strategy in FRBSs, genetic Pittsburgh approach, is considered for comparison. This strategy is selected in a way that the suggested schema can be fairly compared in terms of computational effort to classical learning strategies in FRBSs. It is to be noted that in contrast to other classical learning schemas such as Michigan [52], in Pittsburgh approach every individual of the population makes up a RB to be evolved through a set of iterations so-called generations [51]. Hence, every generation involves the evaluation of a set of RBs which is also the case of the proposed multi-objective KASIA approach that considers the evaluation of every swarm individual at every iteration. Moreover, Pittsburgh approach has shown a better accuracy in the optimization of fuzzy rule-based meta-schedulers knowledge [10]. Configuration for both strategies are shown in Tables 2 and 3. It must be pointed out that the sizes for the RBs for these strategies are based on previous studies in the initialization of RBs in the learning of fuzzy rule-based meta-schedulers in grid computing for the same number of variables and fuzzy sets [59]. Further, as in the case of the proposed strategy, in the Pittsburgh approach the determination of the quality of the bases, which is necessary for the selection process, is based on the Pareto dominance concept and the Goldberg's ranking strategy [29].

Fig. 3 shows the evolution of non-dominated solutions on average in 30 experiments for swarm and genetic-based strategies considering grid performance contradictory objectives (i.e., *makespan* and *normalized flowtime* [3]) though 100 iterations with sampling iterations [25 40 55 70 85 100]. As it can be observed, the accuracy of the generated solutions is improved with the MO-KASIA strategy. Also, it is shown this strategy is able to exercise a deeper exploration of the search space around final locations than the genetic approach. It is to be noted that the distance of non-dominated solutions for MO-KASIA in convergence locations is significantly reduced with respect to non-dominated solutions in previous ending iterations. In this sense, the role of inertia weight setting must be pointed out. Inertia weigh ω allows the configuration of converge rate and stagnation level in the swam-based strategy. However, in genetic-based algorithms stagnation occurs

Table 2
Parameters configuration for MO-KASIA approach.

Parameters configuration				
MO-KASIA	$\omega = 0.9$	$d_1 = 2,\ d_2 = 2$	Number of particles/RBs(NP) = 18	$RB_{size} = 15$

Table 3
Parameters configuration for MO-Pittsburgh approach.

Parameters configuration				
MO-Pittsburgh	Selection rate (elitism) $\lambda = 0.9$	Mutation rate $= 0.1e^{(-iter/Num_{iter})}$	Population size(PS) = 20	init max $RB_{size} = 20$

Table 4
Training results in grid Metacentrum for the swarm-based and genetic-based learning strategies.

Result (s)	Max	Min	Avg.	Standard deviation	Confidence interval (95%)
MO-KASIA (makespan)	1,961,649.032	1,625,058.071	1,721,337.030	95,005.094	1,685,861.544 , 1,756,812.515
MO-KASIA (flowtime)	74,359.577	65,687.445	67,929.438	1,812.246	67,252.735 , 68,606.142
MO-Pittsburgh (makespan)	1,937,984.100	1,648,163.117	1,789,834.330	87,350.822	1,757,216.996 , 1,822,451.663
MO-Pittsburgh (flowtime)	69,549.474	66,507.055	67,923.402	716.827	67,655.734 , 68,191.069

when every individual presents the same genetic code and thus, crossover has little or non effect on the population. Hence, the swam-based strategy can prevent this phenomenon. To be precise, a high inertia weight is considered in a way that particles in the swarm are allowed to stay in a sawing motion around the best global solution and so best fitness locations may be found in convergence positions. Thereby, it is shown that a good set in Pareto dominance sense can be obtained by the multi-objective consideration of the KASIA approach which also achieves an efficient guiding of the search process.

Additionally, the final average non-dominated solutions of the learning process (corresponding to Fig. 3f) and the final solutions for the 30 experiments and both strategies are illustrated together in Fig. 4. It can be observed that the multi-objective KASIA approach is able to achieve a greater accuracy in the simultaneous optimization of grid performance opposed optimization criteria both on average non-dominated and final solutions. Also, as shown in Fig. 4b, the proposed approach increases the diversity of the final solutions which is relevant regarding the provision of different levels of QoS in the grid. Further, Table 4 summarizes statistical results. To be precise, results are studied in terms of minimum (Min), maximum (Max), average (Avg), standard deviation and 95% confidence interval regarding both optimization criteria. The fact that the multi-objective KASIA outperforms genetic-based strategy on average by 3.83% in terms of *makespan* is illustrated. However, it is also shown that both strategies results in terms on average *flowtime* are similar. Furthermore, multi-objective swarm-based strategy best location (Min) improves in 1.4% in terms of *makespan* and 1.23% *flowtime*, respectively, what proves the swam-based strategy ability to achieve a deeper exploration of the search space. On the other hand, it is also observed that multi-objective KASIA presents higher maximum values in terms of *flowtime*. This could be expected given that this strategy is able to reach the minimum values in *makespan* what generally leads to an increase in *flowtime*.

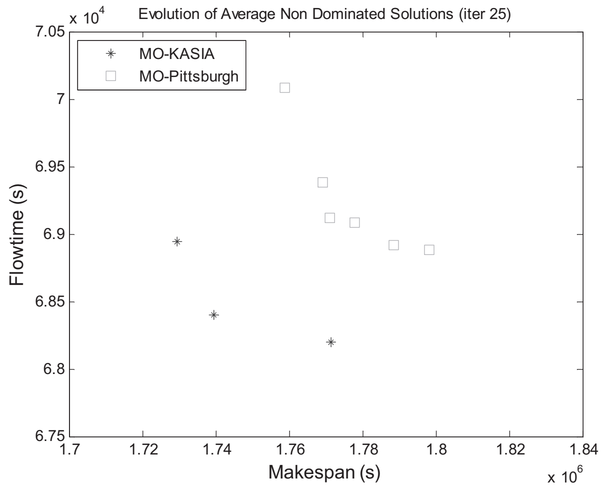
In order to analyze the superiority of our learning strategy non-parametric statistical tests are conducted. Non-parametric statistical tests are useful to contrast whether the distribution of a variable is the same for two populations or if this variable tends to differ in one of them considering sampling data. In this work it could be interesting to study the results for *makespan* and *flowtime* for the two learning strategies which are compared: MO-KASIA approach and Pittsburgh approach. First, to determine if it is possible to make non-parametric tests, the normality of the sampling is studied. Results for the two approaches show that there is no normality in the samples and thus, a non-parametric statistical test can be made to compare the methods. Specifically, a non-parametric test based on the sum of ranges of Wilcoxon or Mann-Whitney is considered for comparison, which is a well-known method available if several statistical applications such as R [60] or KEEL [61]. To make the comparison, two observations have been made: the first one related to *makespan* and the second one related to *flowtime*.

Observation 1

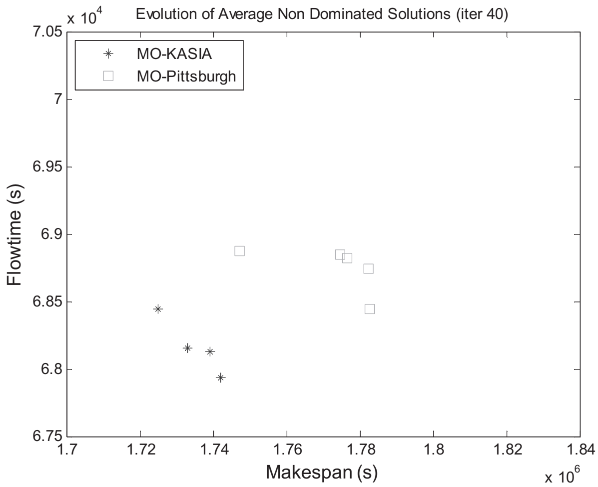
The values of makespan for the two different populations (MO-KASIA and Pittsburgh approaches) with sampling data of size 30, respectively, are considered as in previous sections. The null hypothesis (i.e., the hypothesis that is intended to be contrasted) is that the sum of the ranges of the Pittsburgh-MO approach is higher than the one associated to the MO-KASIA approach. In our case, the obtained value for Wilcoxon W is 649, and the confidence value (p-value) is 0.001646. Thus, the analysis confirms the hypothesis. Therefore, the value for the first objective, i.e., *makespan*, is higher for the Pittsburgh approach, and thus, a worst performance is observed for the Pittsburgh scheduler.

Observation 2

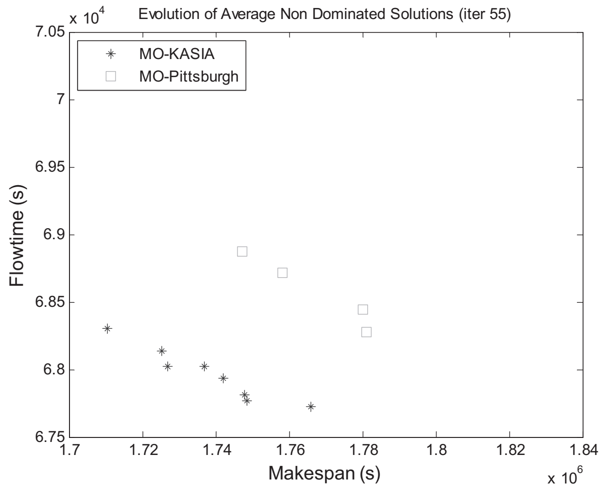
In the first observation, results for makespan have been analyzed. In this observation, results for *flowtime* with sampling data of size 30 and the two different populations (MO-KASIA and Pittsburgh approaches) are studied. As in the previous observation, the null hypothesis shows that the sum of the ranges in the Pittsburgh-MO approach is higher than the MO-KASIA approach one. To be precise, in this case, Wilcoxon W is 567, and the confidence value (p-value) is 0.0425. Thus, the hypothesis is validated after the analysis. This way, the value for the second objective, i.e., *flowtime*, is higher for the Pittsburgh-MO approach, as it occurs in observation 1. Table 6 summarizes the results of the non-parametric tests.



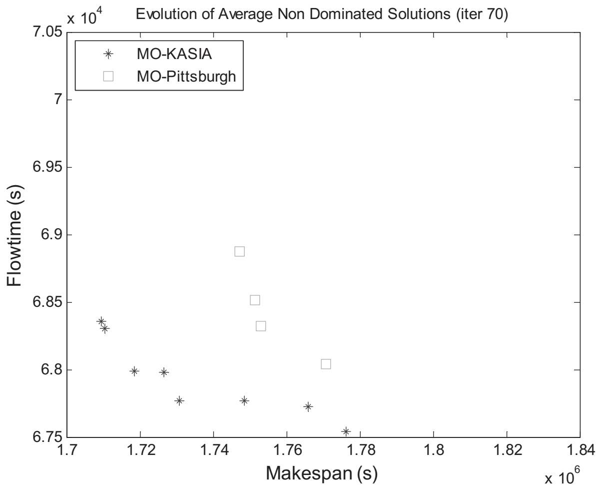
(a) Average non-dominated solutions. Iteration 25.



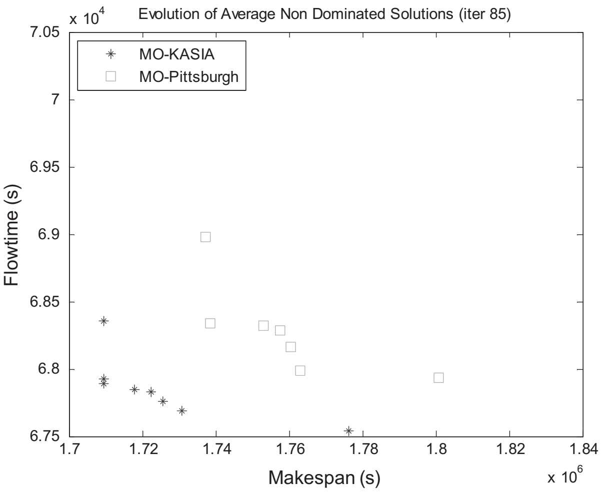
(b) Average non-dominated solutions. Iteration 40.



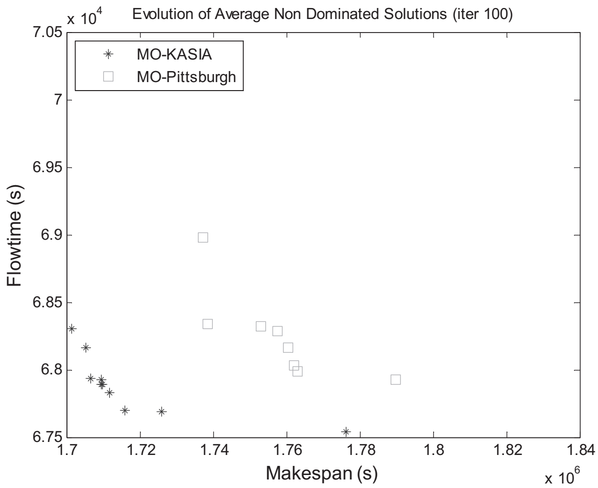
(c) Average non-dominated solutions. Iteration 55.



(d) Average non-dominated solutions. Iteration 70.



(e) Average non-dominated solutions. Iteration 85.



(f) Average non-dominated solutions. Iteration 100.

Fig. 3. Evolution of average non-dominated solutions for genetic and swarm-based learning approaches through sampling iterations.

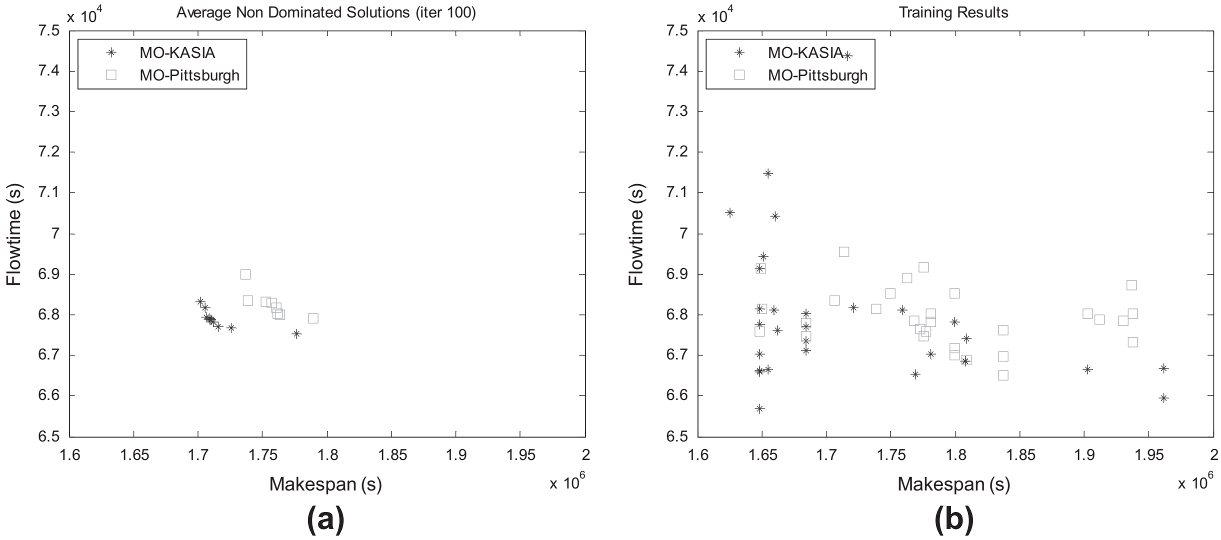


Fig. 4. Final average non-dominated solutions in the evolution and final training results in 30 experiments with the swarm-based and genetic-based learning strategies.

Table 5
Scheduling strategies results for MO-KASIA, EASY-BF, FCFS, ESG, ESG + LS and ESG + LS periodical in validation scenario.

Metric/strategy	MO-KASIA	EASY-BF	FCFS	ESG	ESG + LS	ESG + LS periodical
Makespan (s)	1,696,475.511	1,749,586.008	1,944,220.064	1,973,151.408	1,973,151.408	1,973,151.408
Flowtime (s)	87,390.805	87,491.471	88,721.759	91,395.383	95,474.501	83,379.182
Number delayed jobs	174.5	125.0	172.0	143.0	156.0	151.0
Weighted usage (%)	44.04	44.47	33.55	34.73	34.14	34.74
Classic usage (%)	53.01	47.01	38.63	41.45	40.54	40.91
Tardiness (s)	4,580.235	3,235.311	4,346.803	1,311.627	2,159.932	1,274.822
Wait time (s)	44,917.39	43,897.14	45,176.48	37,974.54	45,756.35	31,485.74
Slowdown (s)	186.965	184.352	186.572	15.454	24.839	17.522
Awrt (s)	156,642.650	210,885.070	193,895.867	256,401.198	239,370.660	244,917.654
Awsd (s)	1.76414	1.70910	1.74431	1.34796	1.51796	1.32881

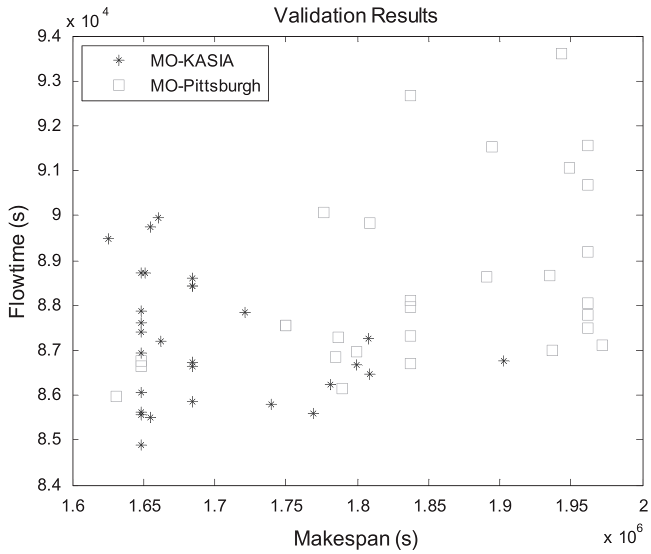


Fig. 5. Validation results in 30 experiments with the swarm-based and genetic-based learning strategies.

In addition, the acquired knowledge in 30 experiments for both strategies are evaluated in a validation grid scenario. To be precise, Metacentrum-based scenario workload is increased by 16.67% (a total of 2400 jobs) and an alternative machine failure and reservation behaviour is considered. Fig. 5 presents rule-bases results in the new setting. As observed, multi-objective KASIA acquired fuzzy rule bases are able to keep their greater accuracy with respect to genetic-based approach what

Table 6
Results summarization of non-parametric tests based on the sum of ranges of Wilcoxon.

Observation index/parameter	Wilcoxon W	*P*-value	Hypothesis
Makespan	649	0.001646	Validated: *makespan*, is higher for the Pittsburgh-MO approach
Flowtime	567	0.0425	Validated: *flowtime*, is higher for the Pittsburgh-MO approach

shows the robustness of its knowledge in dynamic conditions. Moreover, in order to further evaluate the meta-scheduler performance within the grid environment and the quality of the obtained rule bases, the fuzzy scheduler performance is studied in a range of complementary grid performance criteria, namely, *number of delayed jobs* [21], *weighted usage* [62], *classic usage* [63], *tardiness* [48], *wait time* [63], *slowdown* [63], *average weighted response time* (*awrt*) [63] and *average weighted slowdown* (*awsd*) [63]. Further, the performance of the fuzzy meta-scheduler with multi-objective KASIA learning strategy is compared to that of some classical scheduling strategies in grid systems nowadays considering the same scenario in the same configuration. Recent works in the field of grid and cluster scheduling are focused on both queue and schedule based techniques and thus, a comparison with these two types of scheduling algorithms are provided. On the one hand, regarding queue-based strategies, a comparison to *FCFS* [19] and *EASY-BF* [18] is entailed. On the other hand, performance of the proposed fuzzy strategy with respect to schedule-based strategies is analyzed. Schedule-based techniques are considered in simulations by Earliest Suitable Gap (ESG) strategy and Local Search (LS) based methodologies, namely *ESG*, *ESG + LS* and *ESG + LS periodical* [20–22], where ESG is the schedule-based version of the queue-based backfilling strategy founded on the CCS system algorithm [64]. Table 5 presents results for these strategies taking into account both the grid optimization and complementary performance criteria.

First, in order to compare the validation results for all the scheduling strategies, the next statistical procedure is followed for the two optimization objectives, *makespan* and *flowtime* and every classical scheduling strategy. First, it is supposed that μ_0 is the obtained value for the optimization objective by the classical scheduling strategy and μ is the average result of the fuzzy strategy with MO-KASIA learning. Next, an hypothesis H_o is considered: the value for the optimization objective of the classical strategy equals the average result obtained by the fuzzy scheduler with MO-KASIA, i.e., $\mu_0 = \mu$. This hypothesis is to be contrasted with the alternate hypothesis H_1: the average result in the optimization criteria of the fuzzy strategy with MO-KASIA is lower than this value obtained with the classical strategies, i.e., $\mu_0 > \mu$. Finally, we calculate μ and we confirm that the hypothesis H_1 is true against the hypothesis initially considered H_o. Hence, we verify that the average in the both optimization objectives obtained with MO-KASIA are lower than the ones obtained with the classical scheduling methods. Specifically, it is shown that the fuzzy scheduler with swarm-based learning outperforms *makespan* by 3.04% the most precise of the rest of strategies in this sense, (i.e., *EASY-BF*), at the same time it is able to keep the scheduler performance quality in terms of *flowtime*. On the other hand, it is also observed that *ESG + LS periodical* strategy achieves the most accurate solution in *flowtime*. Nevertheless, it must be pointed out that this improvement is associated to the worst performance in terms of *makespan*. Specifically, *makespan* is 14.02% less efficient than the one achieved with the fuzzy scheduler with multi-objective swarm-based learning. Besides, as far as machine usage is concerned, it is shown that the fuzzy meta-scheduler achieves a major harnessing of grid resources. To be precise, it outperforms in terms of *classic usage* in with respect to the best competitor in this sense, *EASY-BF*, by 11.33%. This result was expected since a significant reduction of jobs completion time is generally associated to a more efficient harnessing of resources. Also, it is significant to highlight the scheduler efficient performance in *awrt* in comparison to the rest of strategies in spite of *slowdown* deterioration regarding *ESG-based* strategies.

Finally, the evolution of all the scheduling strategies performance is compared graphically through the whole grid operation. Specifically, the performance of the fuzzy meta-scheduler using the RB with best trade-off in terms of *makespan* and *flowtime* is compared given the relevance and difficulty of minimizing in these objectives simultaneously in a grid system [3,55]. In Fig. 6 the execution of jobs regarding waiting and running states together with the requested CPU is presented. It is observed that the fuzzy scheduling with swarm-based learning is the only strategy able to achieve no waiting jobs states up to the advent of workload peaks (i.e., days between 12 and 16). Moreover, an analog behaviour can be detected regarding used and available CPUs as also shown in Fig. 6. In addition, as discussed for Table 5, jobs execution is significantly faster for the fuzzy scheduler and it can be appreciated in the time axis. On the other hand, Fig. 7 illustrates grid resources usage in terms of cluster and machine average usage per day. Figs. 7(a)-(f) illustrate the cluster usage per day during the whole simulation time in a gradual scale where colors close to dark red and dark green represent high and low usage rates, respectively. Also, black indicate that the cluster is down. It can be appreciated that the fuzzy scheduler with MO-KASIA successes in providing a high usage for a range of clusters up to the end of jobs execution in contrast to the rest of strategies, where low usage rates are frequent in the final stages of workload completion. As shown in Fig. 7(a), all the clusters with the scheduler with MO-KASIA learning reach high rates of utilization in the simulation and workload is not concentrated in a set of clusters as it occurs in *EASY-BF or FCFS*, Fig. 7(b) and (c). For instance, note that clusters 5 and 13 are not employed at all during the operance of *EASY-BF* strategy and that they are scarcely used in *FCFS*. Also, in Fig. 7(e)-(l) it is observed that the proposed fuzzy schema achieves a more regular average machine usage per day through the whole simulation with respect to that of the classical scheduling strategies. Finally, Table 7 shows an example of RB obtained with the proposed strategy.

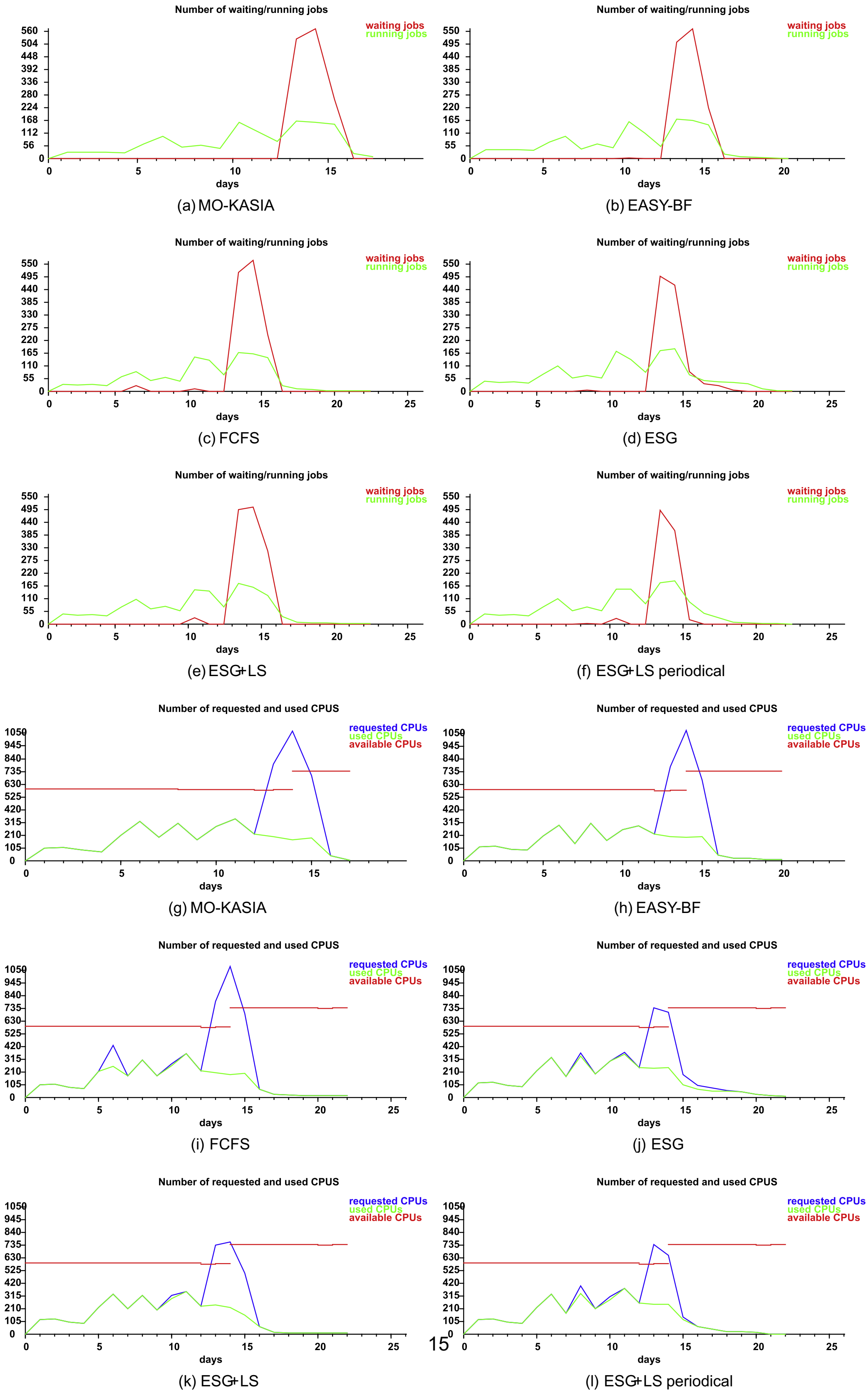


Fig. 6. Waiting and running jobs/requested CPU evolution for scheduling strategies.

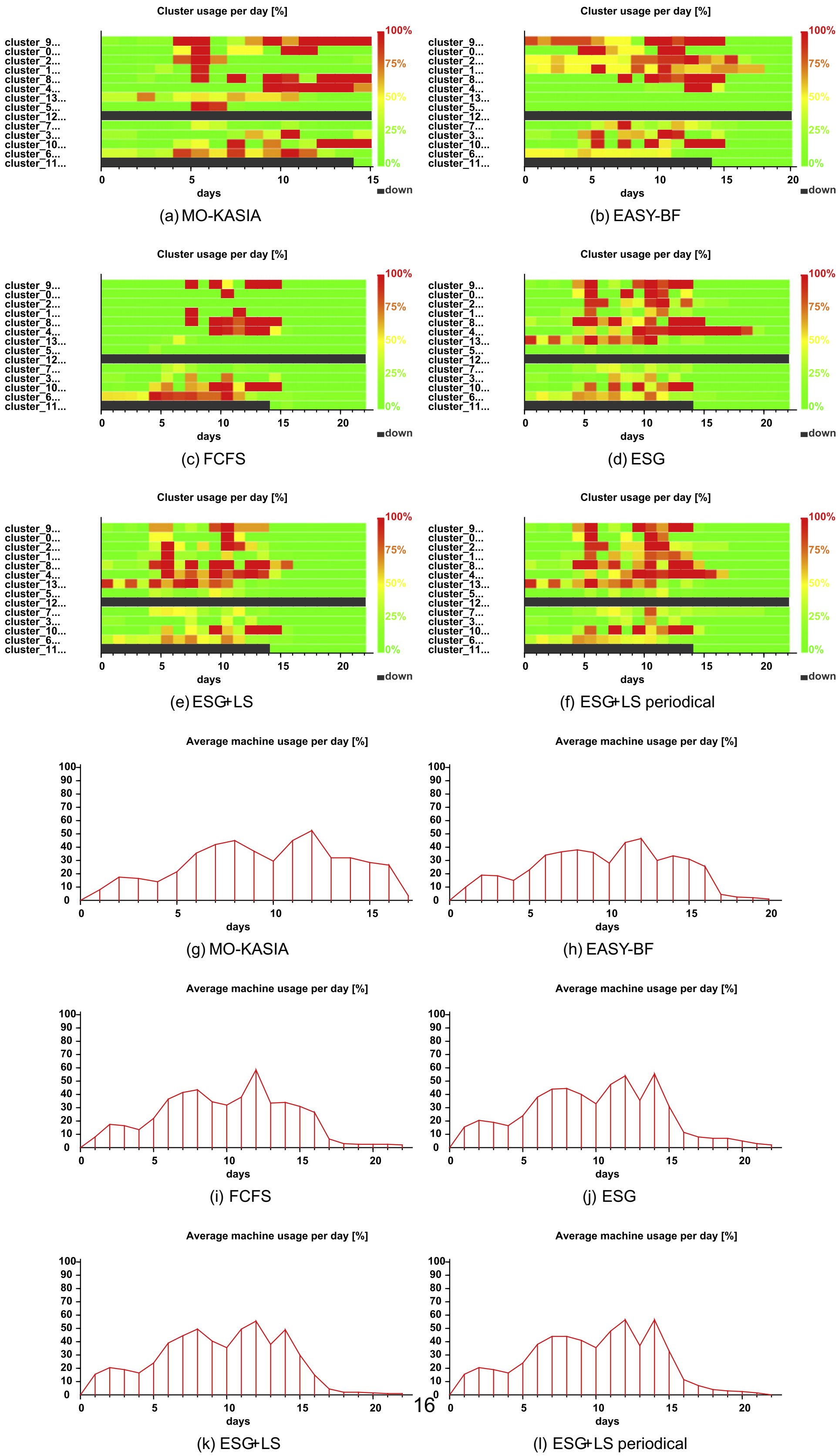


Fig. 7. Sites/machines day usage evolution for scheduling strategies.

Table 7
Example of obtained RB with MO-KASIA strategy.

Rule base obtained with MO-KASIA
1. If (FPE is LOW) and (RM is not LOW) and (RT is LOW) and (PS is LOW) and (RS is not LOW) and (RE is not HIGH) and (MEM is LOW) then (OUTPUT is not HIGH) (1)
2. If (FPE is not HIGH) and (RT is not LOW) and (RS is not LOW) and (RE is not LOW) and (MEM is LOW) then (OUTPUT is not VERYLOW) (1)
3. If (RM is not LOW) and (RT is not LOW) and (PS is not LOW) and (RS is LOW) and (RE is not LOW) and (MEM is not MIDDLE) then (OUTPUT is VERYHIGH) (1)
4. If (FPE is LOW) and (RM is LOW) and (RS is not HIGH) and (RE is not HIGH) and (MEM is not HIGH) then (OUTPUT is LOW) (1)
5. If (FPE is not MIDDLE) or (RM is not HIGH) or (RT is LOW) or (PS is not MIDDLE) or (RS is not LOW) or (RE is not MIDDLE) or (MEM is not MIDDLE) then (OUTPUT is LOW) (1)
6. If (RM is not HIGH) and (RT is not MIDDLE) and (PS is not LOW) and (RE is not LOW) and (MEM is not LOW) then (OUTPUT is not VERYLOW) (1)
7. If (FPE is not HIGH) and (PS is not LOW) and (RS is LOW) and (MEM is LOW) then (OUTPUT is MIDDLE) (1)
8. If (FPE is not MIDDLE) and (RM is not MIDDLE) and (RS is LOW) and (RE is LOW) and (MEM is not MIDDLE) then (OUTPUT is not VERYHIGH) (1)
9. If (FPE is not LOW) or (RT is not HIGH) or (PS is not LOW) or (RS is not HIGH) or (MEM is not LOW) then (OUTPUT is VERYLOW) (1)
10. If (FPE is LOW) and (RM is not MIDDLE) and (RT is not MIDDLE) and (PS is not MIDDLE) and (RS is not LOW) and (RE is LOW) and (MEM is not LOW) then (OUTPUT is not LOW) (1)
11. If (PS is not LOW) or (RS is MIDDLE) or (RE is not MIDDLE) or (MEM is not HIGH) then (OUTPUT is not MIDDLE) (1)
12. If (FPE is not HIGH) and (RM is not HIGH) and (RT is not LOW) and (PS is not HIGH) and (RS is not LOW) and (RE is not MIDDLE) and (MEM is LOW) then (OUTPUT is not LOW) (1)
13. If (FPE is not HIGH) or (RM is not HIGH) or (RT is LOW) or (PS is not MIDDLE) or (RS is not MIDDLE) then (OUTPUT is not MIDDLE) (1)
14. If (FPE is not HIGH) and (RM is not LOW) and (PS is LOW) and (RS is not MIDDLE) and (RE is MIDDLE) and (MEM is not MIDDLE) then (OUTPUT is LOW) (1)
15. If (FPE is LOW) and (RM is not HIGH) and (RT is HIGH) and (PS is MIDDLE) and (RS is MIDDLE) and (MEM is not LOW) then (OUTPUT is VERYLOW) (1)

6. Conclusions

The provision of QoS in grid scheduling requires the consideration of the resources state and thus, dealing with the high dynamism and uncertainty of resources is relevant. Fuzzy rule-based schedulers are knowledge-based systems characterized by their ability to cope with imprecisions in the grid state and are emerging as an alternative for the scheduling problem. However, an expert fuzzy scheduler performance is strongly related to the quality of its knowledge and in this sense, with the learning or optimization processes. Also, the scheduling problem is multi-objective in its general formulation and the provision of QoS needs that several objectives are considered in the scheduler optimization. Furthermore, these objectives can present conflicting or contradictory interests in a way that optimizing an objective may cause at least the deterioration of another one. In this work, the general Pareto optimization theory is considered for the adaptation of the KASIA swarm-based learning strategy to the multi-objective optimization of fuzzy rule-based schedulers in grid computing. Simulations results show that this multi-objective learning approach in the optimization of contradictory optimization criteria (i.e., *makespan* and *flowtime*) is able to increase the accuracy and robustness of its final solutions with respect to classical learning strategies in FRBSs with the same computational effort. Moreover, the performance of the fuzzy scheduler with swarm-based learning improves that of classical strategies in grid computing, *EASY-BF*, *FCFS*, *ESG*, *ESG + LS* and *ESG + LS periodical*. Specifically, it is observed that the multi-objective fuzzy rule-based scheduler operation improve results in the objectives used for optimization and a more efficient harnessing of grid resources.

Acknowledgments

This work has been financially supported by the Andalusian Government (Research Project P06-SEJ-01694). The Metacentrum workload log was generously provided by the Czech National Grid Infrastructure Metacentrum.

References

[1] I. Foster, C. Kesselman, The Grid 2: Blueprint for a New Computing Infrastructure, Morgan Kaufman Publishers Inc., San Francisco, CA, USA, 2003.
[2] F. Dong, S.G. Akl, Scheduling algorithms for grid computing: state of the art and open problems, Technical Report No. 2006-504, School of Computing, Queen's University Kingston, Ontario.
[3] F. Xhafa, A. Abraham, Computational models and heuristic methods for grid scheduling problems, Future Generation Computer Systems 26 (4) (2010) 608–621.
[4] F. Xhafa, A. Abraham, Meta-heuristics for grid scheduling problems, Metaheuristics for Scheduling: Distributed Computing Environments, Studies in Computational Intelligence, Springer Verlag, Germany, 2008, ISBN 978-3-540-69260-7.
[5] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman & Co., New York, NY, USA, 1979.
[6] S. Farzi, Efficient job scheduling in grid computing with modified artificial fish swarm algorithm, International Journal of Computer Theory and Engineering 1 (2009) 1793–8201.
[7] D.A. Menascá, E. Casalicchio, Qos in grid computing, IEEE Internet Computing 8 (4) (2004) 85–87.
[8] I.T. Foster, Globus toolkit version 4: software for service-oriented systems, in: H. Jin, D.A. Reed, W. Jiang (Eds.), NPC, Lecture Notes in Computer Science, vol. 3779, Springer, 2005, pp. 2–13.
[9] C. Franke, F. Hoffmann, J. Lepping, U. Schwiegelshohn, Development of scheduling strategies with genetic fuzzy systems, 10.1016/j.asoc.2007.05.009Applied Soft Computing 8 (1) (2008) 706–721.
[10] R.P. Prado, S. García-Galán, A.J. Yuste, J.E. Muñoz-Expósito, A fuzzy rule-based meta-scheduler with evolutionary learning for grid computing, 10.1016/j.engappai.2010.07.002Engineering Applications of Artificial Intelligence 23 (7) (2010) 1072–1082.

[11] R.P. Prado, S. García-Galán, J.E. Muñoz-Expósito, A.J. Yuste, Knowledge acquisition in fuzzy rule based systems with particle swarm optimization, 10.1109/TFUZZ.2010.2062525IEEE Transactions on Fuzzy Systems 18 (6) (2010) 1083–1097.
[12] P.-A. Monney, M.W. Chan, E.H. Ruspini, Special section on dependence issues in knowledge-based systems, 10.1016/j.ijar.2011.01.006International Journal of Approximate Reasoning 52 (3) (2011) 297–298.
[13] R. Belohlavek, G.J. Klir, E.C. Way III, Concepts and fuzzy sets: misunderstandings, misconceptions, and oversights, International Journal of Approximate Reasoning 51 (1) (2009) 23–34.
[14] O. Cordon, A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: designing interpretable genetic fuzzy systems, 10.1016/j.ijar.2011.03.004International Journal of Approximate Reasoning 52 (6) (2011) 894–913.
[15] J. Sanz, A. Fernandez, H. Bustince, F. Herrera, A genetic tuning to improve the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets: degree of ignorance and lateral position, 10.1016/j.ijar.2011.01.011International Journal of Approximate Reasoning 52 (6) (2011) 751–766.
[16] R. Steuer, Multiple criteria optimization: theory, computation and application, Series in Probability and Mathematical Statistics, 1992.
[17] M. Ehrgott, X. Gandibleux, Approximative solution methods for multiobjective combinatorial optimization, TOP: An Official Journal of the Spanish Society of Statistics and Operations Research 12 (1) (2004) 1–63.
[18] J. Skovira, W. Chan, H. Zhou, D.A. Lifka, The easy – loadleveler api project, in: Proceedings of the Workshop on Job Scheduling Strategies for Parallel Processing, Springer-Verlag, London, UK, 1996, pp. 41–47.
[19] C.L. Liu, J.W. Layland, Scheduling algorithms for multiprogramming in a hard-real-time environment, Journal of the ACM 20 (1973) 46–61.
[20] D. Klusacek, L. Matyska, H. Rudova, Alea – grid scheduling simulation environment, in: R. Wyrzykowski, J. Dongarra, K. Karczewski, J. Wasniewski (Eds.), 7th International Conference on Parallel Processing and Applied Mathematics Parallel Processing and Applied Mathematics, Lecture Notes in Computer Science, vol. 4967, Intel; Microsoft; IBM; Action SA; SIAM, Springer-Verlag Berlin, Heidelberger, Berlin, Germany, 2008, Gdansk, Poland, September 09–12, 2007, pp. 1029–1038.
[21] D. Klusacek, H. Rudova, R. Baraglia, M. Pasquali, G. Capannini, Comparison of multi-criteria scheduling techniques, in: S. Gorlatch, P. Fragopoulou, T. Priol (Eds.), Grid Computing: Achievements and Prospects, European Commiss, Network Excellence CoreGRID Fund, Springer, New York, United States, 2008, coreGRID Integration Workshop 2008, Hersonissos, Greece, April 02–04, 2008, pp. 173–184.
[22] D. Klusacek, H. Rudova, Improving qos in computational grids through schedule-based approach, in: Scheduling and Planning Applications Workshop at the Eighteenth International Conference on Automated Planning and Scheduling (ICAPS 08), Sydney, Australia, 2008.
[23] J.C.F. Xhafa, A. Abraham, Genetic algorithm based schedulers for grid computing systems, International Journal of Innovative Computing, Information and Control 3 (5) (2007) 1053–1071.
[24] F. Xhafa, A hybrid evolutionary heuristic for job scheduling on computational grids, in: A. Abraham, C. Grosan, H. Ishibuchi (Eds.), Hybrid Evolutionary Algorithms, Studies in Computational Intelligence, vol. 75, Springer, Berlin/Heidelberg, 2007, pp. 269–311.
[25] H. Liu, A. Abraham, A.E. Hassanien, Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm, Future Generation Computer Systems, in Press, Corrected Proof , doi:10.1016/j.future.2009.05.022.
[26] J. Carretero, F. Xhafa, Using genetic algorithms for scheduling jobs in large scale grid applications, Journal of Technological and Economic Development 12 (2006) 11–17.
[27] C. Li, L. Li, Utility-based qos optimisation strategy for multi-criteria scheduling on the grid, 10.1016/j.jpdc.2006.09.003Journal of Parallel and Distributed Computing 67 (2) (2007) 142–153.
[28] H. Izakian, B. Tork Ladani, K. Zamanifar, A. Abraham, A novel particle swarm optimization approach for grid job scheduling, in: S.K. Prasad, S. Routray, R. Khurana, S. Sahni (Eds.), Information Systems, Technology and Management, Communications in Computer and Information Science, vol. 31, Springer, Berlin, Heidelberg, 2009, pp. 100–109.
[29] I. Alberto, C. Azcarate, F. Mallor, P. Mateo, Multiobjective evolutionary algorithms. Pareto ranking, Monografias del Seminario Matematico Garcia de Galdeano 27 (2003) 25–35.
[30] N. Sample, P. Keyani, G. Wiederhold, Scheduling under uncertainty: planning for the ubiquitous grid, Proceedings of the 5th International Conference on Coordination Models and Languages, Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 300–316.
[31] A.A. Grosan, C., B. Helvik, Multiobjective evolutionary algorithms for scheduling jobs on computational grids, in: Proceedings of the IADIS International Conference on Applied Computing 2007, Salamanca, Spain, 2007, pp. 459–463.
[32] S. Benedict, V. Vasudevan , Scheduling of scientific workflows using niched Pareto ga for grids, 2006, pp. 908–912. doi:10.1109/SOLI.2006.329031.
[33] A.K.M.K.A. Talukder, M. Kirley, R. Buyya, Multiobjective differential evolution for workflow execution on grids, in: MGC'07: Proceedings of the 5th International Workshop on Middleware for Grid Computing, ACM, New York, NY, USA, 2007, pp. 1–6. <http://doi.acm.org/10.1145/1376849.1376852>.
[34] C. Franke , J. Lepping, U. Schwiegelshohn, On advantages of scheduling using genetic fuzzy systems, JSSPP'06: Proceedings of the 12th International Conference on Job Scheduling Strategies for Parallel Processing, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 68–93.
[35] K. Christodoulopoulos, V. Sourlas, I. Mpakolas, E. Varvarigos, A comparison of centralized and distributed meta-scheduling architectures for computation and communication tasks in grid networks, 10.1016/j.comcom.2009.03.004Computer Communications 32 (7–10) (2009) 1172–1184.
[36] D. Klusáček, H. Rudová, The importance of complete data sets for job scheduling simulations, Proceedings of the 15th International Conference on Job Scheduling Strategies for Parallel Processing, JSSPP'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 132–153.
[37] http://www.openpbs.org.
[38] http://www.cs.wisc.edu/condor.
[39] R. Wolski, N.T. Spring, J. Hayes, The network weather service: a distributed resource performance forecasting service for metacomputing, Future Generation Computer Systems 15 (5–6) (1999) 757–768.
[40] F. Sacerdoti, M. Katz, M. Massie, D. Culler, Wide area cluster monitoring with ganglia, 2003, pp. 289–298.
[41] S. Fitzgerald, Grid information services for distributed resource sharing, HPDC'01: Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing, IEEE Computer Society, Washington, DC, USA, 2001, pp. 181.
[42] D. Thain, T. Tannenbaum, M. Livny, Distributed computing in practice: the Condor experience, Concurrency and Computation Practice and Experience 17 (2–4) (2005) 323–356.
[43] S. Venugopal, R. Buyya, L. Winton, A grid service broker for scheduling distributed data-oriented applications on global grids, in: Proceedings of the 2nd Workshop on Middleware for Grid Computing, ACM, New York, NY, USA, 2004, pp. 75–80.
[44] D. Klusacek, Dealing with uncertainties in grids through the event-based scheduling approach, in: Fourth Doctoral Workshop on Mathematical and Engineering Methods in Computer Science (MEMICS 2008), vol. 1, Ing. Zdeněk Novotnỳ CSc., Ondráčkova, Brno, Further information, 2008, pp. 978–80.
[45] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases, World Scientific Pub. Co. Inc., 2001.
[46] M. Antonelli, P. Ducange, B. Lazzerini, F. Marcelloni, Learning concurrently partition granularities and rule bases of mamdani fuzzy systems in a multi-objective evolutionary framework, International Journal of Approximate Reasoning 50 (7) (2009) 1066–1080. (special Section on Graphical Models and Information Retrieval).
[47] R.P. Prado, S.G. Galán, J.E.M. Expósito, A.J. Yuste, S. Bruque, Genetic fuzzy rule-based meta-scheduler for grid computing, in: Fourth International Workshop on Genetic and Evolutionary Fuzzy Systems (GEFS 2010), 2010, pp. 51–56.
[48] D. Klusáček, L. Matyska, H. Rudová, Alea: grid scheduling simulation environment, Proceedings of the 7th International Conference on Parallel Processing and Applied Mathematics, PPAM'07, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 1029–1038.
[49] P.R. López, F. Jurado, N.R. Reyes, S.G. Galán, M. Gómez, Particle swarm optimization for biomass-fuelled systems with technical constraints, 10.1016/j.engappai.2008.04.013Engineering Applications of Artificial Intelligence 21 (8) (2008) 1389–1396.
[50] J. Robinson, Y. Rahmat-Samii, Particle swarm optimization in electromagnetics, IEEE Transactions on Antennas and Propagation 52 (2) (2004) 397–407.
[51] S.F. Smith, A learning system based on genetic adaptive algorithms, Ph.D. Thesis, Pittsburgh, PA, USA, 1980.

[52] L.B. Booker, D.E. Goldberg, J.H. Holland, Classifier systems and genetic algorithms, 10.1016/0004-3702(89)90050-7Artificial Intelligence 40 (1–3) (1989) 235–282.
[53] Y. Shi, R. Eberhart, Fuzzy adaptive particle swarm optimization, in: Proceedings of the 2001 Congress on Evolutionary Computation 2001, 2001.
[54] H. Liu, A. Abraham, Fuzzy adaptive turbulent particle swarm optimization, Proceedings of the Fifth International Conference on Hybrid Intelligent Systems, IEEE Computer Society, Washington, DC, USA, 2005, pp. 445–450.
[55] H. Liu, A. Abraham, A.E. Hassanien, Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm, Future Generation Computer Systems in Press, Corrected Proof , doi:10.1016/j.future.2009.05.022.
[56] D. Klusáček, H. Rudová, Alea 2: job scheduling simulator, in: Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, SIMUTools'10, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, 2010, pp. 61:1–61:10.
[57] Z. Šustr, J. Sitera, M. Mulač, M. Ruda, D. Antoš, L. Hejtmánek, P. Holub, Z. Salvet, L. Matyska, MetaCentrum, the Czech Virtualized NGI.
[58] C.N.G. Infrastructure, <http://www.fi.muni.cz/xklusac/index.php?page=meta2009>. Metacentrumdatasets, <http://www.fi.muni.cz/xklusac/index.php?page=meta2009>.
[59] R. Prado, S. García-Galán, A. Yuste, J. Expósito, Genetic fuzzy rule-based scheduling system for grid computing in virtual organizations, Soft Computing – A Fusion of Foundations, Methodologies and Applications 15 (2011) 1255–1271.
[60] W. Venables, D. Smith, R development core team (2010) an introduction to r. version 2-11-1, 2010, pp. 1–100. <http://www.r-project.org/>.
[61] J. Alcalá-Fdez, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández, F. Herrera, Keel: a software tool to assess evolutionary algorithms for data mining problems, Soft Computing 13 (3) (2009) 307–318.
[62] X. Tang, S.T. Chanson, Optimizing static job scheduling in a network of heterogeneous computers, Proceedings of the 2000 International Conference on Parallel Processing, ICPP'00, IEEE Computer Society, Washington, DC, USA, 2000, pp. 373–382.
[63] C. Ernemann, V. Hamscher, R. Yahyapour, Benefits of global grid computing for job scheduling, in: Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, GRID'04, IEEE Computer Society, Washington, DC, USA, 2004, pp. 374–379.
[64] M. Hovestadt, O. Kao, A. Keller, A. Streit, Scheduling in hpc resource management systems: Queuing vs. planning, in: Proceedings of the 9th Workshop on Job Scheduling Strategies for Parallel Processing, Springer, 2003, pp. 1–20.