A Combined Fuzzy-Neural Network Model for Non-Linear Prediction of 3-D Rendering Workload in Grid Computing

Nikolaos D. Doulamis, *Member, IEEE*, Anastasios D. Doulamis, *Member, IEEE*, Athanasios Panagakis, Konstantinos Dolkas, Theodora A. Varvarigou, *Member, IEEE*, and Emmanuel Varvarigos

***Abstract*—Implementation of a commercial application to a grid infrastructure introduces new challenges in managing the quality-of-service (QoS) requirements, most stem from the fact that negotiation on QoS between the user and the service provider should strictly be satisfied. An interesting commercial application with a wide impact on a variety of fields, which can benefit from the computational grid technologies, is three–dimensional (3-D) rendering. In order to implement, however, 3-D rendering to a grid infrastructure, we should develop appropriate scheduling and resource allocation mechanisms so that the negotiated (QoS) requirements are met. Efficient scheduling schemes require modeling and prediction of rendering workload. In this paper workload prediction is addressed based on a combined fuzzy classification and neural network model. Initially, appropriate descriptors are extracted to represent the synthetic world. The descriptors are obtained by parsing RIB formatted files, which provides a general structure for describing computer-generated images. Fuzzy classification is used for organizing rendering descriptor so that a reliable representation is accomplished which increases the prediction accuracy. Neural network performs workload prediction by modeling the nonlinear input-output relationship between rendering descriptors and the respective computational complexity. To increase prediction accuracy, a constructive algorithm is adopted in this paper to train the neural network so that network weights and size are simultaneously estimated. Then, a grid scheduler scheme is proposed to estimate the queuing order that the tasks should be executed and the most appopriate processor assignment so that the demanded QoS are satisfied as much as possible. A fair scheduling policy is considered as the most appropriate. Experimental results on a real grid infrastructure are presented to illustrate the efficiency of the proposed workload prediction — scheduling algorithm compared to other approaches presented in the literature.**

***Index Terms*—Grid computing, workload prediction, neural networks, three-dimensional (3-D) rendering.**

Manuscript received December 5, 2002; revised November 21, 2003. This work was supported by the European Union under the program of Information Societies Technology (IST), No. IST-2001-33240, Grid Resources for Industrial Applications (GRIA). This paper was recommended by Associate Editor A. G. Skarmeta.

N. D. Doulamis, A. D. Doulamis, A. Panagakis, K. Dolkas, and T. A. Varvarigou are with the National Technical University of Athens (NTUA), Department of Electrical and Computer Engineering, Athens, Greece (e-mail: ndoulam@cs.ntua.gr; adoulam@cs.ntua.gr).

E. Varvarigos is with the Department of Computer Engineering and Informatics, University of Patras, 26500 Patras, Greece, (e-mail: manos@ceid.upatras.gr).

Digital Object Identifier 10.1109/TSMCB.2003.822282

I. Introduction

SEVERAL EMERGING network applications in the areas of high performance computing or information analysis cannot be satisfied by the quality-of-service (QoS) requirements associated with relatively low-bandwidth flows, such as the Internet. Examples include collaborative visualization of large datasets or computationally demanding data analyzes, which usually require data streaming at hundreds or even thousands of megabits p/s [1]. For this reason, new abstractions and concepts should be introduced at both the architecture and network level to allow applications to access and share resources or services among distributed networks [1]. All these issues are addressed by using a transparently, integrated, distributing computing infrastructure, referred as grid, which support the sharing, interconnection and use of diverse resources in dynamic computing systems that can sufficiently be integrated to deliver the desired QoS [2]. Although computational grid has been initially developed to solve large-scale scientific research problems, it is expected to be applied for several high computational load demanded commercial applications. Implementing, however, a commercial application to a grid infrastructure introduces new challenges in managing the QoS requirements [1]. An interesting commercial application with a wide impact in many fields, is three-dimensional (3-D) image rendering [3] which, however, demands high processing power. For this reason, 3-D rendering can be solved more feasibly in a grid infrastructure. This is the fundamental business objective of the European Grid Resources for Industrial Applications (GRIA) project to create and evolve a grid testbed and apply it to two distinctive commercial application areas, 3-D rendering and dynamic structural analysis [4].

However, to deliver 3-D rendering tasks of very high values of services, advance resource allocation and scheduling mechanisms should be incorporated. Scheduling is an important issue in grid capability of delivering commercial applications, such as the 3-D rendering, since it provides a convenient way to access the end-to end QoS requirements. This need has been confirmed by the global grid forum in the special working group dealing with the area of scheduling and resource management [5]. To efficiently, however, implement a scheduling and resource allocation management algorithm, modeling of QoS is required and prediction of the associated parameters.

Modeling and prediction

1) provide to the users the ability to estimate the service level needed for their application before or during the negotiation phase of the grid submission process;
2) manage the assignment of application loads to resources to guarantee delivery of these service levels to the required standard;
3) implement a recovery model if either the estimation or delivery of service proves inadequate [6].

Modeling and prediction of QoS parameters is application-dependent since the inherent parameters of a problem, which affect the final outcome, should be identified [7]. For this reason, modeling should be performed to a specific class of applications, such as the 3-D image rendering, which is one important commercial application due to its wide impact to a variety of fields. Workload prediction and modeling of ray-tracing algorithms has been reported in [8]. In this work, it is confirmed that the time complexity is less dependent on the number of objects, but more on the object size. In [9], a modification of the previous approach has been adopted to avoid double intersection tests for objects that cross voxel boundaries. In addition, a scaling factor has been added to account for an early ray termination due to intersection with another objects. The work of [10] estimates the average probability for a ray to be intersected with an object in a cell, accomplished by the projected area of the box enclosing the objects in a cell, while an algorithm to estimate the cost of ray tracing in a scene is presented in [11]. The method assumes an octtree spatial subdivision and the cost per voxel is predicted. The cost of ray tracing using adaptive spatial subdivisions has been studied in [12], by analyzing the probability that a ray intersects an object. Other approaches improve the efficiency of radiosity and Monte Carlo irradiance rendering algorithms using parallel/ hierarchical methods and multiple important sampling respectively [13], [14].

These approaches, however, are based on specific rendering algorithm characteristics and cannot be extended to other rendering schemes or modifications of the applied ones. In addition, they do not exploit the complexity of the synthetic scene. To overcome these difficulties in this paper, several descriptors are extracted to represent the scene complexity. Then, a model is adopted to map the synthetic scene complexity with the workload characteristics. Linear models cannot efficiently predict the workload of 3-D rendering algorithms, since there is no a simple linear relationship, which associates the rendering parameters to the respective computational cost [3]. Alternatively, prediction can be preformed using nonlinear simplified mathematical models such as functions of exponential or logarithmic type and then estimate the model parameters to fit the data. However, these approaches present satisfactory results only in case of data that follow a predetermined function type, which is not the case of 3-D rendering algorithms. For this reason, a neural network architecture is adopted in this paper to perform the nonlinear mapping of the extracting descriptors to rendering workload since it can be proved that a neural network model can approximate any nonlinear function within any degree of accuracy [15]. For this reason, neural networks have been extensively used for modeling highly nonlinear complex problems, such as advanced resource allocation mechanisms [16], video traffic prediction [17], and nonlinear dynamic systems [18]. However, the prediction accuracy of a neural network architecture depends on

1) organization of the extracted descriptors;
2) network structure and size;
3) training algorithm adopted.

Usually, the extracted descriptors are organized into classes using a binary classification scheme, i.e., each descriptor is allowed to belong only to one class. In a binary classification, however, it is possible for two descriptors to assign into different classes if they are located on opposite sides of a class boundary. This is, for example, the case of noise in the descriptors. An alternative descriptor organization is to permit each descriptor to belong to several (or even to all) classes but with a different degree of membership. One way to estimate the membership grade is to use probability theory by exploiting the descriptor statistics [19]. Another way refers to fuzzy classification, which models the possibility of an event, i.e., to which extent an event can occur [20]. Fuzzy classification provides a more meaningful representation of the extracted descriptors, closer to the human perception. Furthermore, fuzzy classes are not restricted by the additivity property as the probabilities (they must add together to one). Fuzzy classification of the obtained descriptors has been applied in many applications, such as visual content retrieval [21]. In addition, combination of neural networks with fuzzy classification increases the prediction accuracy. More specifically, in [22] a neural network-based fuzzy model is adopted for predicting transient stability in power systems, while [23] a combined neural network fuzzy model is used for time series prediction.

Furthermore, the network size affects its performance [24], [25] and [26]. Particularly, a small network is not able to approximate complicated nonlinear functions [24], [25], while an unnecessarily large network overfits the data and thus it cannot generalize well [26]. For this reason, training algorithms, which simultaneously estimate network weights and size, are presented, such as constructive or pruning methods [15], [24], [25]. Usually, constructive approaches present a number of advantages over other methods used for network size selection. More specifically, in a constructive scheme, it is straightforward to estimate an initial size for the network. Furthermore, in case that many networks of different sizes provide acceptable solutions, the constructive approach yields the smallest possible size [24].

The rendering descriptors are obtained by parsing a RenderMan Interface Bytestream (RIB) formatted file, which provides a general structure for describing a synthetic world. RIB format includes information about the object geometric primitives (such as cylinder, cone and sphere), object transformation, object material and texture, number of light sources, rendering algorithm parameters and in general any detail used for creating the rendered images.

Two different types of descriptors are considered. The *general and the object descriptors*. General descriptors refer to the entire synthetic world, such as the image resolution, the number and type of light sources and general-purpose parameters of the rendering algorithm used. Instead, object descriptors concerns a specific synthetic geometry, such as the object geometrical complexity, surface texture and material used. As we have stated above, the extracted descriptors are organized in a fuzzy classification and then fed as input to a feedforward neural network architecture, to predict the rendering workload. A constructive

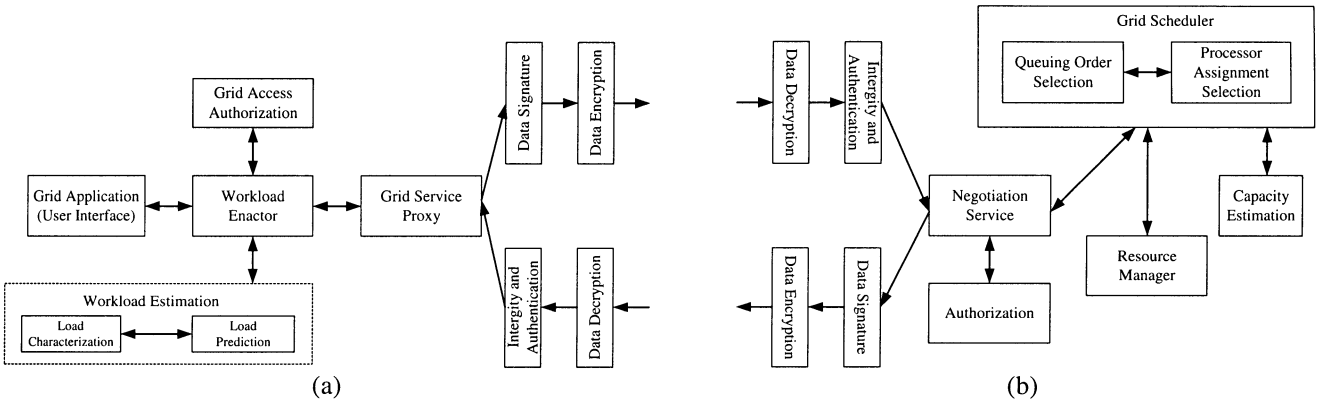

Fig. 1. Adopted grid infrastructure. (a) Client side. (b) Server side.

algorithm is adopted in our case to train the network, which optimally estimates a) the most appropriate network size, i.e., number of neurons and b) the respective network weights [24]. This method belong to the category of constructive algorithms that only the weights of the new added neuron are estimated, yielding to low computational load and storage requirements compared to other techniques. Furthermore, the constructed network size is independent from the size of the training samples as happens with other approaches [27].

This paper is organized as follows. In Section II, the proposed grid infrastructure used in the experiments is described. In Section III, the basic parameters affecting the rendering performance is analyzed along with the RIB format used to organize the extracted descriptors. Section IV refers to the workload estimation, including the fuzzy organization of the extracted descriptors and the neural network based modeling. The adopted fair scheduling algorithm is analyzed in Section V. Finally, experimental results and comparisons with other models are presented in Section VI, while Section VII concludes the paper.

II. Grid Infrastructure

Fig. 1 presents a block diagram of the adopted infrastructure used to apply 3-D rendering algorithms in grid computing. As can be seen, the architecture is discriminated into two main parts; the *client side* architecture [see Fig. 1(a)] and the *server side* architecture [see Fig. 1(b)]. This grid infrastructure has been implemented in the framework of GRIA and a Grid Application Toolkit and Testbed (GridLAB) European Union funded projects. The main module of the client side is the load characterization/prediction. On the other hand, the grid scheduler constitutes the heart of the server side architecture. These two modules collaborate with each other and enhance the capability of the grid infrastructure in delivering commercial applications in a way that satisfies the negotiated QoS requirements. On the contrary, in the current grid architecture the assigned tasks are served using a first come, first serve policy.

The main parts of the adopted grid architecture at the client side are summarized as follows.

A. Grid Application

This module provides an interface required for interacting the user with the grid infrastructure. The interface is designed to control a collection of grid services for the user desktop, i.e., the deadlines of the submitted tasks, the task priorities and so on.

B. Workflow Enactor

This is an intermediate module with interacts with all modules at the client side. The workflow enactor is responsible for activating each time an appropriate module at the client side.

C. Load Characterization/Prediction

This module is responsible for modeling and predicting the task workload characteristics. This information is then provided to the architecture of the server side along with the associated task deadlines so that an appropriate scheduling scheme is accomplished.

D. Grid Access Authorization

The authorization module checks whether the user is authorized to access the grid resources and on which terms.

E. Grid Service Proxy

This module instantiated by the workflow enactor to handle invocation of remote grid servers, either in the application or in the negotiation steps.

On the contrary, the main parts of the grid infrastructure at the server side are the following.

F. Grid Scheduler

The scheduler is the heart of the server architecture and determines *when* and *at which* processor the submitted tasks should be executed so that the demanded QoS parameters are satisfied as much as possible. The scheduler uses information obtained by the load characterization/ prediction module and the current resource availability.

G. Negation Service

In case that the demanded QoS parameters of the submitted tasks can not be satisfied (i.e., the task deadlines are violated), the negotiation service is activated to inform the users for the violation and ask them whether they are willing to submit the task with the supported by the grid infrastructure QoS parameters.

H. Resource Manager

This module is responsible for sending the submitted tasks for execution in the grid clusters.

As can be seen from the above mentioned architecture, the load characterization/prediction and the grid scheduler modules perform probably the most significant jobs in the successful delivering of tasks. For this reason, these two modules are described in more detail in the following.

III. 3-D Rendering Descriptors

As we have stated in Section I, to predict the 3-D rendering workload, several descriptors are extracted, which characterize the complexity of the synthetic world. For describing the synthetic world, however, a RIB encoded file is used. Thus, descriptor estimation is accomplished by parsing a RIB encoded file.

A. Encoding and Estimation of Rendering Descriptors

The purpose of the RIB format is to provide a general structure for describing a synthetic world. Particularly, it offers the possibility of reconstructing any geometric primitive, such as a cone, a sphere, a disk, and so on and it allows the performance of several transformations on each primitive. Thus, any complicated 3-D object is constructed by an appropriate combination of geometric primitives and transformations. RIB format also encodes additional useful information for 3-D rendering, such as object surface characteristics, the number, intensity, location and type of light sources, image resolution and so on. Furthermore, RIB also describes the rendering algorithm used along with the values of the associated parameters.

An example of a RIB file is presented in Table I of a synthetic world that consists of a cylinder. The cylinder surface is "shiny," characterized by diffuse reflection of 0.2 [3]. The statement "WorldBegin" defines the "begin" of the synthetic world, while the statement "WorldEnd" the "end" of it. In this example, the ray tracing algorithm has been used for 3-D rendering with maximum level of tree rays equal to four (4) as indicated by the command line "option "render" integer max raylevel [4]." Perspective projection is adopted, while the image resolution is of 200×150 pixels as results from "format" statement.

B. Three-Dimensional Rendering Descriptors

The extracted descriptors used to predict the 3-D rendering workload can, in general, be classified into two main categories. The first type of descriptors refers to *general* characteristics of the synthetic scene, such as the image resolution, the number and type of light sources and general-purpose parameters related to the rendering algorithm used. We call these descriptors *general descriptors*. The second type concerns descriptors related to a specific synthetic geometry and primitive characteristics, such as the object complexity, surface texture and material used. We call these descriptors *object descriptors*.

RIB format provides a convenient way for discriminating general and object descriptors. Particularly, most of rendering descriptors are encoded using the statements "option" and "attribute." The command line "option" applies to the entire scene and thus encodes general descriptors. Instead, the command line "attribute" applies to a specific geometry corresponding to object descriptors.

TABLE I hskip3pt
EXAMPLE OF A RIB FILE FORMAT

```
Projection "perspective" "fov" 40
Format 200 150 1

Option "render" "integer max raylevel" [4]

WorldBegin

    Cylinder 1 -1 1 360
    Surface "shiny" "Kd" [0.2]

WorldEnd
```

Another important issue, which affects the rendering workload, is the algorithm used to render a synthetic scene. It is clear that different types of algorithms affect the rendering workload in a different way. In this paper, we are dealing with the ray tracing, radiosity and Monte Carlo irradiance analysis algorithms since they are the most commonly used 3-D rendering schemes. For each type of algorithm, different descriptors are extracted and then used for the workload prediction. All these parameters are supported by the RIB encoded file.

Tables II–IV present the correspondence between the main descriptors of the three investigated algorithms and the respective encoding of the RIB format. There are also some other descriptors, which affect the rendering computational complexity but are not coded with the statements "option" and "attribute." Such descriptors include the type of object material and the associated illumination model, the object geometrical complexity and the image resolution. The encoding of these descriptors is shown in Table V.

IV. Workload Estimation

A. Fuzzy Organization

In contrast to general descriptors, object descriptors cannot directly be included in a feature vector, since the number of objects is not constant and varies from scene to scene. This would result in feature vectors of different size, making direct comparisons between different scenes practically impossible. A simple way to overcome this difficulty is to classify each object descriptor into predetermined classes by constructing histograms.

In a binary, however, classification scheme, it is probable for two similar descriptors to assign to different bins (classes) if they are located on opposite sides of a class boundary. In this way, two descriptors are treated either identical or different. To overcome the aforementioned difficulty, an alternative framework should be used, which allows for each descriptor to belong to several (or even all classes) but with a different degree of membership. One way to estimate the membership degree is based on an *a posteriori* probability classification scheme, by exploiting the descriptor statistics [19]. Another method is to apply fuzzy classification to the extracted descriptors [20]. While, probability expresses the likelihood of an outcome, fuzziness refers to the possibility of an event, i.e., models to which *extent* an event can occur. A minimum requirement of probabilities is *additivity* property that is the probabilities must

TABLE II
DESCRIPTORS OF THE RAY TRACING RENDERING ALGORITHM AND THE RESPECTIVE ENCODING IN THE RIB FORMAT. THE "[ ]" INDICATES THE RESPECTIVE DEFAULT VALUES

Ray Tracing Descriptors	
Descriptor	**RIB Statement**
Maximum number of recursive rays	Option "render" "integer max raylevel" [4]
Minimum Shadow distance	Option "render" "float minshadowbias" [0.01]
Surface shadow property	Attribute "render" "string casts shadows" ["Os"]

TABLE III
MAIN DESCRIPTORS OF THE RADIOSITY RENDERING ALGORITHM AND THE RESPECTIVE ENCODING IN THE RIB FORMAT. THE "[ ]" INDICATES THE RESPECTIVE DEFAULT VALUES

Radiosity Descriptors	
Descriptor	**RIB Statement**
Radiosity steps	Option "radiosity" "integer steps" [0]
Minimum number of samples per patch	Option "radiosity" "integer minpatchsamples" [1]
Surface type energy	Attribute "radiosity" "string zonal" ["fully zonal"]
Patch parameterization	Attribute "radiosity" "float patchsize" [4] Attribute "radiosity" "float elemsize" [2] Attribute "radiosity" "float minsize" [1]

TABLE IV
MAIN DESCRIPTORS OF THE MONTE CARLO RENDERING ALGORITHM AND THE RESPECTIVE ENCODING IN THE RIB FORMAT. THE "[ ]" INDICATES THE RESPECTIVE DEFAULT VALUES

Monte Carlo Irradiance Descriptors	
Descriptor	**RIB Statement**
Maximum error metric	Attribute "indirect" "float maxerror" [0.25]
Maximum pixel distance	Attribute "indirect" "float maxpixeldist" [20]
Number of samples cast to compute the irradiance	Attribute "indirect" "integer nsamples" [256]

add together to one. However, this does not hold with fuzzy membership degrees. In addition, fuzzy classification provides a more meaningful representation of the extracted descriptors, which is closer to the human perception while it is independent from the descriptor statistics.

Let us denote as $\mathbf{d}_i$, with elements $d_{i,j}$ the descriptors used for the ith object. Thus,

$$\mathbf{d}_i = [d_{i,1} d_{i,2} \cdots d_{i,L}]^T \tag{1}$$

where L is the size of vector $\mathbf{d}_i$. We then assume that each element $d_{i,j}$ is classified into Q classes (partitions) by means of Q membership functions. Let us denote as $n_j \in \{0, \ldots, Q-1\}$ the partition to which the jth element of $\mathbf{d}_i$, i.e., $d_{i,j}$, belongs. The degree of membership of $d_{i,j}$ to the partition n_j is then estimated by the membership function $\mu_{n_j}(d_{i,j})$. We further assume that all elements $d_{i,j}$ are normalized in the interval [0 1]. Variable n_j refers to the bin of the jth element of vector $\mathbf{d}_i$.

The exact type and shape of membership functions $\mu_{n_j}(d_{i,j})$ can be greatly varied and in general depends on the specific problem [20]. Some interesting types are the triangular with 50% overlap, the quadratic and the cubic ones, which are presented in Fig. 2. In all the above cases, "symmetric" functions have been used since there is no reason to give more importance to a specific class. The actual type of membership functions and the number of partitions Q are estimated to maximize the prediction accuracy as explained in the section of the experimental results.

Gathering all bins n_j for $j = 1, 2, \ldots, L$, a multidimensional class $\mathbf{n}$ is constructed as $\mathbf{n} = [n_1 n_2 \cdots n_L]^T$, which indicates the bin (class) to which vector $\mathbf{d}_i$ is classified.

Taking into consideration the degree of membership $\mu_{n_j}(d_{i,j})$ of the element $d_{i,j}$ to the bin n_j, the degree of membership of vector $\mathbf{d}_i$ to a particular class $\mathbf{n}$ is estimated using the following

$$\mu_{\mathbf{n}}(\mathbf{d}_i) = \prod_{j=1}^{L} \mu_{n_j}(d_{i,j}). \tag{2}$$

Using (2), we can construct the histogram bin of a specific class $\mathbf{n}$, by taking into account the effect of all object descriptors, $\mathbf{d}_i$, $i = 1, 2, \ldots, K$ to the bin $\mathbf{n}$.

$$H(\mathbf{n}) = \frac{1}{K} \sum_{i=1}^{K} \mu_{\mathbf{n}}(\mathbf{d}_i) = \frac{1}{K} \sum_{i=1}^{K} \prod_{j=1}^{L} \mu_{n_j}(d_{i,j}). \tag{3}$$

Thus, the fuzzy histogram is created as

$$\mathbf{f}^o = [f_0 f_1 \cdots f_{Q^L-1}]^T \text{ with } f_{z(\mathbf{n})} = H(\mathbf{n}) \tag{4}$$

with $z(\mathbf{n}) = \sum_{j=1}^{L} n_j Q^{L-j}$.

B. Workload Prediction

Let us denote in the following as $\mathbf{f}$ the feature vector, which include all general and object-based descriptors. Feature vector $\mathbf{f}$ affect the computational load by a nonlinear relationship modeled as

$$y = g_c(\mathbf{f}), c \in \{\Pi_1, \ldots, \Pi_M\} \tag{5}$$

where y is the respective computational cost and $g_c(\cdot)$ the nonlinear relationship. Index c of $g_c(\cdot)$ corresponds to a particular rendering algorithm, denoted in (5) as Π_i. In our experiments, the rendering algorithms of the ray tracing, radiosity method and Monte Carlo irradiance analysis have been applied. The main difficulty of (5) is that the input-output relation $g_c(\cdot)$ is actually unknown. Modeling of $g_c(\cdot)$ is performed through a feed-forward neural network architecture, since it can approximate any nonlinear function within any degree of accuracy ([15, pp. 208–213, 249]). In our case, a neural network of one hidden layer and an output layer of one neuron has been adopted. Linear

TABLE V
OTHER DESCRIPTORS AND THE RESPECTIVE ENCODING IN THE RIB FORMAT. THE "[ ]" INDICATES THE RESPECTIVE DEFAULT VALUES

Other Factors	
Descriptor	**RIB Statement**
Image resolution (width-height- pixel aspect ratio)	Format 200 150 1
Displacement type	Attribute "render" "integer truedisplacement" [0]
Bounding box coordinates	Attribute "displacementbound" "string coordinatesystem" ["current"] "float sphere" [0]
Patch multiplier	Attribute "render" "float patch multiplier" [1.0]
Minimum / Maximum level of subdivision	Attribute "render" "float patch maxlevel" [256] Attribute "render" "float patch minlevel" [1]
Surface complexity	For example: PatchMesh "bicubic" 13 "nonperiodic" 10 "nonperiodic" "P"
Surface material type	Surface "shiny" "Kd" [.1] "Kr" [0.5] "Ka" [0.1]
Light source type	For example: LightSource "pointlight" 1 "from" [0 10 0]

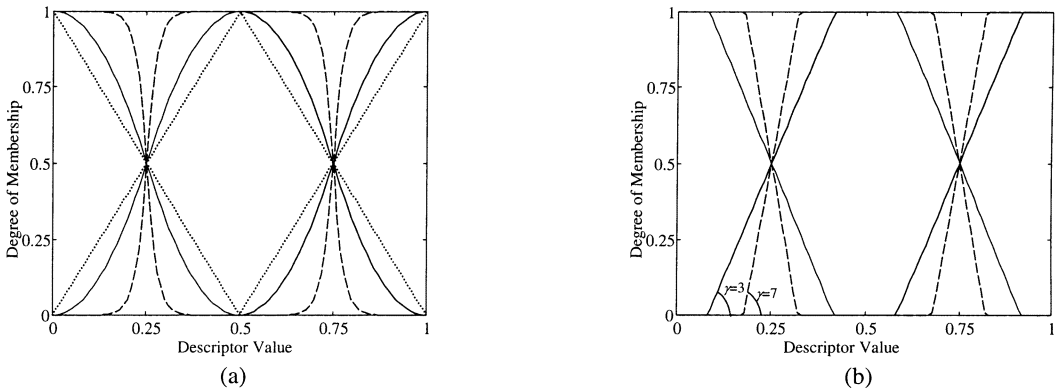

Fig. 2. Different types of membership functions. (a) Quadratic (solid line), the function of order ten (dashed line) and the triangular (dotted line). (b) Trapezoid membership functions with slope $\gamma = 3$ (or 71.5°) (solid line) and $\gamma = 7$ (or 81.9°) (dashed line).

activation function is used for the output neuron, since rendering workload can be any real value.

Constructive Neural Network Training: A constructive algorithm has been adopted in this paper to estimate network weights and size, since usually the constructive approaches present a number of advantages over other methods, such as pruning techniques, used for network size selection. This is due to the fact that in a constructive scheme, among many networks of different sizes that provide acceptable solutions, the smallest one is chosen [24].

Several constructive algorithms have been proposed in the literature for simultaneously estimating the network weights and size. In general, the constructive algorithms can be classified into three main categories. The first approaches train the whole network completely after its hidden neuron addition [28], [29]. However, these schemes yield a high computational load, which depends on the optimization algorithm used for training and the network size [27]. The second approach simplifies the optimization problem by assuming that the hidden units already existing in the network are useful in approximating part of the desired (target) function [24]. Thus, the weights feeding these hidden units can be considered fixed and allow only the weights connected to the new hidden unit to vary. As a result, a significant reduction of the number of weights that should be optimized is accomplished, yielding to a reduction of computational load and storage requirements. The third category concerns memorization methods [30], [31]. The main concept of these algorithms is to train the whole network only for the "easy" training patterns, while using memorization for the "hard" or "novel" patterns. However, these methods tend to produce networks that potentially grow linearly with respect to the size of the training set [27].

In this paper, the constructive algorithm proposed in [24] has been adopted to train the neural network. This method belongs to the second category of constructive algorithms, i.e., training only the new added neuron. This is due to the fact that it yields low computational load and storage requirements compared to other techniques. Furthermore, it yields a network of small size despite the complexity of the target function and possible noise in the training samples.

V. GRID SCHEDULER

The purpose of a scheduling algorithm is to determine the "queuing order" and the "processor assignment" for a given task so that the demanded QoS parameters, i.e., the task deadlines, are satisfied as much as possible. The "queuing order" refers to the order in which tasks are considered for assignment to the processors. The "processor assignment" refers to the selection of the particular processor on which the task should be scheduled.

A. Queuing Order Selection

In the proposed grid architecture, two approaches for queuing order selection have been adopted, which are described in the following. The first algorithm exploits the urgency of the task deadlines, while the second is based on a fair policy.

1) Urgency-Based Algorithms: The most widely used urgency-based scheduling scheme is the earliest deadline first (EDF) method, also known as the deadline driven rule [32]. This method dictates that at any point the system must assign the highest priority to the task with the most imminent deadline. The concept behind the EDF scheme is that it is preferable to serve first the most urgent tasks (i.e., the task with the earliest deadline) and then serve the remaining tasks according to their urgency.

2) Fair Completion Time: The above mentioned queuing order selection algorithm does not make any attempt to handle the tasks requesting for service in a fair way. For example, tasks with relative urgency may be favored against the remaining tasks, regardless of the respective workload. In addition, using an EDF scheduling scheme, there is no motivation for a user to specify an honest deadline, since tasks of late deadlines are given low priority. To overcome the aforementioned difficulties, an alternative approach is presented in this section, by handling tasks requesting for service with respect to their *fair completion times*.

The fair completion time of a task is found by first estimating its *fair task rates* using a Max-Min fair sharing algorithm [33]. For this reason, initially, the task demanded rates are estimated as follows

$$X_i = \frac{w_i}{D_i - \delta_i} \tag{6}$$

where w_i refers to the workload of the ith task requested for serving, estimated by the load characterization/prediction module. D_i corresponds to the task deadline, while the δ_i represents the *grid access delay*, and it can be viewed as mean delay required for the ith task to access the total grid capacity. Thus, δ_i includes the communication delays and delays stem from grid resource availability to execute the respective task [34].

In case of congestion, the computational rate allocated to a task may be smaller than the demanded rate X_i, and thus violation of the task deadline is accomplished. The fair scheduling algorithm attempts to degrade the QoS experienced by the tasks (as measured by the computational rate allocated to the task, or the amount of time by which the deadline is missed as a percentage of $D_i - \delta_i$) in a *fair way*.

Particularly, based on the task demanded rates X_i, the task fair rates r_i are calculated using the Max–Min fair sharing algorithm [33]. In the max-min fair sharing, all users are given an equal share of the total resources, unless some of them do not need their whole share, in which case unused resources are divided equally among the remaining "bigger" users in a recursive way. Using the task fair rates, the fair completion times of the tasks are estimated as follows [34]

$$t_i = \delta_i + \frac{w_i}{r_i} \tag{7}$$

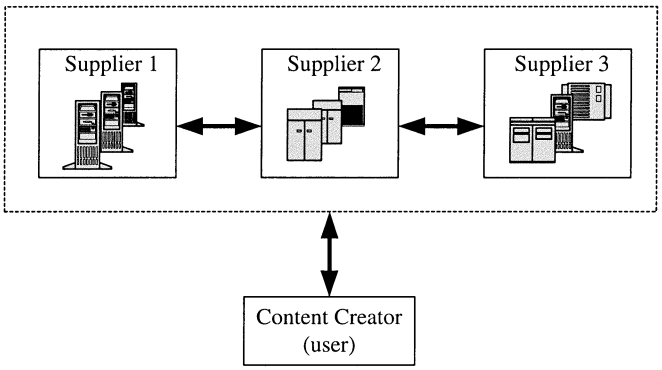

Fig. 3. Implemented grid topology.

where t_i can be thought of as the time at which the task would be completed if it could obtain constant computational rate equal to its fair computational rate r_i starting at time δ_i. It is should be mentioned that finishing all tasks at their fair completion times is unrealistic, because grid is not a single computer that can be accessed at any desired computational rate and uniform delay δ_i. However, the fair completion times are used to perform the queuing order of the task in a fair way. In particular, the tasks are ordered for execution with respect to their earliest fair completion times. For this reason, we call this scheme fair completion time (FCT) queuing order selection in the following.

B. Processor Assignment

Although the above mentioned algorithms select an appropriate order for the task execution, they dot not solve the problem of "processor assignment," i.e., at which particular processor the tasks should be executed. In our case, the earliest completion time (ECT) scheme is proposed as a solution to the "processor assignment" problem.

In this approach, we assume that each task occupies 100% of the processor utilization for its execution, and thus the maximum utilization of a processor is assigned if the tasks are served in the earliest possible time. Then, the ECT rule selects among all the available processors of the grid infrastructure, the one which provides the earliest completion time for the task execution [34].

VI. Experimental Results

A real grid computing infrastructure has been implemented to perform the simulations. The general topology has been shown in Fig. 3 and has been supported under the European research funded project GRIA. In the adopted infrastructure three main suppliers to grid resources are involved, each of which consists of several clusters of different platforms. The 3-D rendering tasks, which are submitted to the grid infrastructure for processing, are provided by a content creator company involved in the project. The grid clusters are distributed in different European countries. In addition, the software of the infrastructure has been developed in a Java platform, while the server side architecture operates in a Linux operation program.

For better clarification of the adopted architecture, let us consider an example, where the content creator (i.e., a user) wishes

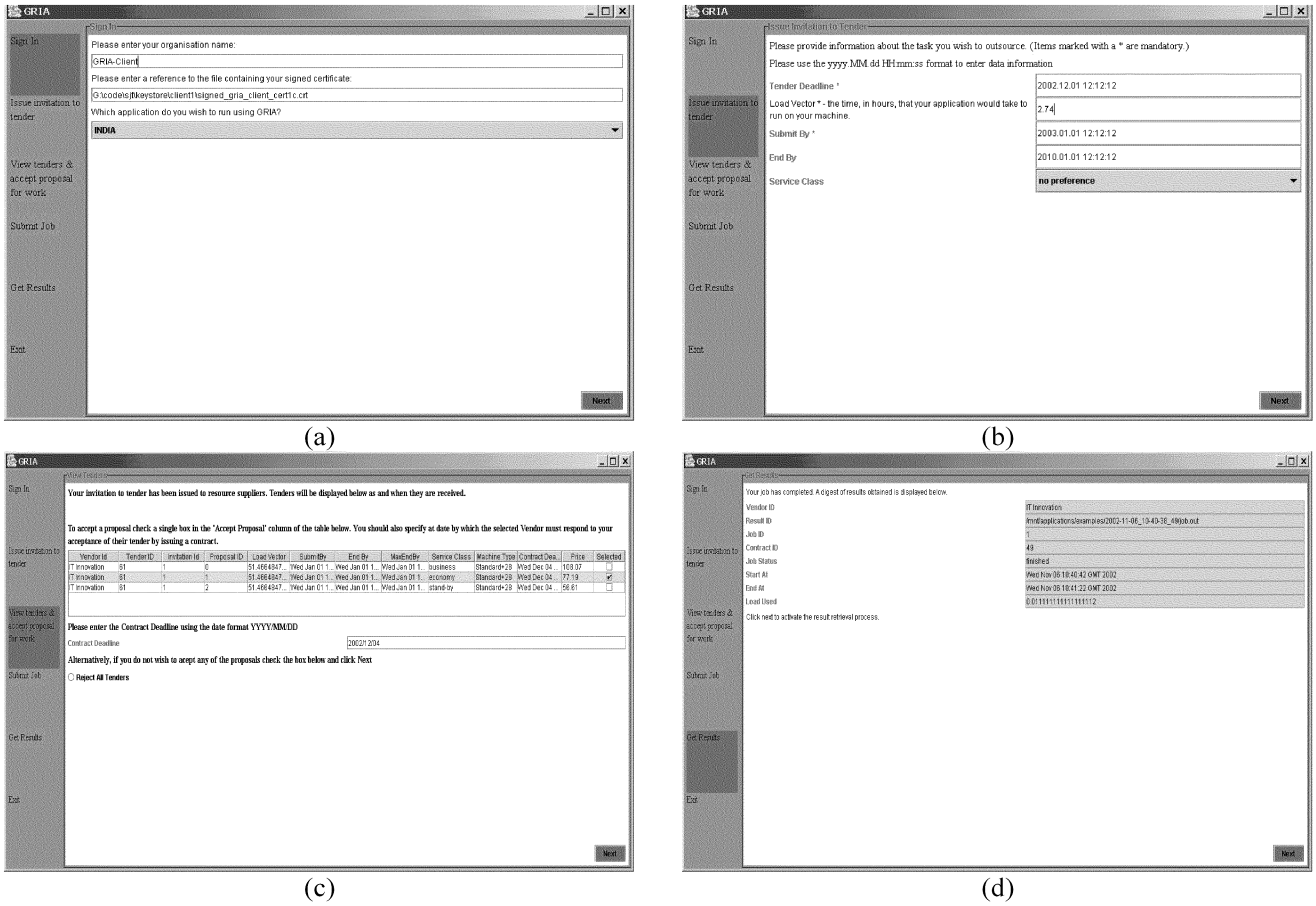

Fig. 4. Examples of the graphical user interface (GUI) used in the experiments in the adopted grid infrastructure. (a) Sign in GUI. (b) The task parameters GUI. (c) Negotiation service GUI. (d) Job finished GUI.

to submit 3-D rendering tasks for processing. Based on the architecture depicted in Fig. 1 (see Section II), initially, the application service module is activated to provide a friendly interface so that the user is able to handle all the necessary information of the tasks, e.g., to define the task deadlines. In the following, the workflow enactor is activated to enable the authorization module and then the workload estimation. All the derived information, along with the QoS parameters of the respective tasks as indicated by the users, are provided to the server side for task execution. Encryption/decryption schemes are involved in this phase to protect the privacy of the transferred data.

Fig. 4 present an example of the graphical user interface (GUI) implemented in the framework of the GRIA project. In particular, Fig. 4(a) presents the initial GUI, where the type of application are selected (i.e., rendering), while Fig. 4(b) the GUI where the task information are defined (i.e., task deadline or the task workload as predicted by the load characterization/prediction module).

At the server side, following the authorization module, the scheduler is activated to define when and on which processor the submitted tasks should be executed. In case that violation of the requested QoS (i.e., deadline) is obtained, the negotiation service module is activated to inform the users and, if possible, to adapt the requested QoS parameters to those that can be supported by the grid architecture. The final task execution is provided by the resource manager, which, based on the information derived from the scheduler, submits the tasks for execution. Fig. 4(c) presents the GUI of "job running" and "job finished," respectively, while Fig. 4(d) the GUI for delivering the results.

A. Workload Prediction

To train the neural network, used for workload prediction, a measurement set is constructed consisting of pairs of rendering descriptors along with the respective computational complexity. In our experiments, the computational cost has been normalized to the CPU speed and the platform that the task is assigned to be executed to increase the prediction accuracy. Normalization is performed with respect to a reference synthetic scene implemented on a particular platform. Three measurement sets are constructed in our case, each of which corresponds to one of the three investigated rendering algorithms. Then, the measurement sets are randomly partitioned into three disjoint sets; the training set, the validation set and the test set. The training set is used to estimate network weights and size. The validation set is responsible to terminate network training by estimating the early stopping point [16], while the test set to evaluate the network performance. The 60% of data for each measurement set comprise the training set, while the rest 40% is equally shared for the validation and test set. Each measurement set consists of 500 samples of randomly variations of the respective rendering descriptors and also includes several different synthetic scenes.

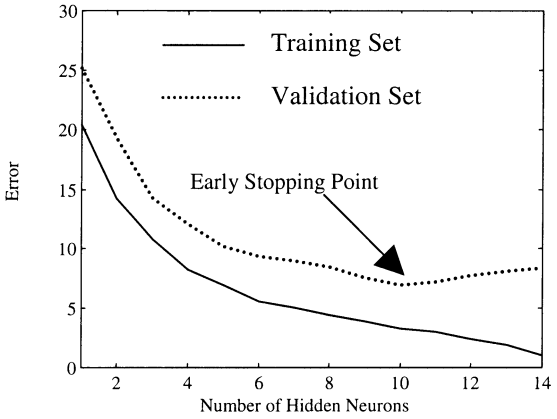

Fig. 5. Network performance, expressed in absolute relative error, versus the number of hidden neurons over data of training and validation set of the radiosity method neurons.

The constructive algorithm described in Section IV.B is used for training the neural network. Fig. 5 illustrates the network performance versus the number of hidden neurons over data of training and validation set in case of radiosity rendering algorithm. The network performance is evaluated as the average absolute relative prediction error. As is observed, the error on the training set decreases monotonically for an increasing number of hidden neurons. Instead, the error on the validation set decreases until ten hidden neurons and then it starts to increase. This is called early stopping point and is depicted in Fig. 5.

Fig. 6(a) presents the prediction performance of the proposed combined fuzzy classification-neural network model for the ray tracing algorithm. In this figure, the results have been shown for presentation purposes for the first 25 frames of the test set. The respective results for the other two investigated rendering algorithms are shown in Figs. 6(b) and (c). The experiments include different combinations of the rendering descriptors and refer to different synthetic scenes. It should be mentioned that there is no correspondence between the experiments of the three investigated algorithms, since there is no relation of the respective rendering descriptors. In all cases, the 3-D rendering workload has been estimated with respect to a reference synthetic scene, the cost of which has been measured on a PC AMD Athlon 1.60 GHZ of physical memory 128 MB of Linux Suse 2.4 operation system to be independent from the platform used. As is observed, the proposed scheme accurately predicts the rendering workload in all the examined cases.

An alternative way to illustrate the prediction accuracy of the proposed fuzzy-neural network model is to use the fractile diagrams or the quantiles-quantiles (Q-Q) plots. According to this method, the actual cost is plotted versus the predicted one. Therefore, perfect prediction lies on a line of 45° slope. The advantage of this method is that it shows all prediction differences with the same accuracy, regardless of the actual values to be predicted. It can be seen in Fig. 7 that the Q-Q plots for the three investigated rendering algorithms lie very close to the line of perfect fit, meaning that the proposed combined fuzzy-neural network model is a very good predictor of rendering computational complexity.

Finally, Table VI depicts the average relative prediction error over all data of the test set (i.e., 100 for each rendering type) for the three examined rendering algorithms and also compares the proposed scheme with other models. By examining this table, it is observed that the proposed fuzzy-neural network model yields efficient prediction performance in all cases (the worst prediction error is below 8%). In this table, we have also presented the prediction error obtained without applying the fuzzy classification scheme, i.e., using only the neural network structure and a binary organization of the extracted rendering descriptors. As is observed, the prediction performance deteriorates compared to the one obtained using the fuzzy classification scheme. In both the aforementioned cases, the neural network has been trained using the constructive scheme discussed in Section IV.B so that the network size is appropriately estimated. The effect of network size on the prediction performance is also shown in Table VI using a network of three and 25 neurons, respectively. As can be seen, both small and large size networks deteriorates the prediction accuracy performance.

The proposed fuzzy-neural network predictor is also compared with other linear and nonlinear models to illustrate the efficiency of the proposed scheme. In particular, initially, modeling is accomplished using a linear predictor, such as the moving average (MA). In this case, the prediction performance severely deteriorates providing unacceptable results since the error per sample often exceeds the 50%. Another comparison is accomplished by applying simplified nonlinear models. In particular, in this case, we have used mixed predictors of polynomial and exponential type. Although such approach improves prediction performance compared to the linear case, it still beyond the obtained by the proposed fuzzy-neural network scheme.

Comparisons with the works of [10], [12] are also accomplished in this Table. Since the approaches of [10], [12] are valid only for ray tracing rendering, the comparison is performed only for this type of algorithms. In all cases, the proposed scheme outperforms the compared ones. In addition, the proposed neural-network based architecture constitutes a general framework, which can be applied for other rendering types and parameters. In Table VI, we also present the prediction accuracy in case that a probability theory is used to estimate the membership grades, while the neural network is adopted for the final classification. As is observed, the average prediction error is less than the binary case but greater than the fuzzy case. This is due to the fact that fuzzy classification provides a more robust framework for ambiguity description in contrast to probability.

The effect of the membership function type and the number of partitions on the workload prediction performance is shown in Table VII. In this Table, triangular, trapezoid and quadratic membership functions have been examined at partitions 5, 10, 15, and 20, respectively. It is observed that the best performance is accomplished for triangular membership functions at $Q = 10$ partitions. The deterioration of the prediction accuracy at large number of partitions Q is due to the fact that at these cases feature vectors of large size are constructed, which complicates the neural network training.

B. Scheduling

In this section, we present the results concerning the scheduling algorithm implemented in the grid infrastructure. More

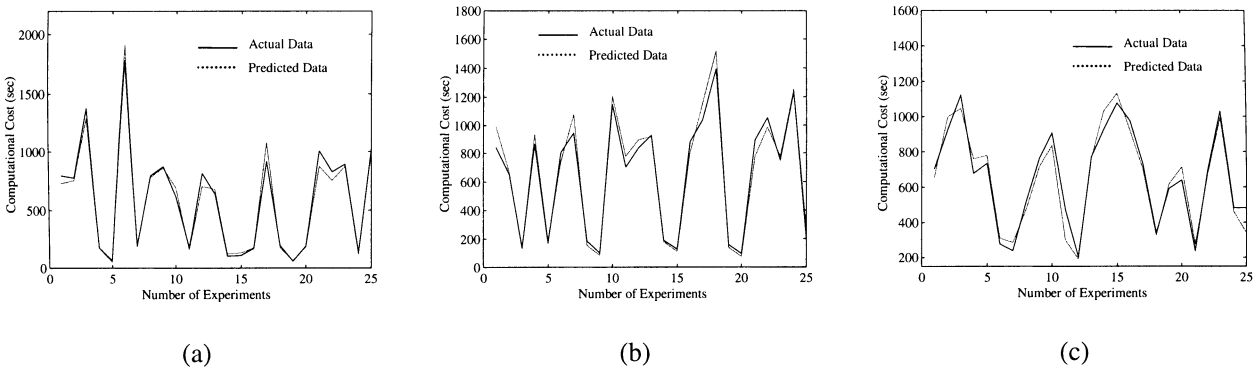

Fig. 6. Actual and the predicted computational cost of various experiments for three different types of rendering algorithms. (a) Ray tracing algorithm, (b) the radiosity algorithm, and (c) Monte Carlo irradiance algorithm.

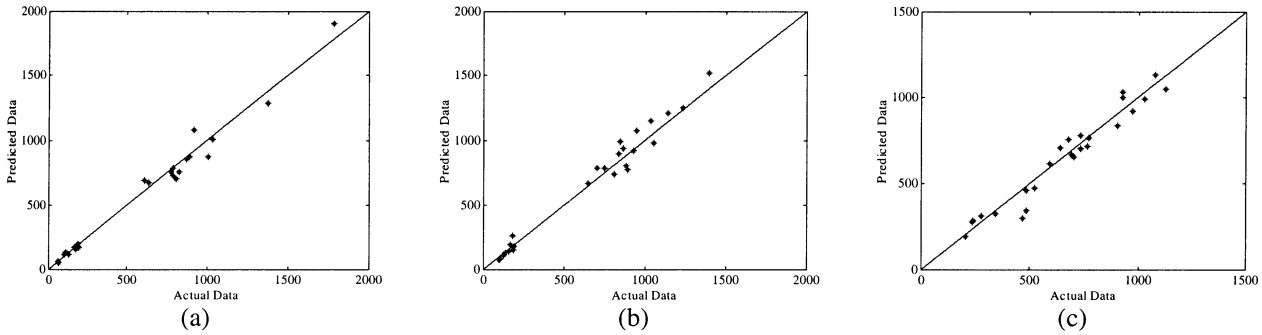

Fig. 7. Q-Q (Quantiles-Quantiles) plots for (a) the ray tracing algorithm, (b) radiosity algorithm, and (c) Monte Carlo irradiance algorithm.

TABLE VI
AVERAGE PREDICTION ACCURACY OVER ALL EXPERIMENTS CONDUCTED FOR THE THREE TYPES OF RENDERING ALGORITHMS. COMPARISONS WITH OTHER APPROACHES PRESENTED IN THE LITERATURE

	Average Absolute Relative Error		
	Ray Tracing (%)	**Radiosity (%)**	**Monte Carlo (%)**
Fuzzy-Neural Network Model	6.97	7.86	6.35
Neural Network	10.12	12.31	9.43
Fuzzy-Neural Network (Small Size)	13.02	14.25	13.22
Fuzzy-Neural Network (Large Size)	11.94	10.43	11.11
Non Linear Simplified Models (Mixed Polynomial + Exponential)	43.67	34.55	39.98
Posteriori Probability- Neural Networks	8.89	10.33	8.44
The method of [10]	23.30	-	-
The method of [12]	17.65	-	-

TABLE VII
EFFECT OF FUZZY MEMBERSHIP TYPE AND THE NUMBER OF PARTITIONS TO THE PREDICTION ACCURACY IN RAY TRACING ALGORITHMS

Ray Tracing Algorithms				
Partition Number Q	**Membership Function Type**			
	Triangular (%)	**Trapezoid (%)**	**Quadratic (%)**	**Binary (%)**
5	7.78	8.54	8.19	11.73
10	6.97	7.78	7.31	10.12
15	7.24	8.01	7.89	10.98
20	7.68	8.46	8.11	11.53

specifically, three different scheduling schemes has been developed and compared. The first assign the tasks in a first come-first serve (FCFS) policy. The second scheduling algorithm follows the earliest deadline first (EDF), while the third scheduling algorithm is based on the task fair completion times (FCT), (see Section V.A.II). In the FCT policy the tasks are assigned with the respective fair rates instead of the first two methods where the demanded rates are used.

In the FCFS and EDF policy the tasks either are executed with their demanded rate, X_i, or are rejected from execution (i.e.,

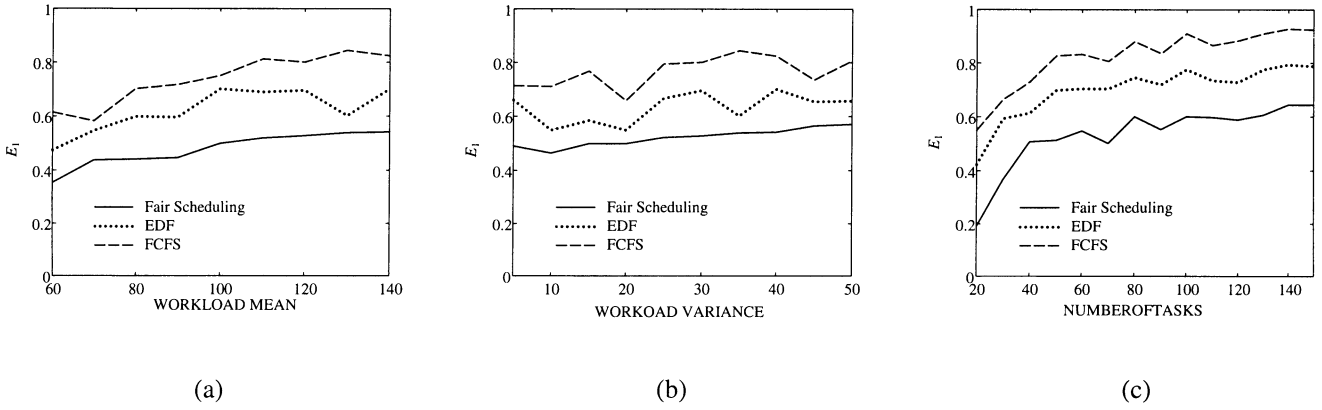

Fig. 8. Measure E_1 versus (a) average task workload (b) workload variance, and (c) number of tasks.

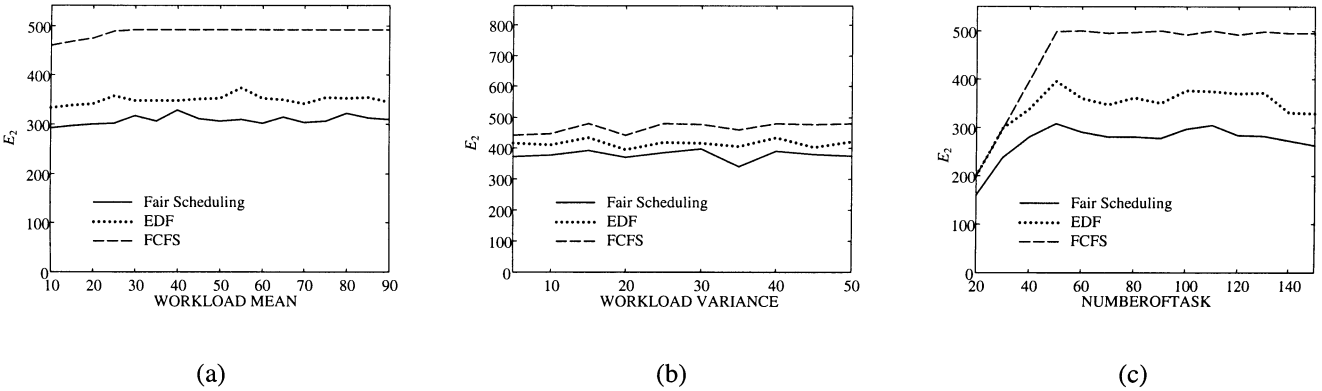

Fig. 9. Measure E_2 versus (a) average task workload, (b) workload variance, and (c) number of tasks.

their rate equals zero). On the contrary, in the FCT policy, all the tasks are executed but with the respective fair rates r_i, which are less or equal to the demanded rates $r_i \leq X_i$. Execution with a rate smaller that the demanded one, means that a violation of the respective task deadline is accomplished. In this case, the negotiation phase is activated to inform the user if he/she is willing to accept the deadline modification.

The efficiency of a scheduling scheme is measure either by the error of the demanded task rate to the actual rate that the task is executed, or by the sum of the task rates, that the scheduler serves.

$$E_1 = \sum_i \frac{|X_i - X_i^c|}{X_i} \text{ or } E_2 = \sum_i X_i^c \tag{8}$$

where X_i and X_i^c refers to the demanded and the actual task rates. The actual task rates equals $X_i^c = \{X_i, 0\}$ for the FCFS and EDF scheduling scheme (depending whether the task are assigned for execution or not) and $X_i^c = r_i$ for the FCT scheme.

Fig. 8 presents the results obtained by applying the three aforementioned schemes in case that the E_1 are used for measuring the scheduling efficiency, while Fig. 9 in case of E_2. More specifically, the FCFS, EDF and the FCT are applied for task ordering selection, in combined with the Earliest Completion Time (ECT) algorithm for processor assignment (see Section V). In all cases the measures E_1 and E_2 have been evaluated by a set of 20 independent experiments submitted to the grid infrastructure. Figs. 8 and 9 present the results of the measures E_1 and E_2 versus

1) average task workload;
2) workload variance;
3) number of tasks.

As is observed in all cases, the smallest error is achieved by the FCT scheme, with the EDF method comes the second. This is due to the fact that the FCT policy exploits better the grid resources than the other two approaches. It should be mentioned that only the FCT queuing order selection algorithm uses information of the workload prediction module. However, the workload prediction is required for the ECT processor assignment scheme for all queuing order selection schemes. In case that the task scheduling is accomplished without ECT processor assignment by distributing, for example, the tasks to the current available processor, a significant reduction of the scheduling performance is observed. This means that the measures E_1 and E_2 yield much greater values than the ones obtained by the ECT approach, verifying that an workload prediction is necessary for implementing efficient grid resource allocation.

VII. Conclusion

Computer graphics and 3-D rendering are an interesting commercial application useful for many fields such as simulation, design, research, education, entertainment and advertisement. However, a fundamental difficulty in achieving total visual realism of synthetic images is the complexity of the real world, which makes 3-D rendering be computationally intensive. On the other hand, grid architecture enables the integration of

diverse services and resources across heterogeneous and geographically distributed systems and presents them as a single unified integrated resource. For this reason, 3-D rendering can be performed more feasibly and reliably in the computational grid, which is verified by the European GRIA that aims at creating a grid testbed for 3-D rendering and dynamic structural analysis [4].

To implement, however, a 3-D rendering algorithm to a grid infrastructure, an efficient resource allocation management scheme should be developed to meet the QoS requirements as specified by the users. Toward this direction, modeling and workload prediction of 3-D rendering algorithms are required along with scheduling of the submitted tasks.

In this paper, both aspects are addressed. As far as the rendering workload prediction, a combined fuzzy classification and neural network model is proposed. Fuzzy classification is used for organizing rendering descriptors, while neural network for modeling and predicting the rendering workload. Fuzzy organization reduces possible noise and simultaneously provides a reliable framework for comparing scenes, composing of different number of synthetic 3-D objects, which results in an increase of prediction accuracy. Neural network models the unknown nonlinear relation of rendering descriptors to the respective computational complexity. For network training, a constructive algorithm is adopted in this paper, which simultaneously estimates network weights and size (i.e., network structure), instead of using conventional training schemes, such as the backpropagation, in which the network size is considered *a priori* known. Among different constructive approaches, the ones which yields the minimum computational cost with a simultaneous efficient performance is chosen.

Three different types of rendering algorithms have been investigated, the ray tracing, the radiosity and the Monte Carlo irradiance analysis. For each rendering type, a different neural network model is constructed, since each algorithm is characterized by different properties and parameters, which affect the computational complexity in a different way. Rendering descriptors are automatically estimated by parsing RIB formatted files. RIB format provides a general structure of describing a synthetic world. Several experimental results on real-life applications have been conducted to indicate the excellent performance of the proposed combined fuzzy classification-neural network model for prediction rendering workload. Therefore, grid scheduling and advance resource allocation mechanisms can benefit from the proposed predictor accelerating the transition of grid technologies from scientific collaboration to industrial and commercial applications.

ACKNOWLEDGMENT

The authors would like to thank the associate editor and all the anonymous reviewers, for their fruitful and constructive comments, which helped us improve the quality, presentation, and organization of the paper. Special thanks to D. Karaoglanoglou for the JAVA implementation of the scheduling algorithms in the adopted grid infrastructure.

REFERENCES

[1] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," *Int. J. Supercomput. Applicat.*, vol. 15, no. 3, 2001.

[2] W. Leinberger and V. Kumar, "Information power grid: The new frontier in parallel computing?," *IEEE Concurrency*, vol. 7, pp. 75–84, Oct./Dec. 1999.

[3] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice*, 2nd ed. Reading, MA: Addison-Wesley, 1997.

[4] "Grid resources for industrial applications (GRIA)," in *Proc. Eur. Union Program Information Societies Technology* , Brussels, Belguim, 2002, IST-2001–33 240.

[5] "Superscheduling," Scheduling Working Group of the Grid Forum, Doc. 10.5, 2001.

[6] A. D. Doulamis, N. D. Doulamis, and S. D. Kollias, "Recursive nonlinear traffic modeling of VBR MPEG-2 video sources," *IEEE Trans. Neural Networks*, vol. 14, pp. 150–166, Jan. 2003.

[7] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[8] J. G. Cleary and G. Wyvill, "Analysis of an algorithm for fast ray tracing using space subdivision," *Vis. Comput.*, vol. 4, pp. 65–83, 1988.

[9] J. D. MacDonald and K. S. Booth, "Heuristics for ray tracing using space subdivision," *Vis. Comput.*, vol. 6, pp. 153–166, 1990.

[10] K. R. Suramanian and D. S. Fussel, "Automatic termination criteria for ray tracing hierarchies," *Graph. Interface*, pp. 93–100, 1991.

[11] E. Reinhard, A. Kok, and A. Chalmers, "Cost distribution prediction for parallel ray tracing," in *Proc. Second Eurographics Workshop Parallel Graphics Visualization*, Rennes, France, Sept. 1998, pp. 77–90.

[12] E. Reinhard, A. J. F. Kok, and F. W. Jansen, *"Cost Prediction in Ray Tracing," Rendering Techniques*. New York: Springer-Verlag, 1996, pp. 41–50.

[13] E. Veach and L. J. Guibas, "Optimally combining sampling techniques for Monte Carlo rendering," in *Proc. SIGGRAPH*, Aug. 1995, pp. 419–428.

[14] M. B. Carter, "Parallel Hierarchical Radiosity Rendering," Ph.D. dissertation, Dept. Elec. Comput. Eng., Iowa State Univ., Ames, 1993.

[15] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.

[16] S. Walczak, "Neural network models for a resource allocation problem," *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, pp. 276–284, Apr. 1998.

[17] A. D. Doulamis, N. D. Doulamis, and S. D. Kollias, "An adaptable neural network model for recursive nonlinear traffic prediction and modeling of MPEG video sources," *IEEE Trans. Neural Networks*, vol. 14, pp. 150–166, Jan. 2003.

[18] J.-Q. Huang and F. L. Lewis, "Neural network predictive control for nonlinear dynamic systems with time delay," *IEEE Trans. Neural Networks*, vol. 14, pp. 377–389, Mar. 2003.

[19] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton Univ. Press, 1976.

[20] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[21] A. D. Doulamis, N. D. Doulamis, and S. D. Kollias, "A fuzzy video content representation for video summarization and content-based retrieval," in *Signal Processing*. New York: Elsevier, 2000, vol. 80, pp. 1049–1067.

[22] M.-C. Su, C.-W. Liu, and S.-S. Tsay, "Neural-network based fuzzy model and its application to transient stability prediction in power systems," *IEEE Trans. Syst., Man, Cybern. C*, vol. 29, pp. 149–157, Feb. 1999.

[23] O. Castillo and P. Melin, "Hybrid intelligent systems for time series prediction using neural networks, fuzzy logic, and fractal theory," *IEEE Trans. Neural Networks*, vol. 13, pp. 1395–1408, Nov. 2002.

[24] T.-Y. Kwok and D.-Y. Yeung, "Objective functions for training new hidden units in constructive neural networks," *IEEE Trans. Neural Networks*, vol. 8, Sept. 1997.

[25] R. Reed, "Pruning algorithms — A survey," *IEEE Trans. on Neural Networks*, vol. 4, pp. 740–747, Sept. 1993.

[26] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[27] T. Y. Kwok and D. Y. Yeung, "Constructive algorithms for structuring learning in feedfoward neural networks for regression problems," *IEEE Trans. Neural Networks*, vol. 8, pp. 630–645, May 1997.

[28] M. R. Azim-Sadjadi, S. Sheedvash, and F. O. Trujillo, "Recursive dynamic node creation in multilayer neural networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 242–256, Mar. 1993.
[29] P. S. Lewis and J. N. Hwang, "Recursive-least squares learning algorithms for neural networks," in *Proc. SPIE Advanced Signal Processing Algorithms, Architectures Implementations*, vol. 1348, 1990, pp. 28–39.
[30] J. Platt, "A resource allocating network for function interpolation," *Neural Comput.*, vol. 3, pp. 213–225, 1991.
[31] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[32] M. S. Fineberg and O. Serlin, "Multiprogramming for hybrid computation," in *Proc. IFIPS Fall Joint Computer Conference*, Thompson, Washington, D.C., 1967.
[33] S. Keshav, *An Engineering Approach to Computer Networking*. Reading, MA: Addison-Wesley, 1997.
[34] J. Brzezinski, J. Nabrzyski, J. Pukacki, T. Piontek, K. Kurowski, B. Ludwiczak, R. Strugalski, M. Hapke, N. Doulamis, A. Doulamis, E. Varvarigos, and K. Dolkas, "A grid application toolkit and tesbted," GridLab, Brussels, Belguim, IST 2001, 32 133, vol. D9.2, 2002.

Nikolaos D. Doulamis (S'96–M'00) received the Diploma degree (with highest honor) and the Ph.D. degree in electrical and computer engineering from the National Technical University (NTUA), Athens, Greece, in 1995 and 2000, respectively.

He joined the Image, Video and Multimedia Lab of NTUA in 1996 as a Research Assistant. From 2001 to 2002, he served his mandatory duty in the Greek army in the Computer Center Department of the Hellenic Air Force. Since 2002, he has been a Senior Researcher in the NTUA. His research interest include video transmission, content-based image retrieval, summarization of video sequences and intelligent techniques for video processing.

Dr. Doulamis was awarded as the Best Greek Student in the field of engineering in national level by the Technical chamber of Greece in 1995. In 1996, he was received the Best Graduate Thesis Award in the area of electrical engineering with A. Doulamis. His Ph.D. dissertation was supported by the Bodosakis Foundation Scholarship. During his studies he has also received several prizes and awards from the National Technical University of Athens, the National Scholarship Foundation and the Technical Chamber of Greece. In 1997, he was given the NTUA Medal as Best Young Engineer. In 2000, he was served as Chairman of technical program committee of the VLBV'01 workshop, while he has also served as program committee in several international conferences and workshops. In 2000, he was given the Thomaidion Foundation best journal paper award in conjunction with A. Doulamis. He is editor in the Who's Who bibliography and reviewer for IEEE journals and conferences, as well as other leading international journals.

Anastasios D. Doulamis (S'96–M'00) received the Diploma degree with highest honor and the Ph.D. degree in electrical and computer engineering from the National Technical University of Athens (NTUA), Greece, in 1995 and 2000, respectively.

From 1996 to 2000, he was with the Image, Video and Multimedia Lab of the NTUA as Research Assistant. From 2001 to 2002, he served his mandatory duty in the Greek army in the Computer Center Department of the Hellenic Air Force, while in 2002, he join the NTUA as Senior Researcher. In 2001, he served as technical program chairman of the VLBV'01. He has also served as program committee in several international conferences and workshops. He is reviewer of IEEE journals and conferences as well as and other leading international journals. He is author of more than 100 papers in the above areas, in leading international journals and conferences. His research interests include, nonlinear analysis, neural networks, multimedia content description, intelligent techniques for video processing.

Dr. Doulamis has received several awards and prizes during his studies, including the Best Greek Student in the field of engineering in national level in 1995, the Best Graduate Thesis Award in the area of electrical engineering with A. Doulamis in 1996 and several prizes from the National Technical University of Athens, the National Scholarship Foundation and the Technical Chamber of Greece. In 1997, he was given the NTUA Medal as Best Young Engineer. In 2000, he received the best Ph.D. thesis award by the Thomaidion Foundation in conjunction with N. Doulamis. His Ph.D. dissertation was supported by the Bodosakis Foundation Scholarship.

Athanasios Panagakis received the B.Tech. degree in electrical and computer engineering in 2000 and the Ph.D. degree in 2003 from National Technical University of Athens (NTUA), Greece.

Since 2000, he has been a Research Associate in NTUA. He has been involved in several national and EU research projects within Gridlab and Grid Resources for Industrial Applications (GRIA). His current research interests are focused on distributed computing and workload characterization.

Konstantinos Dolkas received the B.Tech. degree in electrical and computer engineering from National Technical University of Athens (NTUA), Greece in 2001 and is currently pursuing the Ph.D. degree from the same university.

Since 2001, he has worked as a Research Associate at NTUA. He has been involved in several national and EU research projects. His current research interests are focused on grid scheduling and workload characterization.

Theodora A. Varvarigou (S'88–M'92) received the B. Tech degree from the National Technical University of Athens (NTUA), Greece, in 1988, the M.S. degrees in electrical engineering and in computer science, as well as the Ph.D. degree, from Stanford University, Stanford, CA, in 1989 and 1991, respectively.

She worked at AT&T Bell Labs, Holmdel, NJ, between 1991 and 1995. Between 1995 and 1997, she worked as an Assistant Professor at the Technical University of Crete, Chania, Greece. Since 1997, she has been an Assistant Professor at NTUA. Her research interests include parallel algorithms and architectures, fault-tolerant computation, optimization algorithms, and content management.

Emmanouel Varvarigos was born in Athens, Greece, in 1965. He received the Diploma degree in electrical and computer engineering from the National Technical University of Athens (NTUA), Greece, in 1988, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1990 and 1992, respectively.

In 1990, he was a Researcher at Bell Communications Research, Morristown, NJ. From 1992 to 1998, he was an Assistant, and later an Associate Professor, at the Department of Electrical and Computer Engineering, University of California, Santa Barbara. From 1998 to 1999, he was an Associate Professor at the Electrical Engineering Department at Delft University of Technology, Delft, the Netherlands. In 1999, he became Professor at the Department of Computer Engineering and Informatics at the University of Patras, where he is now Director of the Communication Networks Lab. He is also the Director of Network Technologies Sector of the Research Academic Computer Technology Institute (RA-CTI). His research activities are in the areas of protocols and algorithms for high-speed networks, all-optical networks, high-performance switch architectures, grid computing, parallel architectures, performance evaluation, and ad-hoc networks.

Dr. Varvaigos was the organizer of the 1988 Workshop on Communications networks and was in the program committee of several international conferences.