Adaptive Intelligence Model to Enhance Dynamic Performance of Grid Enabled Applications

Mosleh M. Al-Harthi, Hatim Ghazi Zaini , Mohamed T. Faheem

Abstract — Recently, exploring effective ways for distributed resources that allow sharing large amounts of enabled applications among users in minimum time sharing are presenting new challenges to Grid Enabled Applications (GEA) in computer networks. A dynamical delay time model of GEA based on Neural networks (NN) technique is introduced. A new method for designing a dynamical adaptive intelligence controller based on a free delayed transmission factor is utilized in the adaptation process. An alternative unique and exact model without delays in the variables is also introduced. This alternative model which has no delayed in the state nor in the control is utilized in obtaining an identifier-based adaptive control based on the slow part of this alternative model. Two effective tools for designing this identifier which leads to have an intelligent adaptive controller based on the slow part of the exact model are presented. This exact model has the ability to overcome the drawback due to the presence of delayed transmission inquiry problem which leads to gain maximum resource utilization of GEA while minimizing task completion time. Using this slow part a classical time optimization scheduling algorithm based on back-propagation neural network is presented based on an algorithm that predicts the submitted task run time by training neural network through a training set of samples. In order to show the effectiveness of the proposed scheduling algorithm, a comparison with the bench mark min-min[1] scheduling algorithm and the GridSim time optimization(GridSim TO) [2] is presented.

Index Terms — Grid Enabled Applications, Adaptive Intelligence Model, Computing Networks, Time Delays Factors, Bench Mark min-min Package, GridSim Time Optimization.

1 INTRODUCTION

Recently, grid enabled applications (GEA) has come into attention in the literature and demonstrated the true power of the utilization of the Internet facility, where millions of applications stored in common PCs and sitting idle on desks around the world[1-3]. This kind of GEA can be treated in a way similar to those given for a Peer to Peer Network (P2PN). In a few years and due to a lack of standards and toolkits, early grid computing technology has treated GEA as if they would be P2PN. It is well known that both grid computing [3] and peer-to-peer (P2P) computing networks are emerging as next generation computing platforms for solving very large - scale computational and data intensive problems in science, engineering, and commerce [4].

It is well known that all transactions processes in Grid Computing environments, with different characteristics behaviors, are usually taken place in geographically distributed multiple administrative domains. The main objective of these kind of networking Their ability to allow all aggregated resources to shared in a wide variety of geographically distributed resources including supercomputers, storage systems, databases, data sources, and specialized devices that are owned by different organizations. However, the achievement of this resource sharing process needs to have some users, and applications requirements working together in these kind of environments [3].

Since, the resources are owned and managed by different organizations different organizations and agencies with different access policies and cost models that vary with time, users, and priorities, then this could lead to have different applications with different

• *Mosleh M. Al-Harthi is currently working as the vice dean of College of Engineering, Taif University, (KSA).*
Emails: mosleh2k@hotmail.com
• *Hatim Ghazi Zaini is currently working as the head of EE Dept. at College of Engineering, Taif University, (KSA).*
Emails: rhgz@hotmail.com
•*Mohamed T. Faheem is currently working in EE Dpt. at College of Engineering, Taif University, (KSA) & on leave from College of Engineering, Dept of Computers Eng.& Auto. Cont.l, Tanta University, Egypt.*
Emails: m_t_faheem@yahoo.com.

computational models that vary with the nature of the problem. Another important favor in integrated grid application process is the computational economy framework that provides a mechanism for regulating the supply-and-demand for resources and allocating them to applications which based on the deadline and budget constraints [4]. The administrative process of resource systems needs to provide mechanisms and tools that realize the goals of both service providers and consumers in the presence of transmission delays, where an incentive to resource owners is offered for sharing resources on the Grid and end-users where trade-off between the time frame for result delivery and computational expenses are taking places. It is also noted that a new dynamical model with transmission delays is introduced in the process in order to comply with the resource's consumers need. This model which depends solely on the resource demand, preferences, and brokers will automatically generate strategies for choosing providers. Based on this model which is converted into a free delay model has come recently into attention in the literature [5], where a grid resource broker (scheduler) is concerned with response time, system resource utilization, user satisfaction, and so on [6]. Based on all above important parameters in the designing process of dynamical model, we focus our effort to design a delay free scheduling algorithm that can cope with the "queuing order" and the "processor assignment" for a given task where the demand quality of service(QoS) parameters, i.e., the task deadlines, is satisfied as much as it can be. To achieve this goal, an evolutionary algorithm like time optimization based on back-propagation neural network (BPNN) is used. The main feature of this proposed scheduling algorithm is its implementation using the GridSim toolkit simulator [2]. Our procedure here will follow some similar lines as those given for P2PN as explained in our paper[7] as if they would be needed.

It is well known that one familiar definition for GEA is its ability to act as both a server and a client since it can provide services to other nodes applications as well as request services from other nodes applications. The functionality of grid computing networks is structured in two phases. In the first phase, a host is allowed to find other nodes application hosts and connect to the network, while in the second phase, this connected host is enabled to search for applications by broadcasting queries and test them for reputation based on some security rules to allow them to be downloaded. Any grid node can arbitrarily join or leave the network at any time and each node itself is responsible for making local autonomous decisions based on information received from other node in the network [8]. The GEA technologies exploit the CPUs and storage devices of these PCs to produce and exchange huge applications stores, communications systems, and processing engines. Therefore, an open GEA network is highly dynamic and autonomous[9]. For this reason ,the algorithm that is used with this grid network was designed for resource sharing across the global Internet in a minimal transmission time. Recently, [2], the GEA network is being treated as a network that is used to present a new service and function that are built completely at the application layer where its nodes interact via client programs running on their local machines irrespective of the underlying physical network. The resource searching, connectivity, routing, and other real applications are handled in a complete distributed way where every node nominally equal to every other recent client. The GEA is not more than just the universal applications-sharing model but it is also has a complete self organizing which requires no need for central instances to manage the network. Accordingly, the computation that occurred among GEA working groups, business applications fall into a number of categories and more details can be found in [10]. It should also be noted that, since application nodes are heterogeneous in their natural construction specially in both network and system capacity, then all other application nodes can be subsided with all their needs through the transactions that take place among themselves.

The most distinct characteristic of application GEA computing is its symmetric communication between the application nodes where each application node has both a client and a server role. The most advantages of the GEA systems are their construction as a multi-dimensional layers. This construction helped in improving some properties of the GEA network such as [7]:

1. Scalability by enabling direct real-time sharing of services and information.
2. Enabling knowledge sharing by aggregating information and resources from nodes that are located on geographically networks which enable networked hosts to share resources in a distributed manner.

Another an important issue in such GEA networks is their ability to efficiently search the contents of the other application nodes successfully, where most existing search techniques are based on either the idea of flooding the network with queries or with some form of global knowledge[7].

In recent years, most common definitions are given throughout researches where often certain terminology have distributed entities and are not identified in all systems in the same way. The definitions presented here for some terminology do represent what are given in most of references, however, often number of terms that are used to define a device or capability on a GEA network, e.g., resource, Grid Enabled Applications, Computing networks, time delays factors, bench mark min-min package, and GridSim time optimization,…etc will be given in a short sentences as much as it could be. In this section, common definitions are given which are used throughout this research. Therefore, wherever appropriate, the terminology provided here is given within the context of the system they are described within and some of these terms[7].

The rest of the paper is organized as follows: Section 2 reviews related work on a dynamical model of a grid enabled applications including a transmission delay based on a neural network in grid scheduling. In Section 3, we briefly explain an adaptive control system based on the slow mode of the dynamical construction schema. Section 4 describes the complete steps of an algorithm for dynamical slow mode of GEA based- back-propagation neural network using GridSim toolkit. Section 5 represents the simulation process of the slow part model of GEA using twenty resources with information describe the experimental simulation of the proposed algorithm based on about GridSim toolkit and compare its results with the conventional Min-Min algorithm and GridSim TO. Finally, conclusion is given at the rest of the paper.

2 Dynamical Model of a DEA Including Transmission Delays Based on a Neural Network

It is noted that in a grid enabled applications for a given interacting periods τ and within provider-consumer game perspective, each resource node can be described by his mode of behavior in the provider as well as in the Consumer role. That is, after each period τ, each peer is described by his dispositions. So, we consider a large number of resource providers and a large number of consumers. Providers and consumers meet and play by pair, frequently and randomly. The state of nature is also chosen at random again each time the application is enabled. At any point in time t , the population of providers is characterized by the fraction of providers who use each of the pure strategies which are available to them and so is the population of consumers. A reinforcement rule specifies how these fractions evolve[7]. Let us consider the following dynamical neural network to identify the grid enabled applications with some enhancement of those reported in[11] with an addition of a transaction delay signal τ that is included in the state variable be written as:

$$\dot{x}(t) = Ax(t) + A_1x(t-\tau) + W\varphi(x) + Bu(t) \quad (1)$$

Where $x \in R^n$, n is the state of the neural network, $u_t \varepsilon R^m$ is the input vector , A's and B are matrices of appropriate dimensions and their parameters contained all differences between the average utility of an i - node when meeting a randomly chosen j –node in the grid and the averaged utility obtained by all nodes [5-7]. $W \varepsilon R^{nxm}$ is a weight matrix. As it is mentioned in [Wen Yut], it is assumed that the vector field $\varphi(x) : R^n \longrightarrow R^m$ have the elements increasing monotonically to conclude some information about the original delayed grid transmission system. The elements of this vector field can be presented as sigmoid functions in the form [11]

$$\varphi_i(x_i) = a_i/(1 + e^{-b_i x_i}) - c_i$$

To deal with (1) as a delayed transmission neural network, we need to know perfect knowledge of the states in the regular dynamical form. Therefore, we first start transforming (1) into its unique and exact alternative form in the standard conventional form by moving the delay element τ from the states to the system parameters[said]. After accomplishing this steps and having (1) in its standard state space form with the assumption that no axis to measure the states directly. So, the next step is to use a modified estimator similar of that reported in [Wen Yut] to help obtaining the unavailable states by measuring only the output y and the input u of the process in (1). The next theorem is being used to accomplish our proposed scheme as follows.

Theorem1: for the linear dynamical model of a grid enabled applications transactions with delays transmission based on neural network described by (1), there always exit a linear transformation that moves the delays element from the state variables to the system parameters of the form

$$\dot{x}(t) = A\,x(t) + W\varphi(x) + B\,u(t) \quad \text{for } 0 \le t \le \tau \quad (2\text{-a})$$

And

$$T(\tau)\,\dot{x}(t) = \hat{A}\,x(t) + T(\tau)\,(W\varphi(x) + B\,u(t)) \quad \text{for } t \ge \tau \quad (2\text{-b})$$

With initial value $x(\tau)$ for $t \ge \tau$ is obtained from (2-a) at $t = \tau$ and (2-b) is a unique and exact alternative model of (1) for all $t \ge \tau$ in the form of generalized state space system, $T(\tau) = I + A_1 A(\tau)$, and

$$A(\tau) = \int_0^\tau e^{-A\theta}d\theta \quad , \quad \hat{A} = A + A_1$$

Proof: We prove this theorem by introducing the linear transformation introduced by Saidahmed [5] of the form

$$w(\tau, s) = e^{\tau s}x(s) \quad (3)$$

with initial value $w(0, s)$ is given by

$$w(0, s) = x(s)$$

By taking Laplace transform of (1) and applying the linear transformation given in (3), results in

$$\frac{dw(\tau,s)}{d\tau} = Aw(\tau,s) + e^{\tau s}x_0 + A_1x(s) + e^{\tau s}\,Bu(s) \quad (4)$$

Solving (4) with respect to $w(\tau, s)$, yields

$$w(\tau,s) = e^{A\tau}x(s) + \left[\int_0^\tau e^{A(\tau-\theta)}d\theta\right] A_1x(s) + \left[\int_0^\tau e^{A(\tau-\theta)}e^{\theta s}d\theta\right] Bu(s) + \left[\int_0^\tau e^{A(\tau-\theta)}e^{\theta s}d\theta\right]x_0 \quad (5)$$

By inspection, we note that the most right hand term of (5) can be rewritten as

$$\left[\int_0^\tau e^{A(\tau-\theta)}e^{\theta s}d\theta\right]x_0 = (sI - A)^{-1} * \left[e^{s\tau}I - e^{A\tau}\right]x_0 \quad (6)$$

And the third term on the right hand side of (5) can be integrated by part, so we have

$$\left[\int_0^\tau e^{A(\tau-\theta)}e^{\theta s}d\theta\right]Bu(s) = (sI-A)^{-1}\,[\,e^{s\tau}I - e^{A\tau}\,]\,Bu(s) \quad (7)$$

Collecting (6) and (7) all together and converting the result into the time domain, we end up with

$$x(t) = \left[e^{A\tau} + \int_0^\tau e^{A(\tau-\theta)}A_1\,d\theta\right] * x(t-\tau)u_s(t-\tau) + e^{At}x_0\,[\,u_s(t) - u_s(t-\tau)] + \int_0^t e^{A(t-\theta)}(W\varphi(x(t)) + B\,u(t)) \,[\,u_s(t) - u_s(t-\tau-\theta)]\,d\theta \quad (8)$$

where $u_s(.)$ stands for a unit step function .

It is clear from examining (8) that for $0 \le t \le \tau$, we get

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\theta)}(W\varphi(x(t)) + B\,u(t))\,d\theta, \quad 0 \le t \le \tau \quad (9\text{-a})$$

Obviously, (9-a) is in the form of the linear differential steady state

$$\dot{x}(t) = A\,x(t) + W\varphi(x) + B\,u(t), \quad 0 \le t \le \tau \quad (9\text{-b})$$

and the part of the dynamic system (1) for $t \ge \tau$ with initial value $x(\tau)$ can be also obtained from the (8) at $t = \tau$ as

$$e^{-A\tau}x(t) = (I + A(\tau)\,A_1)x(t-\tau) + \int_0^t e^{A(t-\theta)}(W\varphi(x(t)) + B\,u(t))\,d\theta \quad (10)$$

Substituting (10) into (1) and collecting similar terms, results in

$A(\tau)\dot{x}(t) = x(t) - x(t-\tau)$+A(τ)($W\varphi(x)$+Bu(t)), t≥τ (11)

Premultiplying (11) by A_1, using (1) and collecting similar terms, we end up with singular time invariant system of the form

T(τ)$\dot{x}$ (t) = $\hat{A}$ x(t) + T(τ)($W\varphi(x)$ + B u(t)) for t ≥ τ (12)

Obviously, (12) is in the form of linear dynamical model of a grid enabled applications transactions with no delays transmission in the generalized system form which contains the non-delay system as special case see [5] for more information about (12). To see this, let τ =0 in (12), yields

$\dot{x}$ (t) = $\hat{A}$ x(t) + $W\varphi(x(t))$ + B u(t) t ≥ 0 (13)

which shows direct verification of the present approach. It should be mentioned that (12) is also called a unique alternative representation of (1) in the sense that the behavior of the system is uniquely determined by (12). On the other hand and as seen by (9-b), the dynamical behavior of the system for $0 \le t \le \tau$ with τ > 0 takes the expected form that can be derived directly from(1) as

$\dot{x}(t)$ = Ax(t) + $W\varphi(x(t))$ +B u(t) (14)

Which means, we could have been obtained (14) by inspection from (1) by knowing that $A_1 x(t-\tau)u_s(t-\tau) = 0$ from 0≤t≤ τ. This strengths theorem (1) and supports the idea that (14) describes completely the behavior of the system (1) for $0 \le t \le \tau$. This completes the proof.

It is important to note that, in most practical cases the matrix T(τ) in (12) is invertible and this reduces the difficulty which usually encountered when dealing with states-delay systems. As proved in (33) and in case of having (12) as a singular system which must be checked first for solvability, then (12) can be divided into two essential subsystems: slow and fast subsystems. The slow part contains all the dynamical information about system (1) for t ≥ τ while the fast part include impulsive modes due to the existence of algebraic behavior only at t = τ .

Accordingly, the slow part of our general imitation model gives rise to dynamics which fall into a conventional standard class of models that are considered in evolutionary game theory namely regular, payoff-monotone dynamics. We note that [7] there are various properties of payoff monotone dynamics; where most results focus on the case of single population continuous time dynamics. As it is well known, the stability properties obtained for continuous-time dynamics (12) in general do not directly translate to discrete time formulations; because in discrete-time overshooting phenomena might destabilize equilibrium that are stable with respect to corresponding continuous-time dynamics [7]. To see the usefulness of the preceding approach let us examine the following dynamical model of a grid enabled applications transactions with delays transmission in the state for provider [7]. Our attention here will be focused on the effect of delayed signals on the behaviors of enabled node to another enabled node during the dynamical interactions and the interested people in analysis of grid computing evolution in grid enabled networks are advised to see [7],[9]. For the sake of testing the exactness and uniqueness solution introduced by theorem (1), we introduce the following example.

Example1: Let a Grid computing dynamics state-delay system for either providers or consumers be described by

$\dot{x}(t) = x(t-\tau) + \frac{1}{1+e^{-x}} - 0.5 + u(t)$,x(0) = 2.0 (15)

With transmission delay τ = 1, using theorem (1) for 0 ≤ t ≤ 1, we have

$\dot{x}(t) = \frac{1}{1+e^{-x}} - 0.5 + u(t)$ with x(0) =2.0 for 0 ≤ t ≤1 (16)

Using a nonlinear feed back in the form
u = kx - $\frac{1}{1+e^{-x}}$ + 0.5
with k= - 1 to stabilize (16), thus we get

$\dot{x}(t) = -x(t)$, x(0) = 2.0 (17)

Which has a solution given as

x(t) = $2.0e^{-t}$, 0 ≤ t ≤ 1 (18)

The initial value x(τ) to be used with the second part of theorem (1) for t ≥ τ is obtained from (18) at t = τ =1 as

x(1) = 0.736 (19)

The second part of theorem (1) is obtained from (12) for t ≥ 1 which is a unique and exact alternative form of (15) can also be obtained as

T(τ)$\dot{x}$ (t) = $\hat{A}$ x(t) + T(τ)($W\varphi(x)$ + B u(t)) for t ≥ τ (20)

By using A=0, A(τ =1) = A(1) = $\int_0^1 e^{-A\theta}d\theta = 1$, $\hat{A}$ = 1/4,
u(t) = -1- $\frac{1}{1+e^{-x}}$ + 0.5 , T(1) = I + A_1A(1) =1.25, $\hat{A}$ = 1/4, then (20) reduces to

$\dot{x}$ (t) = - 0.8 x(t) (21)

with initial value x(1) = 0.736, as obtained in (18) at t = 1, t ≥ 1.

It is clear from (21) that the unstable networked grid enabled applications systems with delayed - transmission with a nonlinear controller

u = f(kx) = kx - $\frac{1}{1+e^{-x}}$ + 0.5

has been stabilized by the proper choice of the nonlinear feedback gain u = f(kx) based on the same technique used with the conventional state space design approach for controlling the LTI systems. This results support the effectiveness of our

approach introduced in this paper. Next we show how to establish trust relationship in networked grid enabled applications systems based on an inelegant technique.

3 Adaptive Control System Based on Slow Mode of the Dynamical Construction Schema for Improving Grid Computing Systems

Recently, It is well known that GEA security issues has received considerable attention in the literature[6]. These issues have been raised due to uncertainty as well as the presence of the nullity of any trusted authority among providers and consumers connecting through GEA system. Therefore, we need to build a new rule to be used for constructing dynamical schema relationship among agents in the GEA system. It is well known the dynamical schema models are not well-suited for dynamic GEA environment, because most applications nodes need to connect and interact without being previously known to each other [7]. To design this kind of dynamical schema, we need to deal with system (2-b) in the regular standard form which can be transformed into two modes : slow and fast as had been reporten in Saidahmed [5]. From this paper it is well known that the slow mode of (2-b) takes the form

$$\dot{x}_s(t) = \hat{A}_s(\tau)x_s(t) + W_s(\tau)\,\varphi_s(x) + B_s(\tau)\,u(t), \text{ for } t \geq \tau \quad (22\text{-a})$$
$$y_s(t) = C(\tau)\,x_s(t)$$

and the fast mode is written as

$$\tilde{T}(\tau)\,\dot{x}_{f(t)} = \hat{A}_f(\tau)x_f(t) + W_f(\tau)\bar{\varphi}_f(x_f) + B_f(\tau)u(t) \text{ for } t \geq \tau \quad (22\text{-b})$$
$$y_f(t) = C x_f(t)$$

Where all related matrices and variables concerning the subscripts slow and fast modes can be found in more details in [7]. we note from [5] that the fast mode represented in (22-b) vanishes for all $t > \tau+$, therefore, we will focus our attention on (22-a) for designing the proposed dynamical schema of GEA.

Since it is assumed that all state variables are not available for direct measurement, then the neural network defined in (12) cannot be used directly in the design of an intelligent controller for (22-a) without having all the state available at hand. Therefore, a modified model free Estimator reported in[11] can be used to get a full-state estimation from the output measurement as

$$\dot{\bar{x}}_s(t) = \hat{A}_s(\tau)\bar{x}_s(t) + L\,\mathrm{sgn}\big(y_s(t) - \bar{y}_s(t)\big) - K\big(y_s(t) - \bar{y}_s(t)\big), t \geq \tau \quad (23)$$
$$y_s(t) = C(\tau)x_s(t)$$

Where $L = -\rho P^{-1}C^T$, and K is a positive gain matrix, $P = P^T > 0$ (T denotes transposed of a matrix) is a solution of the Lyapunov equation which is related to $\bar{A}$, C, and K, $p > 0$ is related to C in (23). Then the estimate $\bar{x}$ estimated from (25) is considered as the full-state of the grid computing process and it is utilized to obtain a neural network to identify the GEA model in (1). For More details about the (23) see [Wen Yut]. Based on system (23) and for the sake of brevity, it is an easy task to show that the required proposed dynamical adaptive intelligence model with delayed transmission factor follows similar lines as those reported in [7]. It is reported in [11] that this kind of adaptive controller have different categories of the multi-model neuro control that can be summarized as follows.

1- For one neural estimator and multiple neuro controllers, the modified neuro identifier adaptive control in terms of the transmission delay τ can be obtained from the sliding mode control as

$$u = -k\,P^{-1}\,sgn(e(\tau)),\ k > 0 \quad \text{for } t \geq \tau \quad (24)$$

where $e(\tau)$ denotes the bounded identification error in terms of the transmission delay τ.

2- For multiple dynamic neural networks controller, the modified neuro identifier in terms of the transmission delay τ takes the form

$$\dot{\bar{x}}_s(t) = A_{s\sigma}(\tau)\bar{x}_s(t) + W_s^{\sigma}(\tau)\varphi_{\sigma}(\bar{x}(t)) + B_{s\sigma}(\tau)u(t), \quad t \geq \tau \quad (25)$$

Wherea $\sigma \in Z = \{1, 2, ..., z\}$ is the switching input.

The free time delay multiple neuro controller defined in (24) and (25) are said to be intelligent grid computing adaptive control systems. First, we need to get a slow model which has no delay element in its variables nor in its control and then we design a neuro identifier which leads to get an adaptive control system. The first control approach is based on one neuro identifier while the second approach has utilized the notion of multiple neuro identifiers reported in, [11] where the uncertainties compensation of the bounded control error uses classical control technique where the bad transient response caused by the single neuro identifier could be overcome.

Since we transformed the dynamical delay time model of GEA based on Neural networks technique (1) to its alternative unique and exact model (2) without delays in the state variables nor in the control, then all conventional approaches that are reported in the literature can be easily extended to utilize the improvement of Grid resources management and time scheduling problems. In what to follow we extend some of our works introduced in [9] to develop our proposed TOBPNN scheduling algorithm based on the slow part of this alternative model.

4 An Algorithm for Dynamical Slow Mode of GEA Based- Back-propagation NN

It is well known that sigmoid function [12] is the most frequently activated function used in representing nodes of a dynamical delayed neural network as mentioned in system (1). To construct an algorithm to deal with this kind of network defined in (1) we construct the following algorithm.

i – we first set a weight matrix of a hidden layer and output layer with bounded error E and a learning rate with network training accuracy.

ii - A new training sample set is read with a single training

sample and using the current sample x_i , y_i , i runs from 1 to n, o construct a vector group named X and Y.

iii - The output of hidden layer and output layer with the formula $o_h = f(net_h)$ and $o_o = f(net_o)$ is then used to calculate the output of each layer.

iv - The output error of a single sample i denoted by the relation

$$\mathbf{E_i} = \frac{1}{2}\sum_{k=0}^{k-1}(y_{ik} - o_{ik})^2$$

is calculated and be used to get its minimum in the next step v.

v – In case of losing some the samples that couldn't read, then we should go back to step iii , otherwise we use the relation

$$E_{REM} = \sqrt{\frac{1}{i}\sum_{i=1}^{i}(E^i)^2}$$ to get the MSE

vi – Next, testing $E_{min} \leq E_{REM}$ and if it is satisfied, then the training is stop, otherwise we go to the next step vii.

vii - Finally, we adjust the weights of each layer by using the relations $W_{ih} = \eta\, o_h(y_k - o_k)$ and $W_{ho} = \eta\, o_{oj}(y_k - o_k)$.

4.1 Grid Resources Management and Scheduling Using GridSim Toolkit for the Slow Mode of Dynamical Model (1)

In order to demonstrate the effectiveness of resource brokers and associated scheduling algorithms for the dynamical model (22-a), their performance needs to be evaluated under different scenarios such as varying the number of resources and users with different requirements. In a real Grid environment, it is hard and perhaps even impossible to perform scheduler performance evaluation in a repeatable and controllable manner for different scenarios the availability of resources and their load continuously varies with time and it is impossible for an individual user/domain to control activities of other users in different administrative domains. The designers of resource management and scheduling systems and algorithms for large-scale distributed computing systems need a simple framework for deterministic modelling and simulation of resources and applications to evaluate their design strategies and algorithms. When access to ready-to-use test bed infrastructure is not available, building it is expensive and time consuming. Also, even if the test bed is available, it is limited to a few resources and domains; and testing scheduling algorithms for scalability and adaptability, and evaluating scheduler performance for various applications and resource scenarios is harder to trace and resource intensive. Researchers and educators in Grid computing have also recognized the importance and the need for such a toolkit for modelling and simulation environments [13]. We have used a Java-based discrete-event Grid simulation toolkit called GridSim. The toolkit supports modelling and simulation of heterogeneous Grid resources (both time- and space-shared), users and application models. It provides primitives for creation of application tasks, mapping of tasks to resources, and their management. To demonstrate suitability of the GridSim toolkit, we have simulated a Nimrod-G like Grid resource broker and evaluated the performance of deadline and budget constrained Min-Min and Time Optimization based on Back-Propagation neural network (TOBPNN) scheduling algorithms. Our interest in building a simulation environment arose from the need for performing a detailed evaluation of deadline and budget constraint scheduling algorithms implemented within the Nimrod-G broker [18]. We performed many experiments using the Nimrod-G broker for scheduling task farming applications on the WWG (World-Wide Grid) [24] test bed resources with small configuration (like 2 hours deadline and 10 machines for a single user). The ability to experiment with a large number of Grid scenarios was limited by the number of resources that were available in the WWG testbed. Also, it was impossible to perform repeatable evaluation of scheduling strategies as the availability, allocation, and usage of resources changed with time. Also conducting performance evaluation on a real Grid tested for a number of different scenarios is resource intensive and time consuming task, which can be drastically minimized by using discrete event simulation techniques.

The GridSim toolkit supports modelling and simulation of a wide range of dynamical heterogeneous resources, such as single or multiprocessors, shared and distributed memory machines such as PCs, workstations, SMPs, and clusters managed by time or space-shared schedulers. That means, GridSim can be used for modelling and simulation of application scheduling on various Classes of parallel and distributed computing systems such as clusters, Grids, and P2P networks. The resources in clusters are located in a single administrative domain and managed by a single entity whereas, in Grid and P2P systems, resources are geographically distributed across multiple administrative domains with their own management policies and goals. Another key difference between cluster and Grid/P2P systems arises from the way application scheduling is performed. The following section introduces system architecture for GridSim platform and components.

4.2 System Architecture

We employed a layered and modular architecture for Grid simulation to leverage existing technologies and manage them as separate components. A multi-layer architecture and abstraction for the development of GridSim platform and its applications is shown in Fig.1. The first layer is concerned with the scalable Java's interface and the runtime machinery, called JVM (Java Virtual Machine), whose implementation is available for single and multiprocessor systems including clusters [8]. The second layer is concerned with a basic discrete-event infrastructure built using the interfaces provided by the first layer. One of the popular discrete-event infrastructure implementations available in Java is SimJava [14]. Recently a distributed implementation of SimJava is also made available. Simulations in SimJava contain a number of entities each of which runs in parallel in its own thread. The third layer is concerned with modeling and simulation of core Grid entities such as resources, information services, and so on; application model, uniform access interface, and primitives application modelling and framework for creating higher level entities. The GridSim toolkit focuses on this layer that simulates system entities using the

discrete-event services offered by the lower-level infrastructure. The fourth layer is concerned with the simulation of resource aggregators called Grid resource brokers or schedulers. The final layer focuses on application and resource modeling with different scenarios using the services provided by the two lower-level layers for evaluating scheduling and resource management policies, heuristics, and algorithms. In this section, we briefly discuss SimJava model for discrete events (a second-layer component) and focus mainly on the GridSim (the third-layer) design and implementation. The resource broker simulation and performance evaluation is highlighted in the next two sections.

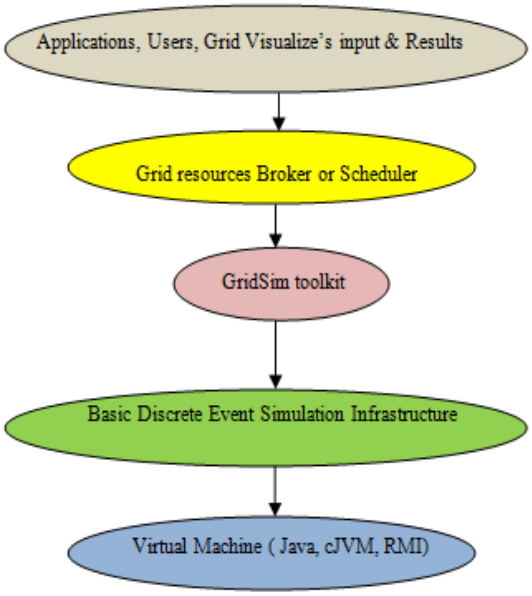

Fig.1 A modular architecture for GridSim platform and components [55]

4.4 Building Simulations with GridSim

To simulate the slow mode of (1) using GridSim toolkit, we note that the developers need to create new entities that exhibit the behaviour of Grid users and scheduling systems without delayed transmission. The slow mode is capable to inherit the properties of concurrent entities that are capable of helping user-defined entities to be extended to the GridSim base class communicating with other entities using events. The high-level steps involved in modeling resources and applications, and simulating brokers using the GridSim is important in the design approach. In order to achieve our goal we focus on the resource broker scheduling policy and propose a TOBPNN scheduling algorithm to be used within the scheduling flow manager as a choice that can be selected by the user in order to minimize the total completion time of users applications and to maximize the throughput. This proposed scheduling algorithm that is utilizing the dynamical neural networks (1) specially using the back- propagation neural networks that predict the task run time in order to take an accurate scheduling decision. This predicted information helps the designer of the proposed scheduling algorithm for (1) to achieve its objectives for minimizing the make span and maximizing the throughput. Next and in general, an introduction to the neural networks and the Back-propagation neural network mathematical background are introduced next.

4.3 Grid Computing with GridSim Based on TOBPNN Scheduling Algorithm

All components of grid computing which are user entities, broker entity, resources entities are simulated using Gridsim toolkit. In order to check the effectiveness of the proposed scheduling, it is compared with other approaches which are one of the well-known benchmark in this field. The detailed performance evaluation of economic driven scheduling algorithms is carried out through a series of simulations by hanging deadline, budget, application's gridlets, optimization strategies and simulating geographically distributed Grid. An object-oriented toolkit, called GridSim, for distributed resource modelling and scheduling simulation is used to simulate time and space-shared resources with different capabilities, time zones, and configurations. It also supports different application models that can be mapped to resources for execution by developing simulated application schedulers based on the architecture and components of the GridSim toolkit along with steps involved in creating GridSim as a based application-scheduling simulators. The Back-Propagation neural network (BPNN) algorithm that is introduced above is used in the computational economy as a metaphor for devising scheduling strategies for large scale applications on distributed resources. The simulation environment of grid enabled computing scheduling based on BPNN can be developed by creating a simulation Grid environment using the general simulation model in GridSim. The data is created from users with different Gridlets properties, length, size and different scheduling policies. As an application example in GridSim the properties of a task may be represented by a Gridlet object as seen from the following table 1.

Table 1: A Sample Gridlet Object.

Gridlet ID	Gridlet Length(MI)	Input File Size (MB)	Output File Size (MB)
1	4000	100	10
2	3502	77	950
3	7124	366	5433
4	9123	663	873
……	……	…….	……
n	5454	88	98

These Gridlets are owned by a user with different requirements. That means while we create a user, we also defined its number of Gridlets, connection speed (baud rate), maximum time to run simulation and scheduling policy like Gridsim time optimization (Gridsim TO),TOBPNN, and Min-Min scheduling algorithms as seen from Table 2.

Table 2: A Sample User Object.

User Name	Baud Rate (Mbit/s)	Max. Simulation Time (hour)	Scheduling Policy	# of Gridlets
User 1	100	10 hour	TOBPNN	50
User 2	200	15 hour	Min-Min	300
User 3	400	10 hour	Cost opt.	150
.....	….	…..	…..	…..
User n	150	10 hours	Time opt.	200

Our goal next is to use GridSim default broker as a resource broker for creating a Task Run Time Estimation Model which consists of two classes and being used in scheduler adviser method in GridSim broker class. The TOBPNN algorithm then finds the most suitable resource based on the expectation of the task run time in all available resource then computes the minimum completion time. Finally it assigns the Gridlet to the found resource. Before the simulation start, a user creates an experiment that acts as a placeholder. It is composed of GridletList that stores a set of Gridlets to be processed and user requirements with a scheduling policy. Whenever simulation start the broker creates a resource list to store dynamic information and characteristic properties of available resources acquired from the GIS. During the simulation each broker continuously queries the GIS and gets dynamic information about the available resources and load on these resources then the user sends its application to its resource broker via the application interface. The broker of each user gets their Gridlets from its experiment object via experiment interface of that broker. After that the broker of user puts all Gridlets to be sent for execution into the unfinished GridletList. In our design, the problem of having delay time is no longer exit since we use the slow mode of (1). That means each user doesn't wait for long time and sends all Gridlets at one moment packed in experiment object to its brokers. The advantage of developing dynamical GEA scheduling algorithm allows a new Gridlet during the simulation to be served. That means new users can send new Gridlets without waiting for the current simulation to be finished as it happened with the static operation. Then the broker scheduling all Gridlets at once and scheduling them according to their requested inquiry. Therefore, resource management engine selects the most suitable resource for the Gridlets according to minimum completion time computed by the TOBPNN where Gridlets mapped to a specific resource are added to the Gridlet List by the broker. A dispatcher embedded in each broker selects the number of Gridlets from the Gridlet list and assigns them to the resource according to that resource with no occurrences of overloaded. Since a different type of resources in terms of time and space shared is created, all Gridlets executed depending on the resource type are assigned to them. In the following sections we will discuss in details the Modelling of Grid Scheduling Framework [9].

5 Simulating Slow Part Model of GEA Using Twenty Resources

In this section we exercise a simulation process using 20 resources with different characteristics as shown in Table 3. In implementing this simulation of virtual grid environment some applications gridlet's are submitted as shown in Table 4. The performance evaluation among the three proposed TOBPNN, GridSim TO, and the Min-Min scheduling algorithm are summarized in Fig.1 in terms of the total completion time. The improvement ratio is shown in both Fig.2 and Fig.3.

Table 3 Resources characteristics in case of using 20 resources

Resource Name	No. Of Processors	Resources Type/ Node OS	Rating (MIPS)	Cost per MI ($/MI)
Resource1	69	"Sun Ultra", "Solaris"	23201	0.2642385639626583
Resource 2	65	"Intel Pentium", "Linux"	26065	0.05818470789512373
Resource 3	43	"Intel Pentium", "Linux"	7009	0.3644420647246252
Resource 4	84	"Sun Ultra", "Solaris"	23942	0.08673930498867856
Resource 5	83	"Compaq AlphaServer", "OSF1"	22392	0.24656192594983423
Resource 6	31	"Intel Pentium", "Linux"	8342	0.3740509344179928
Resource 7	26	"Intel Pentium", "Linux"	4264	0.2839333342237074
Resource 8	50	"SGI Origin", "Irix"	11000	0.08916689959763686
Resource 9	50	"Sun Ultra", "Solaris"	11000	0.19374108743859503
Resource 10	30	"Intel Pentium", "Linux"	6550	0.10140968253672342
Resource 11	38	"Sun Ultra", "Solaris"	380	7.666726370567578
Resource 12	42	"Compaq AlphaServer"	3864	0.8859346238636973

		, "OSF1"		

Table 3 Resources characteristics in case of using 20 resources (continues)

Resource 13	47	"Intel Pentium", "Linux"	15064	0.21783724382 1147
Resource 14	40	"Compaq AlphaServer" , "OSF1"	16800	0.16889943478 64112
Resource 15	46	"Compaq AlphaServer" , "OSF1"	9614	0.17405151497 310065
Resource 16	39	"Compaq AlphaServer" , "OSF1"	9179	0.18556626919 114733
Resource 17	101	"SGI Origin", "Irix"	46460	0.18234707949 873716
Resource 18	75	"Compaq AlphaServer" , "OSF1"	26700	0.14258818965 816747
Resource 19	48	"SGI Origin", "Irix"	7033	0.95672611315 66703
Resource 20	113	"Intel Pentium", "Linux"	29719	0.12809602372 75591

Table 4 summarizes the obtained results using the three min-min, GridSim TO, and TOBPNN scheduling algorithms. The comparison regarding performances between the proposed algorithm and other two known algorithms are done using two criteria: the first one is on the total completion time (Make span) while the second one is implemented on the total spent budget, for more details see [9].

Table 4 Summary of performance evaluation in case of using 20 resources

Test No.	NO. GL.	Deadline Time	Spent time(sec)			Per. Imp. Min-Min &TOBPNN	Per. Imp. TOBPNN & GridSim TO	Available Budget
			Min-Min	TOBPNN	GridSim TO			
1	100	3.82497 06831 467444E 8	60 822	60 19 7	609 02	1.027 589	1.171 1547	3.674 11671 99750 E11

Table 4 Summary of performance evaluation in case of using 20 resources (Contiues)

2	150	5.7362892 33339657 E8	89 56 9	80 80 8	89 76 3	9.78 128 6	4.89 431 74	5.50923 656058 7E11
3	200	7.6505369 95527767 E8	11 72 97	11 10 21	11 81 26	5.35 052	6.39 969 01	7.34716 883403 5E11
4	250	9.5542996 00727187 E8	14 62 96	13 36 68	14 83 92	8.63 181 5	25.9 777 96	9.17503 396319 71E11
5	300	1.1462691 76313283 2E9	17 42 92	17 01 20	17 62 91	2.39 368 4	3.62 743 95	1.10073 440825 89E12
6	350	1.3388450 99522185 3E9	20 32 28	20 21 01	20 67 38	0.55 455	2.29 439 74	1.28563 288971 25E12
7	400	1.5301488 13450669 5E9	23 19 59	23 15 39	23 52 61	0.18 106 6	10.2 453 58	1.46930 988116 833E12
8	450	1.7209468 76872738 6E9	25 96 06	25 94 80	26 38 29	0.04 853 5	1.67 604 44	1.65250 139945 034E12
9	500	1.9118903 01580651 5E9	28 87 46	28 56 24	29 38 27	1.08 122 7	2.87 195 75	1.83583 246536 848E12
10	550	2.1025982 35504971 E9	31 74 41	31 23 21	32 45 26	1.61 289 8	3.90 783 84	2.01893 744314 0E12
11	600	2.2932989 45124483 6E9	34 67 85	34 16 66	35 11 72	1.47 613 1	2.78 224 93	2.20203 547031 4E12
12	650	2.4847028 85081636 4E9	37 52 79	36 91 59	38 18 27	1.63 078 7	3.43 158 37	2.38580 869207 840E12
13	700	2.6751423 93796667 E9	40 54 27	40 03 38	41 15 62	1.25 522	7.79 940 95	2.56865 663375 487E12
14	750	2.8654636 16632153 5E9	43 41 24	43 00 63	44 04 93	0.93 544 7	7.07 570 75	2.75140 200176 32E12
15	800	3.0561943 76285332 E9	46 18 99	45 85 00	47 17 26	0.73 587 5	5.06 564 89	2.93454 060472 124E12

Since our approach is focusing on the time optimization for GEA and for the sake of comparison, we construct three scheduling algorithms: the min-min, the GridSim TO, and TOBPNN scheduling algorithm using 20 resources as shown in Fig.1. It is clear from Fig.1 that the proposed TOBPNN scheduling algorithms gives better total completion time than those reported by both the benchmark min-min and GridSim TO. The percentage improvements among the previous scheduling algorithms are shown in Fig.2 and Fig.3. Fig.2 shows the percentage improvement between TOBPNN and Min-Min scheduling algorithms while Fig.3 indicates Comparison between TOBPNN and GridSim TO using 20 resources. The percentage improvements among the previous scheduling algorithms are shown in Fig.2 and Fig.3. In Fig.2, it is shown that the percentage improvement between TOBPNN and Min-Min scheduling algorithms while Fig.3 indicates Comparison between TOBPNN and GridSim TO using 20 resources.

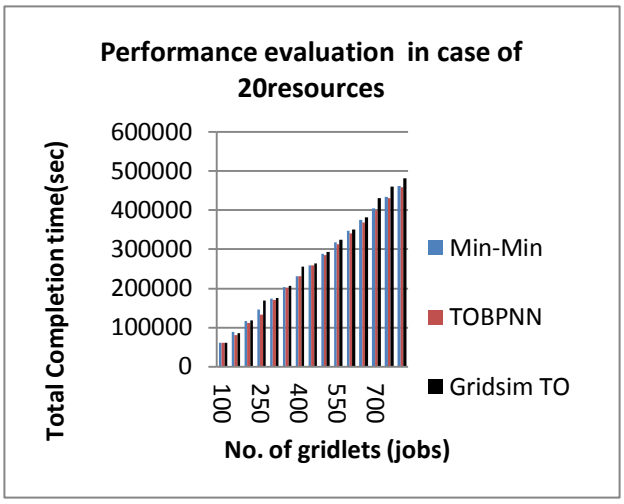

Fig. 1 Performance comparison among Min-Min, TOBPNN, and GridSim TO

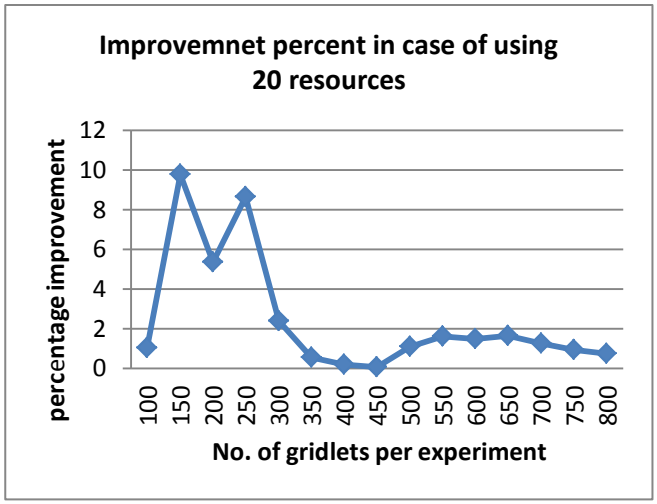

Fig.2 Percentage improvement between the Min-Min and the TOBPNN using 20 resources

Fig.2 shows that the improvement percent between Min-Min and the TOBPNN using 20 resources decreases towards the higher number of gridlets. It is also clear from the figure that the improvement occurred due to the proposed TOBPNN varies in the range approximately from 0% to 1.8%. The low improvement shown in the figure is occurred because when the number of resources and gridlets increased, the numbers of threads that are generated by the GridSim toolkit are also increased. It should also be noted that the increment in threads that run simultaneously on parallel GEA needs a large computer specification with great capabilities. On the other hand, in Fig.3 one can easily see that the improvement percent between TOBPNN and GridSim TO goes a bit larger than that is given in Fig.2 and varies approximately between 0% and 10%. In essence, we conclude that the proposed algorithm TOBPNN gives better operation in GEA than those reported in both Min-Min and the GridSim TO.

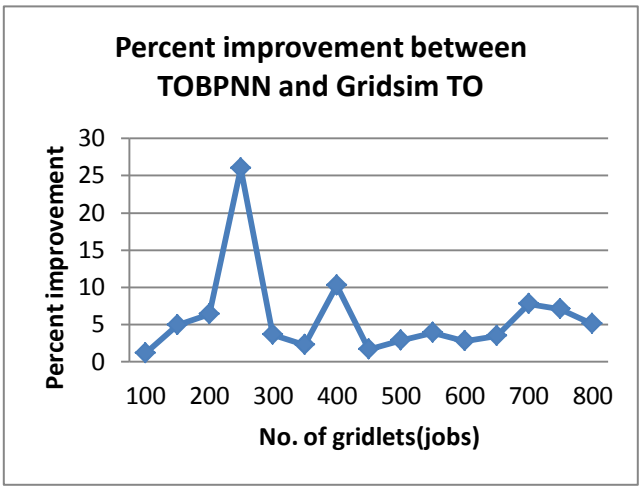

Fig. 3 Percentage improvement between the TOBNN and the GridSim TO

6. CONCLUSION

This paper introduced effective ways for Grid Enabled Applications (GEA) in computer networks with distributed resources that allow sharing large amounts of enabled applications among users in minimum time sharing. A dynamical delay time model of GEA based on neural networks technique has been constructed and was shown to be suitable for GEA in computer networks. The presence of transaction delay in the grid resources networks has been tackled to overcome the drawback due to the presence of this delay. Including the dynamical delay factor in the process has strengthen our work based on a new novel approach that helped in improving the stability and robustness of the transaction process as well as overcoming many drawbacks issues that appeared in grid enabled applications networks which include delayed time transmission elements.

A new method for dealing with such delayed model was utilized in the adaptation process and used in the designing process of a dynamical adaptive intelligence controller based on an alternative unique and exact model which has no delays in the state variables

nor in the control. This alternative model has been utilized for obtaining an identifier-based adaptive control which based on the slow part of this alternative model. It was shown that using this exact model has the ability to overcome the drawback due to the presence of delayed transmission inquiry problem which could lead to gain maximum resource utilization of GEA while minimizing task completion time. This new alternative model has also been used to design an identifier for the proposed dynamical GEA which was easily implemented on overlay networks without extra cost.

Two effective tools for designing this identifier were presented and have lead to have an intelligent adaptive controller based on the slow part of the exact dynamical model. Since the slow part of the dynamical model is in the form of conventional neural networks (NN), so the improvement behaviors of GEA model based on NN has been achieved in the regular sense and follow almost similar lines as those reported in the literature [9] based on an object-oriented toolkit, called GridSim. This GridSim simulator was used to simulate both time- and space-shared resources with different capabilities, time zones, and configurations. This GridSim toolkit simulation package is considered as the best popular simulation package in this field [9]. This package has helped us in obtaining the best improvement of the required shortest path including the effect of transmission delays, where the performance of transactions among grid enabled applications nodes have been greatly improved. The great achievement in this work is its new treatment of the presence of transmission delays due to the presence of transaction delays where all delayed elements have been completely removed to the system's parameters. This new treatment has led to obtain the most minimal time evaluation algorithms based on Fuzzy decision approach. Based on the slow part of the alternative model, a time optimization scheduling algorithm based on back-propagation neural network was obtained which has the ability to predict the submitted task run time by training neural network through a training set of samples. For the sake of showing the effectiveness of the proposed scheduling algorithm, a comparison with the bench mark min-min[40] scheduling algorithm and the GridSim time optimization(GridSim TO)[7] has been discussed and the results end up with an indication that the proposed algorithm TOBPNN gave better results for GEA than those they were reported in both Min-Min and the GridSim TO with the same distributed resources.

REFERENCES

[1] Elmroth, E. and J. Tordsson, "Grid resource brokering algorithms enabling advancereservations and resource selection based onperformance predictions", Journal of Future Generation Computer Systems, 24, 2008,pp585-593.

[2] Buyya,R., and Murshed,M. GridSim: A Toolkit for the Modeling and Simulation Distributed Resource Management and Scheduling for Grid Computing, The Jour. of Concurrency and Computation: Practice and Experience (CCPE), Wiley Press, May 2002 .

[3] Armstrong, D., Hensgen, andKidd, T. , "The Relative Performance of VariousMapping Algorithms is Independent of Sizable Variances in Run-timePredictions", 7th IEEE Heterogeneous Computing Workshop (HCW '98), pp.79-87, Mar. 1998.

[4] B.Khoo, et al "A multi-dimensional Scheduling Scheme in a Grid Computing Environment", Journal of Parallel and Distributed Computing, 67, 2007, pp659-673.

[5] Saidahmed, M. T. F., "A New Approach for Designing a feedback Controller for a Wind Tunnel Model Involving a Delay", Proc. of the 35th Mid. Symp., GWU, IEEE Circuits and Systems Society, Aug. 9-12, 1992.

[6] Berstis, "Fundamentals of Grid Computing", IBM RedBooks Paper, November 2002.

[7] Mosleh M. Al-Harthi , Mohamed T. Faheem , Ahmad Aziz Al-Alahmadi," An Intelligent Technique for Improving Data Access Based on Merging Some of Protocols package using P2P Delayed Networks," Intern, Jour.of Scientific &Technology Research, VOL 4, ISSUE 2, Feb.2013.

[8] Abraham, R. Buyya, and B. Nath, "Nature's Heuristics for Scheduling Jobs on Computational Grids", Proceedings of 8th IEEE International Conference onAdvanced Computing and Communications, 2000.

[9] M. Faheem, et al," Grid Computing Scheduling Based on Neural Networks," IJICIS, Vol.*11, No. 2*, july 2011.

[10] C.Gui, et al, " Towards Trustworthy Resource Selection: A Fuzzy Reputation Aggregation Approach", B. Xiao et al. (Eds.): ATC 2007, LNCS 4610, pp. 239–248, c_Sprin.-Verlag Berlin Heidelberg 2007.

[11] Wen Yut and Xiaoou Lit," Adaptive Control with Multiple Neural Networks," Proceedings of the American Control Conference Anchorage, AK May 8-10,2002.

[12] http://en.wikipedia.org/wiki/Backpropagation.

[13] Weissman, J.,"Grids in the Classroom, IEEE Distributed Systems Online", Volume 1, No. 3, 2000.

[14] Howell, F. and McNab, R., "SimJava: A Discrete Event Simulation Package For Java With Applications In Computer Systems Modelling ", First International Conference on Web-based Modelling and Simulation, San Diego, CA, Society for Computer Simulation, January 1998.