A Comparison Study of Metaheuristic Techniques for Providing QoS to Avatars in DVE systems ⋆

P. Morillo[1], J. M. Orduña[2], M. Fernández[1] and J. Duato[3]

[1] Instituto de Robótica. Universidad de Valencia. SPAIN.
[2] Departamento de Informática. Universidad de Valencia. SPAIN
[3] DISCA. Universidad Politécnica de Valencia. SPAIN
Pedro.Morillo@uv.es, Juan.Orduna@uv.es

Abstract. Network-server architecture has become a de-facto standard for Distributed Virtual Environment (DVE) systems. In these systems, a large set of remote users share a 3D virtual scene. In order to design scalable DVE systems, different approaches have been proposed to maintain the DVE system working under its saturation point, maximizing system throughput. Also, in order to provide *quality of service* to avatars in a DVE systems, avatars should be assigned to servers taking into account, among other factors, system throughput and system latency. This highly complex problem is called *quality of service (QoS) problem* in DVE systems. This paper proposes two different approaches for solving the QoS problem, based on modern heuristics (simulated annealing and GRASP). Performance evaluation results show that the proposed strategies are able no only to provide quality of service to avatars in a DVE system, but also to keep the system away from the saturation point.

1 Introduction

Distributed Virtual Environments (DVE) are systems where many users can connect their client computers through different networks and interact in the same 3D virtual scene [16]. Each user of the DVE appears in the virtual world as an entity, usually humanoid, called *avatar*. Avatars are controlled by the users, and each avatar offer a different point of view of the scene. DVE systems are currently used in many different applications such as collaborative design [15], civil and military distributed training [7], e-learning [8] or multi-player games [5]. Nowadays, most of current DVE systems have a network-server architecture. In this architecture (also denoted as mirrored-server) each user of the system is assigned to a server, so that when a client performs a movement in the virtual scene it sends updating messages to the server where it is assigned to. This server will be responsible for distributing this message to the rest of clients and servers of the system, in order to maintain a consistent view of the virtual world for all the avatars. In order to avoid a message outburst when the number of clients increases, concepts like areas of influence (AOI) [16] have been defined. This concept describes a neighborhood area for avatars, in such a way that a given avatar must notify his movements (by sending an updating message) only to those avatars located in that neighborhood. These destination avatars are denoted as neighbor avatars.

⋆ Supported by the MCYT under grants DPI2002-04438-C02-02 and TIC2003-08154-C06-04

Lui and Chan have shown the key role of finding a good assignment of avatars (clients) to servers in order to ensure both a good frame rate and a minimum network traffic in DVE systems [6]. The problem of efficiently assigning avatars to the different servers of the system is called the *partitioning problem* [6], and several approaches have been proposed for solving it. Lui and Chan model the problem as a formal numerical optimization problem and obtain the solution using an ad hoc algorithm [6]. Keeping the same specifications of the problem, the results obtained by this technique have been improved by using metaheuristic techniques [9]. However, none of these approaches takes into account the non-linear behavior of DVE systems with the number of avatars in the system, shown in [10]. This work shows that the main purpose of any partitioning method should be keeping all servers in the system away from reaching 99% of CPU utilization. Otherwise, the entire DVE system enters saturation and system latency greatly increases. Recently, an adaptive strategy that takes into account this non-linear behavior of DVE systems has been proposed for solving the partitioning problem [12].

However, once the partitioning method has ensured that the system is under its saturation point (it has provided a partition where the estimated percentage of CPU utilization in all the DVE servers is under 99%), then the computing resources can still be used to decrease the average system time response provided to avatars. This improvement should be carried out also by the partitioning method, since it is really a trade-off between system throughput and system latency. The problem of solving the partitioning problem ensuring that the system is under its saturation point and at the same time the average latency provided to avatars is minimized is known as the *quality of service problem (QoS problem)*. This problem can be modelled as finding a partition minimizing a new quality function.

In this paper, we propose a comparative study of two different metaheuristics for solving the QoS problem. One of them is Simulated Annealing (SA), a stochastic metaheuristic. The other one is GRASP, a constructive metaheuristic. Performance evaluation results show that the proposed metaheuristics are both valid methods for solving the QoS problem, simultaneously providing quality of service to a large set of avatars and also maintaining the DVE system under the saturation point. Therefore, they can be used as a valid partitioning methods to provide QoS to avatars in DVE systems. The rest of the paper is organized as follows. Section 2 details the problem of providing QoS to avatars and how it has been addressed in DVE systems. Also, we propose in this section a method to provide QoS through the partitioning problem. In Section 3, we describe the tuning of two different heuristics when applied to the solving of this problem. Next, Section 4 presents the performance evaluation of the proposed heuristics. Finally, Section 5 presents some concluding remarks and future work to be done.

2 The Quality of Service Problem in DVE Systems

The Quality of Service problem (QoS problem) has been already described in DVE systems, and some strategies have been proposed for solving it [1, 17]. Approaches like [17] use latency compensating methods in order to repair the effects of high network jitter. Adaptative rendering strategies like [1] or [16] modify the resolution of the 3-D models depending on the client connection speed. However, none of these strategies takes into account the non-linear behavior of DVE systems with the workload assigned to each server, as described in [10]. Therefore, these strategies cannot guarantee that the performance provided to avatars will not degrade beyond any threshold value.

QoS problem can be expressed in DVE systems as latency constraints. In order to fulfill these constraints, and taking into account the non-linear behavior of these systems described in [10], a trade-off among server saturation, clients' interactivity and system stability must be reached. A DVE system can only offer QoS to clients if it is working under its saturation point and at the same time the average round-trip delay for the messages sent by each avatar (denoted as ASR, for *average system response*) is lower than 250 ms. [17]. However, the ASR provided to a given avatar i depends on where avatars located in the AOI of i are assigned to. If avatar i is assigned to server s then the ASR for avatar i linearly decreases with the number of avatars in the AOI of i that are migrated from other servers to server s . Therefore, the problem of offering QoS to avatars can be expressed as a new partitioning problem. A partitioning method that provides QoS to avatars will have to maximize the number of neighbor avatars assigned to the same server and at the same time it will have to keep the system away from saturation. Additionally, since this strategy is a global load balancing scheme it must not migrate more than 30% of avatars in the system [4]. Therefore, the partitioning problem will consists of finding a partition complying with all these three requirements.

In order to solve this partitioning problem, we propose a quality function that takes into account all these requirements. Equation 1 represents the proposed evaluation function, composed of three terms. This quality function measures the quality of each partition (assignment of n avatars to s servers).

$$f_{QoS} = \sum_{i=0}^{s} h_{cpu}(i) + \sum_{j=0}^{n} h_{asr}(j) + n_m \qquad (1)$$

The term $h_{cpu}(i)$ is a function of the percentage of CPU utilization in server i. The behavior of this function is exponential, as shown in Figure 1(a). While the percentage of CPU utilization in server i is under 80%, this function provides a low value. However, as the percentage of CPU utilization goes beyond this threshold value, function $h_{cpu}(i)$ greatly increases. In this way, this function rejects any partition where at least one of the DVE servers is close to saturation.

The term $h_{asr}(j)$ is a function of the ASR provided to avatars by the systems, and it is composed of two sections. The section from that zero value to an ASR of 25 ms. is has an inverse exponential behavior, as shown in Figure 1(b). From 250 up this function shows a parabolic behavior. Therefore, function $h_{asr}(j)$ penalizes partitions where the ASR of avatars higher than 250 ms.

Finally, the term $n_m(i)$ is a function of the number of avatars that should be migrated in order to obtain a given partition. This function is also composed of two sections. Section from the zero value to one third of the existing avatars shows a linear behavior, as shown in Figure 1(c). From one third up, this function also show a parabolic behavior. In this way, this function avoids partitions that only can be obtained by migrating more than 30% of avatars. are migrated.

Thus, the QoS problem in DVE system is reduced to find the minimum value of f_{QoS}. Because of the high complexity of this problem, labelled as NP-hard in other

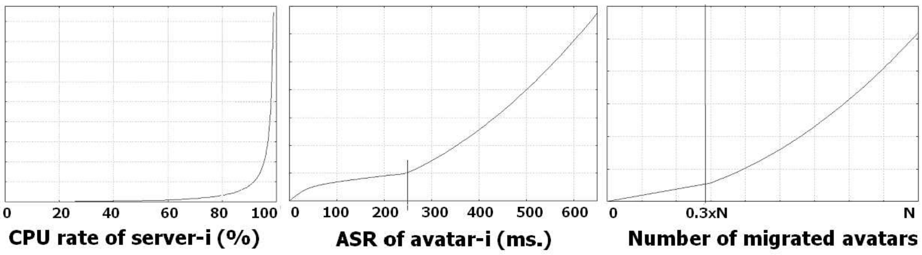

Fig. 1. Behavior of constraints in f_{QoS} for a) CPU utilization b) ASR and and c) migrations

systems [18], we propose two different approaches based on metaheuristic procedures. These approaches solve the problem in a domain composed by n^s different feasible solutions.

3 Heuristic adaptation for QoS problem in DVE systems

Metaheuristic strategies are widely considered as one of the most practical approaches for highly complex problems. A wide range of different problems have been solved using these strategies. Moreover, metaheuristics have been used in DVE systems for solving the partitioning problem [9]. In this section, we present the implementation and tuning of two different heuristics, based on simulated annealing and GRASP, for solving the QoS problem in DVE systems.

3.1 Simulated Annealing

Simulated Annealing (SA) is a stochastic metaheuristic applicable to arbitrary combinatorial optimization problems [3]. SA is a randomized local search strategy which is able to perform climbing moves. In this sense, it can also escape from local minima and find solutions which are much better than those of pure local search.

SA has been used in a wide range of problems [3, 9, 13]. This method models system temperature as the probability of accepting a worsening result. SA starts with a high system temperature, and in each iteration system temperature is decreased. In this way, SA can leave local minima by accepting worsening results at intermediate stages. The search method ends when either the number of iterations finishes or system temperature is so low that accepting worsening results is practically impossible.

The proposed implementation of SA method when applied to solving the QoS problem in DVE systems deals with *boarder avatars*. Two avatars (a_i and a_j) are boarder avatars if they are assigned to different servers (s_r and s_x) and their AOIs intersect. The assignment of boarder avatars is critical, and it allows to obtain partitions with low levels of h_{asr}. An iteration of SA consists of randomly selecting two different boarder avatars and randomly performing one of these three actions: exchanging the assignment of servers, both avatars server s_x or assigning both avatars to server s_r. Once this action is performed, the quality function for the resulting partition is computed. If the resulting value of f_{QoS} is higher than the previous one plus a threshold T, that change is rejected. Otherwise, it is accepted (the search method must decrease the value of the quality function f_{QoS} associated with each assignment). The threshold T used in each iteration i of the search depends on the rate of temperature decreasing R, and it is defined as

$$T = R - \left(\frac{R \times i}{N}\right) \quad (2)$$

where N determines the finishing condition of the search. When N iterations are performed without decreasing the quality function f_{QoS}, then the search finishes.

As literature shows ([3], [9]) the two key issues for properly tuning this heuristic search method are the number of iterations N and the temperature decreasing rate R. Figure 2 shows the tuning of SA method. The graphic on the left shows the f_{QoS} values obtained with SA method when different number of iterations are performed, and the graphic on the right shows the f_{QoS} values obtained with SA method when different cooling rates are considered. These results have been obtained for a DVE system composed of 700 avatars assigned to 10 servers.

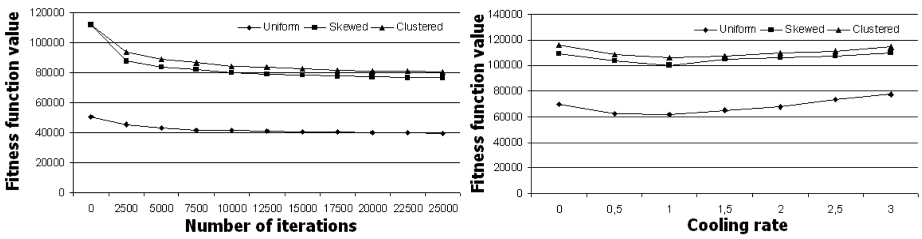

Fig. 2. Variation of quality function f_{QoS} for different number of iterations and cooling rates

Figure 2-left shows that f_{QoS} decreases as the number of iterations increases, until value of 7500 iterations is reached. From this point this value remains constant or decreases very slightly. This behavior is due to the impossibility of finding better search paths even when more iterations are performed. On other hand, Figure 2-right shows that f_{QoS} decreases until a value of 1% is reached. From this point quality function also increases, since cooling rate is too high and the search method accepts too many worsening solutions. Therefore, for this DVE configuration, the number of iterations and cooling rate selected for SA method has been 7500 and 1% respectively.

3.2 Greedy Randomized Adaptive Search (GRASP)

GRASP is a constructive technique designed as a multi-start heuristic for combinatorial problems [2]. It has been shown to quickly produce good quality solutions for a wide variety of problems [14],[9].

The proposed implementation of GRASP method for solving the QoS problem in DVE systems starts with an initial partition. This initial partition is provided by a load balancing technique [12]. Therefore, we ensure that the initial partition is well-balanced and all servers have a percentage of CPU utilization as low as possible. At this point, GRASP method will be used for searching a near optimal partition that provides QoS to the maximum number of avatars migrating the minimum number of avatars.

The first step in our GRASP implementation consists of sorting the avatars whose messages show a round-trip delay higher than 250 ms. (those avatars not provided with QoS) by their presence factor f_p. We define the *presence factor ($f_p(i)$)* of avatar i [11] as the number of avatars in whose AOI avatar i appears. The idea is to provide QoS to those avatars that require the least system efforts. The avatars with the higher presence factor should receive updating messages from a lot of avatars, and it will send messages to a lot of avatars. Therefore, migrating these avatars to the proper server can decrease the round trip-delay for the messages sent by a lot of avatars (can provide QoS with the least effort). Moreover, if GRASP method focuses only on this kind of avatars then it will significantly decrease the term $\sum_{j=0}^{n} h_{asr}(j)$ in f_{QoS} function without significantly increasing the term n_m.

The first c elements in the sorted list of avatars (from a population of n avatars) are denoted as *critical avatars*. GRASP method considers critical avatars as non-assigned avatars, and they will be assigned by GRASP method to a server in such a way that QoS is provided to them. The rest of the n avatars (denoted as the e *easy avatars*, where $n = c + e$) will not be re-assigned, and they will remain assigned to the same server where they were initially assigned to. The assignment of each of the c critical avatars is obtained in each of the iterations of the GRASP method. The number of iterations (the number of re-assigned avatars) is the only parameter to be tuned for GRASP method. If c value is set too low, then only a few avatars will be provided with QoS. Also, if c value is set too high (trying to provide too many avatars with QoS) then GRASP method will not be able to find a partition fulfilling all the requirements for all avatars, and it will take a long time for providing bad partitions.

Each iteration of GRASP method consists of two steps: construction and local search. The *construction* phase builds a feasible solution choosing one critical avatar by iteration, and the *local search* derives this temporal solution following a neighborhood criterion. Concretely, we propose an implementation where each iteration i performs the next steps:

- Constructive phase:
  1. The first avatar in the sorted lists of critical avatar is randomly assigned to a server, and the quality function f_{QoS} is computed for the partition composed of the e easy avatars plus this new avatar assigned to that server.
  2. The previous step is repeated with the remaining $c-1$ avatars in the sorted list of critical avatars. This step will provide a list of $c-i+1$ different critical avatars each of them randomly assigned to a server and each one with a f_{QoS} value. This list will be denoted as the *list of candidate avatars for iteration i (LCA(i))*. LCA(i) will have a size equal to $c-i+1$, that is, the number of non-assigned avatars for iteration i. Each element in LCA(i) will have the form (*non-assigned border avatar, server, resulting* f_{QoS}.

– Local search phase:
  1. LCA(i)list is sorted (using Quick-sort algorithm) by the resulting cost f_{QoS} in ascendent order, and then is reduced to its top quartile. An avatar j in this reduced LCA(i) list is randomly selected for local search.
  2. Local search on avatar j consists of looking of non-assigned critical avatars that are neighbors of avatar j. If any avatar k exists in LCA(i) and k is a neighbor avatar of j, then all possible assignments of avatar k to the different servers are considered. For each of them, the quality function f_{QoS} is computed for the partition composed of the $e + i - 1$ assigned avatars plus this new avatar k assigned to that server.
  3. If there not exists any avatar k that is a neighbor avatar of avatar $j$and at the same time appears in LCA(i), then the solution of iteration i consists of assigning avatar j to the server where avatar j is assigned in LCA(i).
  4. If any avatar k exists, then all assignments of avatar k (and any other existing avatar in the previous step) to the different servers and their resulting f_{QoS} are sorted in ascending order to form the *list of local search assignments LLSA(i)*, together with avatar j and its assignment in LCA(i). The first element in LLSA(i) (the assignment of avatar j or its neighbors with the minor f_{QoS} value) is selected as the solution of GRASP for iteration i

The main parameter to be tuned in GRASP method is the number of avatars c that the initial partition must leave unassigned. Figure 3 shows the tuning of GRASP method when this value is varied. The graphic on the left shows the values of f_{QoS} obtained for different values of this parameter, and the graphic on the right shows different execution times required for performing the search when this parameter is varied. The results shown in these figures have been obtained from a DVE system composed of 700 avatars and 10 servers.

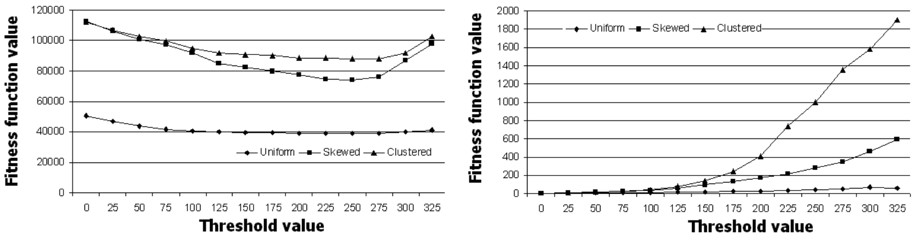

Fig. 3. Variation of quality function f_{QoS} and execution times for different values of c

This figure shows that as the number of critical avatars increases the quality of the provided partitions increases and so does the required execution time. In the case of this DVE configuration, 125 iterations has been chosen in order to obtain good quality solutions without spending too much execution time. A higher number of iterations would require too much execution time and it would not provide significantly better solutions.

Finally, if this value is excessively increased (more than 275 iterations) then f_{QoS} even increases. This behavior is produced by the greedy component of the algorithm that offers suboptimal solutions within the construction phase.

4 Performance Evaluation

In this section, we present the performance evaluation of the heuristics described in the previous section when they are used for solving the QoS problem in DVE system. We have empirically tuned SA and GRASP search methods in two different DVE configurations, denoted as MEDIUM1 and MEDIUM2. Empirical results have been obtained from our DVE simulation tool described in [10], [11] and [12]. This tool models the behavior of a generic DVE system with a network server architecture on a real network of heterogeneous computers. MEDIUM1 is composed by 250 avatars and 3 servers, and MEDIUM 2 is composed by 700 and 10 servers. In both configurations uniform, skewed and clustered distribution of avatars have been simulated. However, due to space limitations, we present here the result for MEDIUM2 configuration. The results obtained for MEDIUM1 configuration were very similar.

Table1 shows the performance evaluation results obtained for MEDIUM2 configuration when the proposed method is simulated under different initial distributions of avatars in the virtual world. For comparison purposes, we have evaluated the DVE performance obtained with ALB method [12] and the performance obtained with the proposed methods, SA and GRASP. For each distribution of avatars, table 1 contains three columns, one for each partitioning method. These columns show the performance provided by each method. The first nine rows, labelled with Sx, show the percentage of CPU utilization reached in each DVE server with each partitioning method. Last but two row shows the number of avatars in the partition provided by each partitioning method whose messages showed an average round trip-delay lower than 250ms. That is, this row shows the number of avatars provided with QoS by each partitioning method. Next row shows the number of migrated avatars $\Gamma(P_0)$ in the whole simulation. Finally, last row shows the execution time required by each partitioning method to provide the final partition tested in the simulation.

Table 1 shows that for large DVE configurations the proposed methods allows to provide QoS to a significatively higher amount of avatars. Thus, for example, for a uniform distribution of avatars in the virtual world GRASP method is able to increase about a 22% the amount of avatars provided with QoS with respect of ALB method, while maintaining all servers far from reaching 95% of CPU utilization and also migrating less than one third of the population of avatars. In the case of a clustered distribution of avatars, GRASP is able to increase in about 50% the amount of avatars provided with QoS, in relation to ALB method. In the case of an skewed distribution of avatars, the increasing is about a 300%. These results fully validate the proposed method as a valid approach for providing QoS to the highest number of avatars as possible.

It is also worth mention the great differences in absolute terms that the same method provides for the different distributions of avatars, particularly between the uniform distribution of avatars and the other two distributions. This difference is due to the differences in the presence factor of avatars between the distributions. Since most of the

	Uniform distribution			Skewed distribution			Clustered distribution		
	ALB	SA	GRASP	ALB	SA	GRASP	ALB	ALB	GRASP
S0 (%)	25	23	21	67	85	42	88	75	87
S1 (%)	28	30	29	58	46	78	71	69	61
S2 (%)	18	19	35	64	69	56	93	74	76
S3 (%)	17	14	14	63	48	58	80	75	76
S4 (%)	20	20	8	66	71	67	69	77	77
S5 (%)	17	18	12	89	67	51	74	76	74
S6 (%)	16	13	41	36	67	58	77	78	61
S7 (%)	22	23	9	55	62	84	71	79	84
S8 (%)	17	21	16	78	59	87	74	77	85
S9 (%)	17	16	12	65	66	59	67	81	80
QoS	544	629	663	92	232	270	256	363	388
$\Gamma(P_0)$	-	97	101	-	209	204	-	201	184
$T_{exe}(s.)$	-	7.6	5.3	-	20.1	18.8	-	38.6	34.3

Table 1. Results for a MEDIUM2 DVE configuration

avatars are very close each other in both skewed and clustered distributions, then most of avatars are highly connected with other avatars. As a consequence, it is more difficult to distribute these avatars between more servers while still providing QoS to all of them.

Finally, table 1 also shows similar execution times and number of migrated avatars for both proposed methods. However, for all three distributions of avatars GRASP method obtains partitions with less numbers of avatars without QoS than SA method. The reason of this behavior is the fast and powerful mechanism of GRASP approach in order to explore huge domains of solutions.

5 Conclusions and future work

Traditionally, DVE(Distributed Virtual Environments) systems have addressed the QoS of clients with graphical approaches. However these approaches, based on multiresolution models or compensation mechanisms, do not take into account the non-linear behavior of DVE systems with the workload they support.

In this paper, we have proposed the implementation, tuning and comparison study of two different search methods, based on Simulated Annealing (SA) and GRASP, in order to solve the QoS problem in DVE systems. These approaches models the QoS problem as an evaluation function to be minimized when solving the partitioning problem.

Performance evaluation results shows that the proposed methods can be considered as a good mechanism in order to offer QoS to avatars in a DVE system. These results show similar performance, in terms of quality of the provided solutions and execution times, for both methods and for a small DVE configuration (MEDIUM1). For large DVE configurations (MEDIUM2) GRASP method manages to provide more avatars with QoS than SA method.

As future work to be done, we plan to design a parallel implementation of GRASP method that can take advantage of the DVE servers where it will be performed. This new design will be based on a master-slave configuration and will be implemented in conjunction with a post-optimization path-relinking procedure.

References

1. Z. Choukair, D. Retailleau, M. Hellstrom. “Environment for Performing Collaborative Distributed Virtual Environments with QoS”,in Proceedings of the IEEE Conference *IC-PADS’00*, pp.111-118, Iwate, Japan. July, 2000.
2. T. A. Feo and M. G. C. Resende, “Greedy Randomized Adaptive Search Procedures”, *Journal of Global Optimization*, No.1, pp.109-136 1995.
3. P.V. Laarhoven and E. Aarts, “Simulated annealing: Theory and applications”, *Reidel Publication*, Dordrecht, Holland, 1987.
4. K. Lee and D., “A Scalable Dynamic Load Distribution Scheme for Multi-Server Distributed Virtual Environment Systems With Highly-Skewed User Distribution”, *Proceedings of the 10th ACM Conference VRST-2003*, October 2003, pp.160-168, Osaka, Japan.
5. M. Lewis and J. Jacboson, “Game Engines in Scientific Research”, in *Communications of the ACM*, Vol. 45, No. 1, pp. 27-31, January 2002.
6. Jonh C.S. Lui and M.F. Chan, “An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems”, *IEEE Transactions TPDS*, Vol.13, No.3, pp. 193-211. March 2002.
7. D.C.Miller and J.A. Thorpe, “SIMNET: The advent of simulator networking”, in *Proceedings of the IEEE*, Vol. 83, No.8, pp. 1114-1123. August, 1995.
8. T. Nitta, K. Fujita and S. Cono, “An Application Of Distributed Virtual Environment To Foreign Language”, in *Proceedings of FIE’2000. IEEE Education Society.* Kansas City, Missouri, October 2000.
9. P. Morillo, M. Fernández and J. M. Orduña, “A Comparison Study of Modern Heuristics for Solving the Partitioning Problem in Distributed Virtual Environment Systems”, in *International Conference in Computational Science and Its Applications (ICCSA’ 2003)*, volume 2669 of Springer LNCS, pp. 458-467, Montreal, Canada. May 2003.
10. P. Morillo, J.M. Orduña, M. Fernández and J. Duato, “On the Characterization of Distributed Virtual Environment Systems”, *Proceedings of European Conference on Parallel Processing (Euro-Par’ 2003)*, Klagenfurt, Austria. August, 2003.
11. P. Morillo, M. Fernández and J. M. Orduña, “On the Characterization of Avatars in Distributed Virtual Worlds”, in *Annual Conference of the European Association for Computer Graphics (EUROGRAPHICS’ 2003)*, pp. 215-129, Granada, Spain. September, 2003.
12. P. Morillo, J.M. Orduña, M. Fernández and J. Duato, “An Adaptive Load Balancing Technique for Distributed Virtual Environment Systems”, *Proceedings of the 15th IASTED International PDCS-03*, pp.256-261, California, USA. November, 2003.
13. C.A.S. Oliveira, D. Paolini and P.M. Pardalos, “A Randomized Algorithm for Minimizing User Disturbance Due to Changes in Cellular Technology”, *Proceedings of the International Conference CCCT-03*, Vol. 5, pp. 45-50, Orlando, Florida. July, 2003.
14. M.G.C. Resende and C.C. Ribeiro, “A GRASP with path-relinking for privatevirtual circuit routing”, *Networks*, No. 41, pp. 104114, 2003.
15. J.M. Salles, Ricardo Galli, A. C. Almeida et al, “mWorld: A Multiuser 3D Virtual Environment”, in *IEEE Computer Graphics*, Vol. 17, No. 2, pp.55-65. March-April 1997.
16. S. Singhal and M. Zyda, *Networked Virtual Environments* (ACM Press, New York, 1999).
17. T. Henderson and S. Bhatti, “Networked games: a QoS-sensitive application for QoS-insensitive users?”, in *Proceedings of the ACM Conference SIGCOMM 2003*, pp. 141-147, Karlsruhe, Germany. August 2003.
18. X. Yuan, “Heuristic Algorithms for Multi-Constrained Quality of Service Routing”, *IEEE/ACM Transactions on Networking*, Vol. 10, No. 2, pp. 244-256. April 2002.