

A GRASP-based algorithm for solving DVE partitioning problem

Pedro Morillo, Marcos Fernández

Instituto de Robótica. Universidad de Valencia.(Spain)

{Pedro.Morillo,Marcos.Fernandez}@uv.es

Abstract

Graphic cards performance increase and fast Internet connections are popularising Networked Virtual Environments. This immature paradigm of real-time applications has still to solve a set of internal problems associated with the heterogeneity in client-side hardware, network bandwidth and graphic application requirements efficiently. The problem of properly assigning clients (3D avatars) to servers in a server-network architecture is one of them. This paper describes a new solution to this NP-complete task, called partitioning problem. This solution is based on GRASP, a multi-start metaheuristic for combinatorial problems. The results of the performed experiments, compared to the main reference in this field for a proof-of-concept system, show better clustering solutions with less execution times.

1. Introduction

Nowadays, professional graphic cards offer a very good frame-rate for rendering complex 3D scenes in real time and are becoming cheaper and widely spread. In turn, high-speed connections to Internet are becoming common in general-purpose computers. These factors make possible the development of Distributed Virtual Environments (DVE). These systems allow a collection of users, working on several computers within a network, to run an interactive a 3D-graphic interface program that simulates the experience of immersion in a virtual world. This is achieved by rendering images of the environment as if they were perceived by the user's simulated viewpoint. Each user is represented by an entity, usually humanoid, called avatar in the shared virtual environment. The avatar state is controlled by the user's input. Because of these systems support visual interactions between multiple users in the shared 3D-virtual environment, every change in the simulation has to be propagated to the rest of simulation participant. DVE systems are nowadays used in a wide variety of applications ([24]), such as collaborative design ([22]), civil and military ([19]) distributed training simulations, distributed and interactive e-learning ([20]) and multiplayer games ([13][1]).

There are two relevant concepts to be solved when a DVE is designed. First of all, the concept of heterogeneity appears in different ways:

- Each avatar, linked to a client's computer in the simulation, does not have a restricted installed hardware; since a wide range of resources such as processor speed, memory size, and graphic card technology are found in a conventional or professional client.
- Internet connection type among all the simulation nodes: the shared medium topologies that range from Ethernet or Fast-Ethernet in LAN to ISDN, fiber-optic or ATM in WAN, are simultaneously allowed in some DVE.
- Avatars communication rate: These applications are based on a message-passing paradigm. Their data transmission rate needs to be controlled and it is quite different in a collaborative 3D environment or in a 3D virtual military battle.

The other concept is associated with the inherent problems defined in these technologies. Each of them, produces an important and open research field:

- *Data Model.* Detailed in [18], describes some conceivable ways of distributing persistent or semi-persistent data in a DVE. Principally, data can be managed in a replicated, in a shared or in a distributed methodology.
- *Communication Model.* Bandwidth determines the size and performance of a DVE; the world behaviour is related to the way that all the scene clients are connected. Broadcast, peer-to-peer or unicast schemes define different network latency values for exchanging information between two avatars.
- *View Consistency.* This problem has already been defined in other computer science fields such as database management ([3]). In a DVE system, this problem consists of ensuring that all the avatars sharing a virtual space with common objects have the same local vision of them.
- *Message Traffic Reduction.* Keeping a low message volume lets DVE scale the scene number of avatars efficiently. Traditionally, techniques such as dead-reckoning described in [24] offered some level of independence to the avatars. With network support,

broadcast solutions ([7]) decrease the number of messages used to keep a consistent state of the system.

Most part of these issues is related to a key aspect called partitioning problem. Defined by Lui-Chan ([14]) for DVE, this problem establishes a logical connection between assignment of clients and network performance in a network-server architecture for a real-time 3D simulation. This problem and the previous solutions to it, such as [14], [17], [24] and [25] are described in Section II. The rest of the paper is organized as follows: Section III presents a solution based on GRASP (Greedy Randomized Adaptive Search Procedures), a randomized heuristic that has produced high quality solutions for a wide range of combinatorial optimization problems.

Section IV shows the results obtained with a shared visualization program. This application was built incorporating the GRASP-based algorithm described in this paper on a network-server architecture. The result obtained by a GRASP-based algorithm demonstrates that this method reaches better grouping solutions by spending lower execution times.

Finally, in section V the contributions of this paper are summarized and future research work is suggested.

2. Partitioning problem. State of the art

In spite of the heterogeneity described above, modern DVE share some common features. In the whole set of communication models presented in many papers ([24][18][14]), network-server architectures are becoming in a de-facto standard. Parallel or network-based distributed technologies developed on DVE, such as high performance web-server management are on the way to get the same conclusions ([12]). In these architectures, denoted as hierarchical by others researchers ([8]), the simulation control is organized by interconnected servers. Multiplatform clients are forced to be allocated to one of these servers. When an avatar is updated by a client one updated message is sent to its attached server. Depending on the state of the simulation this server can resend this message to its allocated avatars or, on the contrary, to other servers. In order to avoid a message outburst when the number of clients increases, areas of influence (AOI in [24], locales in [2] and aura in [12]) are defined for each one of the avatars. In this way, messages are only propagated from one avatar to the avatars that fall into its AOI. Depending on which is the destination server, two kinds of communication messages are defined in a DVE scheme (see figure 1). Fast inner-server communications when both the sender and receiver are allocated in the same server, and long inter-server otherwise. With this approach the traffic distribution load is moved out of the

clients into servers. For each local update, clients only send one message to a server and receive messages from a server in order to update all neighbours of the avatars.

Therefore, they must complete very little workload, storage or messaging in order to maintain consistent state among many avatars in a large DVE.

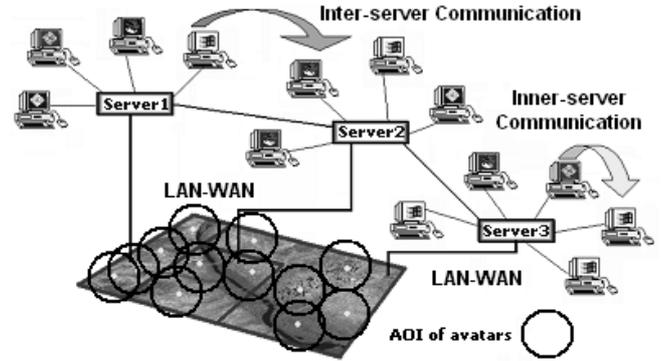


Figure 1. Communication model for a basic DVE

For these systems Lui-Chan have redefined in [14] the partitioning problem. It consists of efficiently assigning the avatars of the simulation among different servers in the system. The partitioning problem may seriously affect the overall performance of the DVE system, since it determines not only the workload each server is assigned to, but also the inter-server communication requirements (and, therefore, the network traffic).

In order to achieve this goal, two factors must be considered in an optimal design of a DVE:

On the one hand (C_p^W), every server should share the workload of the clients in a balanced way. Logically, it is not necessary for the servers to be identical, but their pondered hardware features have to support the same number of clients. (For example, in a DVE where 150 avatars are running on a network of 10 identical Pentium IV computers, each of them should allocate about 15 avatars. On the contrary, an heterogeneous network formed by 4 Pentium IV PCs, 2 Silicon Graphics Reality Monster and 3 IBM eServer could host 10, 15 or 20 avatars, respectively).

On the other hand (C_p^L), in order to minimize the overall inter-server traffic, neighbouring avatars should be assigned to the same server. Due to topological reasons, it is not possible for the majority of the cases to accomplish this condition; therefore, it is necessary to find a suitable topological grouping.

According to these two parameters a cost evaluation function is offered in equation 1 by [14] to estimate the goodness of a client partition:

$$C_\rho = W_1 C_p^W + W_2 C_p^L \text{ such as } W_1 + W_2 = 1 \quad (1)$$

W_1 and W_2 denote the relative importance of the computational workload and the inter-server communication cost above mentioned. In the general formal case W_1 is equal to W_2 and equal to 0.5. It is evident that, when a DVE is working on a high performance network, the quotient W_1/W_2 may be much bigger than one. Quite the opposite for the DVE working in a slow-shared network or Internet, where large messages latencies make the quotient to be close to zero. During the simulation play, the algorithm, which is continuously regrouping clients, must be run several times. The advisable execution rate is published in [16].

In [17] the authors demonstrate that the partitioning problem is NP-complete and offer a platform test to scientific community, where every approximation to the problem can be checked and compared. Furthermore, a refinement of their initial algorithm is proposed and a parallelization based on the amount of avatar criteria is presented. Theoretical bases to this algorithm are described in [15]. This refinement divides the algorithm into three parts: a *recursive bisection partitioning*, which carries out an initial allocation based on a cell criterion, a *layering partitioning* which maximizes workload balance, and a *communication refinement partitioning* which maximizes inner-server communication. Depending on the DVE size or state, some of them can be repeated in the same execution.

There are other approaches, with different denominations, to this problem published in [25][26]. In [25] an approach groups the avatars following regular distributions. In order to ensure good graphic performances to the clients, this algorithm generates a regular number of this distribution equal to the simulation server number. This solution do not obtain good results when avatars are located in a non-uniform distribution. Other approach ([26]) rejects the dynamic concepts associated to avatars such as aura, AOI or locale. This technique divides the 3D virtual scene into a regular grid. For each cell, a multicast group is created, so that avatars sharing a cell are assigned to the same server saving a lot of messages and sharing multicast packets. In spite it was a quick and determinist solution, this static classification performs badly when avatars are grouped in zones. The server that controls this zone is massively charged and the avatars in frontiers are going up the global cost.

3. A new approach

The solutions already commented for the partitioning problem share the same feature: they are custom solutions that are defined like *ad-hoc* heuristics to solve the problem. The algorithm presented in this paper is based on a modern metaheuristic called GRASP.

These kinds of common heuristics, oriented to solve general NP-complete problems and described in [27], are

plain procedures usually based on the common sense. They are supposed to obtain a good solution in a short period of time, but it is not necessary the best.

Usually, when searching for solutions for problems, some results are better than others. Each point in a solution domain is a feasible solution, which has a goodness value depending on the adaptation to the problem. This value is calculated, in polynomial time, thanks to an evaluation function. Searching a solution is equivalent to find a minimum value in the solution domain. In this point, we have connected DVE problem formulation in the way that those metaheuristics work. If the cost function is used as the evaluation function to the partitioning problem, then the process is reduced to obtain a good minimum for C_p , the whose associated solution is a high-quality grouping.

8.1. A GRASP-based solution

In the set of modern metaheuristics, a constructive technique, called **GRASP** has been chosen. GRASP was introduced by Feo and Resende in [6] as a multi-start metaheuristic for combinatorial problems.

A GRASP solution is an iterative process consisting of two steps: construction and local search. The construction phase builds a feasible solution choosing one element by iteration. This choice is achieved by a greedy function. This function measures non-classified elements choosing one of them in relation to an evaluation function. The element chosen is not necessarily the best in the ranking of the obtained elements. A subset of elements is created from the best measures and a random component is elected from this subset. In the search phase, the neighbourhood of the chosen element is analysed in order to find the best local minimum.

In spite of the contributed newness, some solutions to NP-complete problems or very complex processes based on this technique had been appearing in last years. In [4] a GRASP solution to the plant location problem has been presented. For the Feedback Vertex Set Problem, in [21], an approximation has been published. MOCO problems find a good solution in [9], and a very good elucidation has been shown for the incapacitated location problem in [10]. GRASP implementations are very robust and it is not possible to find examples where the method performs badly. For a good adaptation to the DVE partitioning problem it is necessary to take into account the following problem: GRASP bases his efficiency on a exhaustive sampling and a massive checking over the solutions tree. Due to this reason, a full GRASP-based solution implementation would lead to an intractable execution time in the solution finding. To achieve optimal and rational goals a reduction technique is presented in this paper.

3.2. A density-based pre-solution

In order to avoid this unfeasibility, a method composed of two phases is proposed. In the first phase a fast pre-solution forces an avatar-to-server pre-assignment, grouping sets of elements. This first approximation is based on a brute force mechanism and it does not take into account any special case or refinement. Also, as already mentioned, the approximation has to perform good allocations for several determined avatars. In the second phase, an adapted GRASP method is applied.

Because of the reasons given, the first initial assignment had to obtain a fast good allocation to some avatars, while some others are not in the proper location. In order to prepare the execution of the second part of the method, the avatars that had been wrongly allocated have to be determined.

For this first phase, several algorithms based on different computer science methodologies are tested. However, in this paper a density-based pre-solution is presented. This algorithm is near the general idea of the graph theoretic method published in [5]. The new algorithm divides the virtual 3D scene into square sections. Each section is labelled with a number indicating the number of avatars within it, called “*na*”.

The sections are sorted out (using QuickSort) by the *na* number and the biggest *S* is chosen. *S* is the number of servers in the simulation. For each one of these massively settled sections mass-centre of the avatars is calculated. The mass-centres (*mc*) are associated to the different servers in the process assignment. Using rotary turns each *mc* chooses its nearest free avatar. Every choice includes this marked avatar in the own server. Gradually, this selection will create concentric figures over the mass-centres. Since the search for avatars is driven by server (not by avatar) the workload balance is maximum.

The general idea, for *n* avatars and *S* servers has the following steps for the proposed pre-assignment process algorithm:

Algorithm 1. Density pre-assignment Algorithm

1. **begin**
2. Divide virtual 3D scene map in square sections
3. Obtain *na* for every section
4. sorted_list=Quicksort (sections, na)
5. Choose the *S* biggest elements in sorted_list
6. **from** *i*=0 to *S* do
 representant [*i*]= ObtainMC (*i*)
7. pivot = 0, elect = -1, ncentr=0
8. $\forall_{i=0}^n$ assigned[*i*] = 0
9. **from** *i*=0 to *n* do{
10. elect=-1, min_dis= ∞
11. **from** *j*=0 to *n* do{

12. if (assigned[*j*]=0){
13. dist_tmp= D^1 (avatar[*j*],represent([ncentr]))
14. if (dist_tmp<min_dis){
15. pivot=*j*
16. min_dis=dist_tmp
17. }
18. }
19. }
20. avatar[pivot] = ncentr
21. Increment ncentr module *S*.
22. }
23. **end**

With this algorithm, the behaviour of the avatar-to-server assignment has an important feature. It is related to the order of solution calculation.

As the new avatar assignment is being created concentric figures are painted over the virtual 3D scene. These figures are composed of avatars and are usually circumferences. Given that, due to the way that C_p cost is calculated, the best partitioning problem solutions, which minimize inter-server communications, C_p^L , are characterized by surrounding avatars with neighbours what belong to the same avatar. This fact leads to scene avatars, being located far away or equidistant from a mass-centres subset, are critical. These nodes, called boarder-nodes, can present a deuce in the assignment process. One mistake in this procedure will raise the total cost of the solution C_p . In order to support the assignment of this critical avatars it is necessary to spend more resources. For this reason, non-assigned avatars are the inputs to the real GRASP allocation in the second phase. Figure 2 shows one execution for a density-based pre-solution where about sixty-six avatars are divided among three servers in a 3D virtual simulation

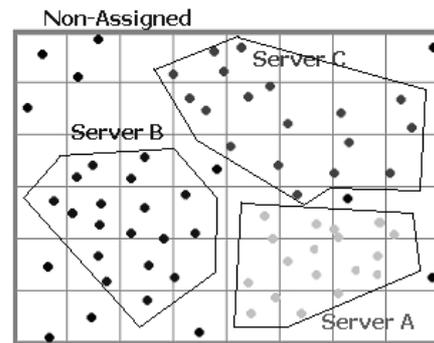


Figure 2. Result from a the density-based pre-solution

¹ Note: D is a function that obtains the Euclidean distance defined by two points (in a 2D or 3D space)

3.3. GRASP adaptation for complete solution

Second phase finds complete solutions adapting a GRASP method for these critical non-assigned avatars. Continuing the Feo and Resende mechanism referred to above ([6]), each iteration of the adapted GRASP process is going to choose an element for this non-assigned avatar subset. Also, in the same iteration a server allocation is associated to this element. To achieve this goal, a greedy function has been specified for the non-assigned elements in the cited construction phase.

For the partitioning problem the greedy procedure has been associated to the cost evaluation function. In this way, the cost increase is evaluated for each avatar. This cost, in terms of C_p , is obtained adding a new element to the already classified avatars. Different configurations, equals to the number of servers, can be assigned to the added avatar in the established solution. With the new sets of solutions, the additional cost is re-evaluated as section II describes.

For every free avatar (non assigned), when all assignments of servers are recalculated, a ranking list is obtained. This list, called *candidate list (CL)*, details one evaluation of the benefits derived from including each element in the built solution.

Each element of the list has the form {avatar number, server number, new total cost} and it must be sorted out in descendent order for the total cost criterion. Due to the size of the problem, this process is executed by QuickSort, on average, the fastest known sorting algorithm ([23]) for a large number of elements requiring only $\Theta(n \times \log n)$ steps. In order to obtain a better final result this list is reduced to its top quartile, discarding the rest of elements. The new list is called *reduced candidates list (RCL)*. One element of the RCL is chosen randomly from all the candidates. The chosen element can be other than the top element of the list. The final solution performance depends on the quality of the elements in RCL. In turn, the solution diversity depends on the set cardinality of the RCL.

The justification is as follows: the diversification in the search process relies on a controlled randomisation so that each GRASP iteration builds a different solution. Greedy constructions leads to solutions close to local optima, getting the local search quickly.

When a solution is selected from the RCL, that is, an avatar number with an attached server, the method starts the local search phase. In this step, an extensive search based on neighbourhood criteria looks for a global solution near of the intermediate result obtained previously from the RCL.

The appropriate neighbourhood choice consists in a search inside of the AOI of each avatar. When the avatar, that represents the new element to be added to the global solution, finds some avatars that had not been previously

assigned to a server, it tests them for all the different server configurations, recalculating anytime C_p , and keeping the best configuration which offers the minimum cost.

If the quality of intermediate solution obtained by the greedy and randomized procedure is high (close to the best) then the fast local search is fairly accelerated.

An approximation in pseudo-code to the implemented technique commented above could be represented as:

Algorithm 2. GRASP-based algorithm

```
1. begin
2. Make a density-based pre-solution(threshold)
3. nas = Detect non-assigned avatars
4. from i=0 to nas do{
5.   from j=0 to n do{
6.     from k=0 to S do{
7.       tmp_cost=TestSolution(j,k)
8.       AddToList(list,tmp_cost,j,k)
9.     }
10.  }
11. QuickSort(list)
12. ReduceToFirsQuartile(list)
13. ChooseRandomElement(list,avatar,server)
14. best_cost = tmp_cost =  $\infty$ 
15. from j=0 to AvatarsInAOI(avatar) do{
16.   from k=0 to S do{
17.     tmp_cost=TestSolution(j,k)
18.     AddToList(list,tmp_cost,j,k)
19.     if (tmp_cost < best_cost){
20.       best_cost = tmp_cost
21.       savej = j
22.       savek = k
23.     }
24.   }
25. }
26. AssingSolutionServer(savej,savek)
27. }
28. end
```

In an overall view, there is an important detail when the algorithm is going to be implemented over a DVE platform. It is based on finding out the connection avatars numbers (*can*). In a 3D virtual scene, for the technique presented in this paper, the value of “can” determines the number of avatars that must not be assigned by the density-based pre-solution and are introduced to the real GRASP algorithm to obtain the complete and suitable solution.

In other words, as it is described in the pseudo-code above, it is necessary to obtain an avatar threshold value that allows (line 2 in a GRASP-based algorithm) some avatars not to be labelled and be assigned by the rest of the algorithm in a feasible execution time. This threshold is discussed in the next section.

4. Results

As it was commented in section II an important contribution to the DVE partitioning problem is found in [17]. In this paper, a test field is offered to the research community to test and compare the proposed partitioning algorithms. This test defines two different virtual scene sizes where a paradigm must be checked.

The first of them, called SMALL, is composed by 13 avatars in a squared (divided in 4x4 cells) virtual scene; these avatars must be assigned into 3 servers. For the opposite, to scan all the wide diversity of DVE, a LARGE world is defined by 2500 avatars. They must be located into 8 servers in a square world of 25x25 cells. In both cases, 3 different avatar distributions have to be tested to verify 3D application dynamic features.

Uniform, skewed and three-focused clustered distributions are generated to test SMALL and LARGE worlds. In both worlds, after implementing and testing deeply these patterns, it is necessary to point out three features: a) an homogeneous bandwidth is considered for all connections in the system. For these tests, the algorithm behaviour is independent from the type (shared media, point-to-point...) and the bandwidth connection; b) all the servers in the simulation are identical (in technical features) and c) because of an existent relationship between the size of AOI and the cell area the dimensions of the 3D virtual scene are not given in Km. or meters. If an avatar has a diameter of AOI equal to D units, then, the scene cells that define the scene have an area of D^2 units. Keeping these specifications a test battery, made of four thousand different DVE, is thoroughly passed in a proof-of-concept application. Following this different avatars distribution patterns this program generates randomized elements over the 3D scene world. Every avatar, located in a random point of the scene, is also defined by other important value, which is going to simulate the workload avatar. This value has been defined as a real number in a range from 0 to 5. In turn, as it was previously mentioned, two avatars have a communication link if one of them is located inside of the other's AOI. This value is very important (such as workload) in the C_p calculation. For this simulation, when two avatars are close then a defined connection value binds them. This quantity, inversely proportional to avatars distance, has a range between 0 (avatars are separated near of $D/2$) and 5 (avatars together).

In this system, SMALL and LARGE populations are created and simulated. The previously described algorithms are measured over them. For test and development, hardware platform is formed by a PC cluster systems. The test platform is based on a PC-cluster, using 1.5Ghz Pentium IV, with 256 MB of RAM and a nVidia Geforce 2 MX- 400 graphic card as node for the cluster.

4.1. Results in SMALL virtual worlds

In order to present the problem size, it was possible, only in SMALL worlds, to reproduce an exhaust solution. Due to the problem size, in this case, a full solution tree scanning requires to explore 3^{13} (1.594.323) different solutions. Table 1 shows the computation time and the system cost C_p obtained for different distributions of avatars in a SMALL virtual world when three different algorithms are applied: the exhaustive algorithm, the Lui-Chan algorithm and the proposed algorithm. It is necessary to spend more than four second, in some cases, to ensure that the algorithm reaches the solution. In spite of this measure, it is very far of real-time specifications. A perfect approximation to other algorithms efficiency is offered by its C_p associated cost. It is important to take into account that this cost would never be decreased by any technique.

Table 1. Results in virtual Worlds (SMALL)

(a) Avatars in a Uniform Distribution

	Computation t(s)	System Cost C_p
Exhaustive Alg.	3.411	6.54
Lui-Chan Alg.	0.0009	6.56
Grasp-based Alg.	0.0002	6.92

(b) Avatars in a Skewed Distribution

	Computation t(s)	System Cost C_p
Exhaustive Alg.	3.843	7.04
Lui-Chan Alg.	0.0010	8.41
Grasp-based Alg.	0.0002	8.30

(c) Avatars in a Clustered Distribution

	Computation t(s)	System Cost C_p
Exhaustive Alg.	4.783	7.91
Lui-Chan Alg.	0.0011	10.56
Grasp-based Alg.	0.0003	10.01

If the GRASP-based approach presented in this paper is compared to the previous Lui-Chan work presented in [17] it is possible to observe a better time performance in the entire evaluation test.

In the accomplished experiments, only in uniform distributions, the Lui-Chan algorithm reduces the costs obtained by the GRASP proposal. Both values (differentiated a 5.2%: 6.92 from 6.56) are quite close to the best value (6.54) obtained in the exhaustive algorithm. That is, when the avatar's location is distributed under skewed or clustered distributions neither Lui-Chan nor the proposed GRASP-based implemented algorithms can decrease the cost of the system in order to match an exhaustive solution. In spite of this, the GRASP-based

algorithm obtains better solutions (8.30-8.41 C_p units gains in skewed and 10.01-10.56 in clustered distributions) in less time (1.0ms-0.2ms in skewed and 1.1ms-0.3ms for clustered distributions) than the implemented Lui-Chan approximation.

In these little worlds it has been possible to avoid the initial pre-solution phase. In order to complete the final solution, the little size of the problem has allowed to add avatars directly in thirteen GRASP-steps.

4.2. Results in LARGE virtual worlds

When the 3D world size grows substantially, there are some important changes in the algorithms performance. Table 2 represents the experimental results under uniform, skewed and clustered avatars distributions in a LARGE virtual scene.

Since there is a great amount of avatars that can be assigned to several servers, it is not feasible to obtain the optimal solution for every world scene. 8^{2500} must be scanned with an exhaustive algorithm to get the lower C_p cost.

Table 2. Results in virtual Worlds (LARGE)

(a) Avatars in a Uniform Distribution

	Computation t(s)	System Cost C_p
Lui-Chan Alg.	30.939	1637.04
Grasp-based Alg	6.622	1779.76

(b) Avatars in a Skewed Distribution

	Computation t(s)	System Cost C_p
Lui-Chan Alg.	32.176	3460.52
Grasp-based Alg	15.535	2883.84

(c) Avatars in a Clustered Distribution

	Computation t(s)	System Cost C_p
Lui-Chan Alg.	43.314	5903.80
Grasp-based Alg	26.704	5306.24

Contrary to SMALL experiments, a RBP step and a triple CR and LP phases carry out a LARGE Lui-Chan execution. In Table 2 presented above the effectiveness of the GRASP-based algorithm is shown. Comparing to Lui-Chan implementation, in the solution proposed the computational time is diminished for all the checked cases in a 78.59, 51.71 and 38.24% for uniform, skewed and clustered distributions, respectively. In terms of the cost estimation, only in the case of uniform distributions the Lui-Chan method performs better than the GRASP method (8.72%).

For the rest of the patterns, when this perfect situation changes (logically, it is the normal case) to clustered or skewed distribution shapes, this implementation obtains better computational costs than the ones obtained by the Lui-Chan solution. The reduction achieved is about 16.66% for skewed and 10.12% for clustered distributions in the tested experiments.

Finally, when the size of the problem grows it is necessary to accomplish an optimal election for the threshold value commented above. Figure III shows the results in this tuning phase in order to compose an intermediate solution for allocating 2500 avatars located in a uniform distribution..

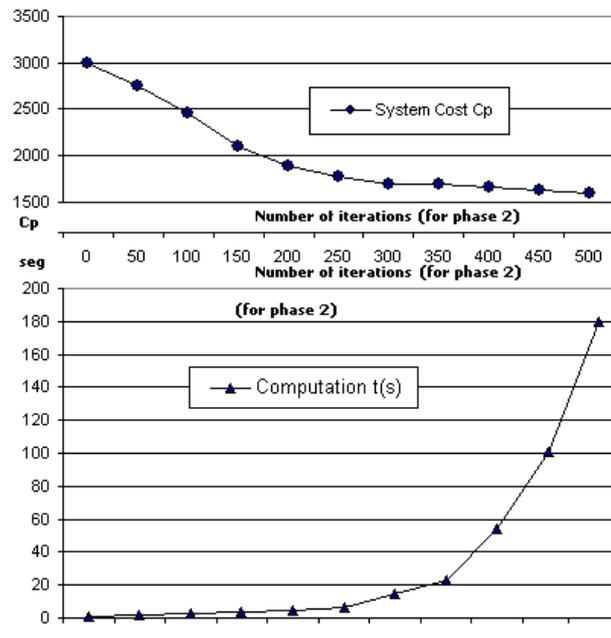


Figure 3. Variation of the performance measures for different threshold values

A logical threshold value had been obtained by the control of two parameters: the execution of the complete algorithm and the C_p cost of the final solution. In the case of LARGE worlds, 250 iterations for the GRASP algorithm obtain low cost solutions, spending close to 6.5 seconds to complete a full solution. It is important to note that values bigger than this number of iterations use too much execution time and do not reach significant solutions.

4. Conclusions and future work

Distributed Virtual Environments (DVE) systems are becoming an important research area in parallel and distributed processing field. Inherent heterogeneity

appears in several points of this architecture and controlling a group of clients (avatars) interacting in shared 3D world is not a trivial task.

In this paper, a new method to solve the problem of efficient client allocation for simulation servers is presented. Instead of developing an ad-hoc technique for this NP-complete problem, as other groups researches, a metaheuristic method proposed for combinatory mathematics, has been adapted to the problem specifications.

The experiments performed in the proof-of-concept system demonstrate that the proposed algorithm reduces the Lui-Chan solution significantly, the most important reference in this field for the same context test. These improvements are obtained in lower execution times and best solutions qualities. Therefore, the presented technique achieves better scalability in the large-scale DVE systems implementations.

As future work, the development of an efficient parallelization of the algorithm proposed is planned. This may be based on performing iterations of groups of avatars at each of the processors and collecting the best solution found.

6. References

- [1] M.Abrash "Quake's game engine: The big picture" Dr. Dobb's. Journal.Spring. Year 1997.
- [2] D.B.Anderson, J.W.Barrus, J.H.Howard, "Building multi-user interactive multimedia environments at MERL". IEEE Multimedia, 2(4):77-82, Winter 1995.
- [3] P.A.Berstein, V.Hadzilacos and N.Goodman."Concurrency, Control and Recovery in Database Systems". Addison-Wesley. 1997.
- [4] H. Delmaire, J.A. Díaz, E. Fernández, and M. Ortega. "Comparing new heuristics for the pure integer capacitated plant location problem". Technical Report DR97/10, Department of Statistics. UPC. Spain. 1997.
- [5] R.Duda, P.Hart, D.Stork. "Pattern Classification". Ed.Wiley Intescience Publication. 2000. p:567-580.
- [6] Thomas A. Feo, Mauricio G.C Resende "Greedy Randomized Adaptive Search Procedures". Journal of Global Optimization. 1995.
- [7] E.Frecon, M. Stenius. "DIVE: A scaleable network architecture for distributed virtual environments". Distributed Systems Eng., vol- 5. 1998.
- [8] T.A. Funkhouser. "Network Topologies for Scalable Multi-User Virtual Environments ".Technical Report Bell Laboratories 16. 1996.
- [9] F X. Gandibleux, D. Vancoppenolle, and D. Tuytens. "A first making use of GRASP for solving MOCO problems". Technical Report. Univ. of Valenciennes, France. 1998.
- [10] M.J.N. Gomes and J.B.C. da Silva. "An experimental evaluation of the GRASP metaheuristic applied to the capacitated location problem". Technical Report 004/99, Department of Statistics and Computation, State University of Ceara, Fortaleza, Ceara, Brazil. Year 1999.
- [11] F.C. Greenhlagh "Awareness -based commuication management in MASSIVE systems" Distributed Systems Engineering, vol 5, 1998.
- [12] J.C.Hu, I.Pyrali, D.C.Schmidt, "Measuring the Impact of Event Dispatching and Concurrency Models on Web Server Performance Over High-Speed Networks," 2nd.IEEE Global Internet Conference, November.1997.
- [13] Michael Lewis and Jeffrey Jacobson " Game Engines in Scientific Research ". Communications of the ACM., Vol 45. N°1 January 2002.
- [14] Jhon C.S. Lui, M.F.Chan, Oldfield K.Y "Dynamic Partitioning for a Distributed Virtual Environment" Department of Computer Science & Engineering. The Chinese University of Hong Kong. Year 1998.
- [15] John C.S. Lui, W.K. Lam, "General Methodology in Analysing the Performance of Parallel/Distributed Simulation under General Computational Graphs". Third International Conference on the numerical Solution of Markov Chain, September 6-10, 1999.
- [16] John C.S. Lui, M.F. Chan, K.Y. So, T.S. Tam. "Deriving An Optimal Synchronization Interval for Supporting View Consistency in a Distributed Virtual Environment". Fifth International Workshop on Multimedia Information Systems (MIS'99), pp, 99-106, October 21-23.1999.
- [17] Jonh C.S. Lui, M.F. Chan. "An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems". IEEE Transactions on Parallel and Distributed Systems, Vol. 13, N°3, March 2002.
- [18] Michael R. Macedonia. "A Taxonomy for Networked Virtual Environments". IEEE Multimedia, 4(1) 48-56. January-March 1997
- [19] D.C.Miller, J.A. Thorpe. "SIMNET: The advent of simulator networking ". Proceedings of the IEEE, 83(8):1114-1123. August 1995
- [20] Tohei Nitta, Kazuhiro Fujita, Sachio Cono "An Application Of Distributed Virtual Environment To Foreign Language". Kansas City, Missouri. IEEE Education Society. October 2000.
- [21] P.M. Pardalos, Tianbing Qian Mauricio G.C. Resende "A Greedy Randomized Adaptive Search Procedure For The Feedback Vertex Set Problem" J. Comb. Opt. Vol.2 (1999) pp.399-412 1999.
- [22] J.M.Salles Dias, Ricardo Galli, António Carlos Almeida, Carlos A. C. Belo, José Manuel Rebordão "mWorld: A Multiuser 3D Virtual Environment" IEEE Computer Graphics March-April 1997 (Vol. 17, No. 2)
- [23] R. Sedgewick, "Implementing quicksort programs", Communications of the ACM 21, 10, 847-857. Year 1978.
- [24] S.Singhal, and M.Zyda. "Networked Virtual Environments" ACM Press, New York, 1999.
- [25] P.T.Tam, "Communication Cost Optimization and Analysis in Distributed Virtual Environment", M.Phil. second term paper, Technical report RM1026-TR98-0412. Department of Computer Science & Engineering.The Chinese University of Hong Kong. 1998.
- [26] P. Barham, T.Paul "Exploiting Reality with Multicast Groups," IEEE Computer Graphics & Applications September 1995, pp.38-45.
- [27] S.H.Zanakis, J.R.Evans, "Heuristic Optimization: Why, when and how to use it". Interfaces, Vol. 11, n° 5, October. 1981.