

# Un nuevo enfoque genético para el problema del particionado en Entornos Virtuales Distribuidos (DVE)•

Pedro Morillo<sup>1</sup>, Pedro López<sup>2</sup>, Juan Manuel Orduña<sup>2</sup> y Marcos Fernández<sup>1</sup>

<sup>1</sup>Instituto de Robótica  
Universidad de Valencia  
Polígono de la Coma, s/n  
46980 Paterna (Valencia)  
Pedro.Morillo@uv.es

<sup>2</sup>Departamento de Informática  
Universidad de Valencia  
Av. Vicente Andrés Estelles, s/n  
46100 Burjassot (Valencia)  
Juan.Orduna@uv.es

**Abstract.** El problema del particionado es uno de los aspectos clave en el diseño de Entornos Virtuales Distribuidos (DVE). Este problema consiste en la asignación eficiente de los clientes de la simulación (avatares) a los servidores del sistema. Sobre este problema NP-completo la literatura en la materia recoge diferentes aproximaciones evolutivas, entre las cuales se encuentra una basada en algoritmos genéticos. En este trabajo analizamos el impacto de la baja diversidad que presenta la población inicial de este algoritmo. Además proponemos un sistema de generación de poblaciones iniciales de mayor calidad. Adicionalmente, presentamos un conjunto de nuevas técnicas de cruce muy adaptadas al problema. Junto al nuevo generador de poblaciones iniciales, estas técnicas definen un algoritmo genético que obtiene excelentes soluciones al problema del particionado, tanto en tiempo de ejecución como en calidad de éstas.

## 1 Introducción

Durante estos últimos años los sistemas basados en Mundos Virtuales Distribuidos (DVE) han experimentado un espectacular crecimiento. Estos sistemas permiten a un conjunto de usuarios conectados desde diferentes computadores (usualmente a través de Internet) interactuar en un mundo virtual compartido. Cada uno de estos usuarios está representado en el mundo virtual a través de una entidad denominada *avatar*, cuyo comportamiento en el sistema en cada momento viene determinado por el conjunto de acciones que tome su cliente asociado.

---

• Este trabajo ha sido financiado por la Comisión Interministerial de Ciencia y Tecnología (CICYT), proyecto DPI-2002-04438-C02-02.

La interacción simultánea de usuarios en estos sistemas de colaboración abre nuevas posibilidades en el campo de la simulación. El ámbito de aplicación de este tipo de sistemas es bastante amplio [20]. Se han realizado aproximaciones enfocadas al diseño cooperativo ([18]), a las simulaciones distribuidas para el entrenamiento civil o militar ([11]), el aprendizaje interactivo a distancia ([17]) o los conocidos juegos en red multiusuario ([1][6]).

Uno de los aspectos fundamentales en la problemática de los Mundos Virtuales Distribuidos es el *problema del particionado*. Definido en [7] por Lui y Chan para arquitecturas red-servidor, este problema consiste en la asignación eficiente de los clientes de la simulación a los servidores del sistema. Una solución eficiente al problema del particionado es determinante para conseguir un DVE con buenas prestaciones. Aunque tradicionalmente numerosos DVE han implementado su mecanismo de particionado siguiendo el procedimiento ad-hoc de Lui-Chan [8], recientemente se han propuesto nuevas aproximaciones al problema siguiendo esquemas de computación evolutiva([12][14][15] y [16]).

En el presente trabajo se detallan un conjunto de mejoras realizadas sobre una de estas aproximaciones evolutivas que estaba basada en algoritmos genéticos. Estas mejoras maximizan la diversidad estructural de las soluciones en la partición inicial. Además, se introduce un nuevo mecanismo basado en la elección aleatoria del tipo de cruce para cada cromosoma a derivar. El análisis de los resultados obtenidos con dichas mejoras muestra un excelente comportamiento de nuestro algoritmo, evitando la convergencia prematura en la obtención de soluciones que puedan hacer caer al algoritmo en un óptimo local. Sobre un patrón estandarizado de pruebas para DVEs, esta propuesta obtiene soluciones de particionado de mayor calidad y en menor tiempo que las propuestas que recoge la literatura de la materia.

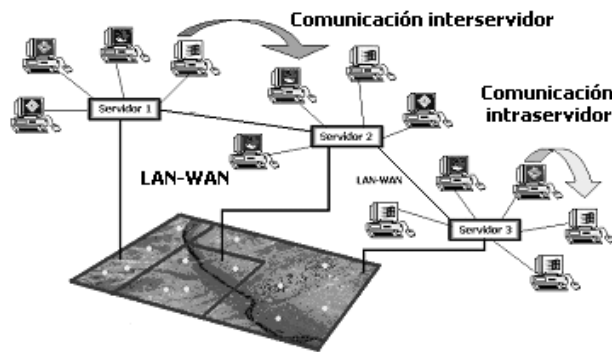
El resto del artículo está organizado de la siguiente manera: la sección II describe el problema del particionado, así como las aproximaciones previas más destacables en su solución. La sección III presenta el nuevo algoritmo genético que proponemos. En la sección IV se expone la evaluación de prestaciones del algoritmo propuesto. Por último, la sección V presenta las conclusiones.

## 2 Estado del arte

Las arquitecturas basadas en servidores interconectados se están convirtiendo en un estándar de facto en la implementación de los DVE. En estas arquitecturas, denominadas red-servidor o servidor-espejo ([9][2]), todo cliente se conecta a uno sólo de los servidores interconectados que controlan la simulación. Cada vez que un usuario modifica la posición o el estado del avatar que lo representa en el mundo virtual compartido, el cliente ha de mandar un mensaje al servidor al cual está conectado. Este servidor, atendiendo a la configuración del sistema en ese momento, redirige la comunicación hacia otro servidor del sistema o bien la redirige sólo a sus propios clientes.

Para evitar una explosión de mensajes circulando por la red se han definido áreas de influencia (AOI) para los avatares. Esta propiedad permite que un avatar sólo propague mensajes hacia los avatares que pertenezcan a su AOI.

Dependiendo de la localización del receptor de un mensaje pueden definirse en un DVE (fig.1) dos tipos de mensajes. Si el avatar receptor está conectado al mismo servidor que el avatar emisor, entonces se realizarán rápidas transferencia intra-servidor. En caso contrario se dice que el mensaje es inter-servidor ya que serán necesarias costosas comunicaciones entre diferentes servidores para que dos avatares puedan intercambiar información.



**Fig. 1.** Modelo de comunicaciones básico en un DVE red-servidor

A partir de una caracterización de los DVE sobre arquitecturas red-servidor [7], Lui y Chan proponen una función de evaluación que valora la eficiencia de la asignación de un conjunto de clientes a los servidores de una simulación. Esta función de evaluación considera dos factores básicos. En primer lugar, la carga computacional que representan los avatares, definida en [7] como  $C_p^W$ . Para optimizar este parámetro (disminuirlo al máximo) el número de avatares asignados a cada servidor ha de ser lo más proporcional posible a la potencia computacional de los servidores, intentado hacer constante el equilibrado de cargas. El segundo factor de esta función es el volumen de las comunicaciones inter-servidor, definido como  $C_p^L$ . Para minimizar este parámetro avatares topológicamente próximos en la escena virtual deberían de ser asignados al mismo servidor.

De acuerdo con estos dos parámetros, se ofrece una función de evaluación del coste para estimar la bondad de una solución de agrupación de avatares cualquiera:

$$C_p = W_1 C_p^W + W_2 C_p^L \text{ tal que } W_1 + W_2 = 1 \quad (1)$$

$W_1$  y  $W_2$  denotan la importancia relativa del factor de carga computacional y de las comunicaciones inter-servidor comentadas anteriormente. En el caso general se tiene que  $W_1 = W_2 = 0.5$ . Es evidente que, cuando el sistema DVE este funcionando sobre una red de altas prestaciones (ej: Myrinet) el cociente  $W_1/W_2$  debe ser mucho mayor que uno. Por el contrario, en redes lentas de medio compartido, o sobre Internet, este factor estará cercano a cero.

Lui y Chan han demostrado la NP-completitud del problema y han propuesto además una plataforma común de pruebas para evaluar sus soluciones y contrastar resultados. Estos mismos autores presentan en [8] un refinamiento de su algoritmo inicial de particionado que prueban sobre estos mismos patrones de test.

Recientemente, debido a la naturaleza intrínsecamente combinatoria del problema se han publicado diferentes aproximaciones basadas en metaheurísticas modernas ([12][14][15][16]). Aunque en [12] se describe un mecanismo constructivo basado en GRASP, las mejores soluciones actualmente recogidas en la literatura de la materia se han obtenido mediante técnicas de computación evolutiva. Así, en [16] se muestran, entre otras, una adaptación de “simulated annealing” al problema y en [14] un sistema de colonia de hormigas para DVE genéricos que obtiene particiones de avatares entre los servidores del sistema. En [15] se detalla un completo estudio donde se comparan un conjunto representativo de aproximaciones evolutivas orientadas al problema frente a la técnica refinada Lui-Chan. Este estudio pone de relieve la sencillez y los excelentes resultados de particionado obtenidos por dichas metaheurísticas, entre las que destaca una aproximación al problema basada en algoritmos genéticos [13].

A pesar de sus buenos resultados, la aproximación basada en algoritmos genéticos [13] parte de una población inicial obtenida a partir de la derivación de una solución única obtenida mediante un algoritmo de k-medias. Ello produce una población inicial bastante acelerada pero con una baja diversidad estructural entre sus individuos integrantes. Tal y como destacan varios trabajos básicos sobre la materia [10][3], este hecho puede provocar que bajo ciertas circunstancias el proceso evolutivo del algoritmo acelere la población inicial en término de *fitness*, (representado por  $C_p$  en este problema), pero llegando a una solución que puede ser un óptimo local o incluso una deficiente aproximación a éste. Junto a este inconveniente es importante destacar que el tipo de cruce o recombinación de individuos de la población aplicado al problema estaba basado en un cruce de auto-fertilización simple a un solo corte [4]. Este cruce, bien conocido en problemas del dominio de las permutaciones [5], es excesivamente generalista, de forma que es posible proponer nuevos cruces asociados a la naturaleza del problema.

### **3 Una nueva técnica genética para el balanceo de cargas**

Con el objetivo de dotar a la propuesta actual basada en algoritmos genéticos [13] de una componente de mayor diversidad estructural en los individuos de la población y de una mayor calidad en la obtención de nuevas generaciones, se presenta una versión mejorada de éste.

El nuevo algoritmo sustituye la generación de la población inicial creada mediante un mecanismo de k-medias por un nuevo sistema basado en proyecciones aleatorias. Junto a esto, la producción de nuevas generaciones se realizará mediante la elección del tipo de cruce entre un conjunto de cinco mecanismos de derivación de soluciones distintos.

### 3.1 Generación de una población inicial heurística

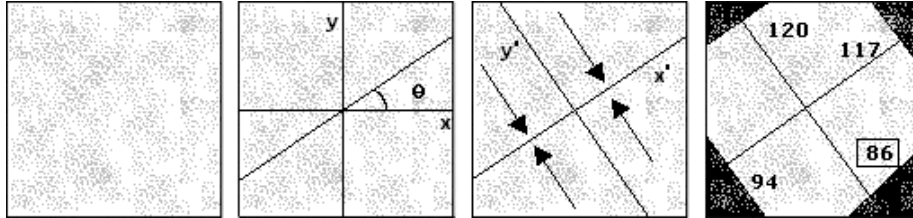
La mayoría de metaheurísticas modernas basan su mecanismo de funcionamiento en la generación de una o varias configuraciones o soluciones iniciales al problema. Durante la ejecución del procedimiento heurístico éstas van siendo transformadas en la solución final a través de una sucesión controlada de iteraciones, de forma que si la población inicial ha sido lo suficientemente bien generada de forma aleatoria, el algoritmo es capaz de alcanzar el óptimo global con cierta facilidad [3]. Tal y como se destaca en [10], si la población inicial es acelerada es importante garantizar la diversidad estructural de estas soluciones para tener una representación de la mayor parte de población posible y evitar con ello la convergencia prematura del algoritmo.

Por este motivo, en el problema del particionado para sistemas DVE es necesario recrear una solución con una fuerte componente aleatoria y lo suficientemente rápida como para que se ejecute de forma totalmente independiente tantas veces como individuos haya en la población. De esta forma, el algoritmo genético partirá de un conjunto de soluciones independientes y aceleradas pero con marcadas diferencias estructurales entre ellas que, por definición del procedimiento, ya se garantizan.

El algoritmo presentado, denominado de *proyecciones*, genera una solución de particionado del DVE independiente para cada avatar. En cada una de estas soluciones el conjunto de  $N$  avatares ( $A_0..A_{N-1}$ ) es alojado totalmente entre los  $M$  servidores ( $S_0..S_{M-1}$ ) del DVE. Por claridad de notación, la solución presentada a continuación está planteada para un escenario virtual en 2D, aunque su transformación para escenas 3D es relativamente trivial.

El algoritmo en primer lugar asigna todos los clientes del DVE al servidor  $S_0$  de la simulación (fig. 2a). A continuación, genera aleatoriamente un valor real  $\theta$  entre 0 y  $\pi/2$ . Dado que todos los avatares están localizados en un plano cartesiano, el algoritmo traza una recta que pasa por (0,0) y tiene como pendiente el valor  $\theta$  generado (fig.2b). Esta nueva recta y su perpendicular en (0,0) definen a su vez un nuevo eje de coordenadas rotado  $\theta$  radianes con respecto al original. Con ello, la posición  $\{x,y\}$  de cualquier avatar en el mundo virtual, con respecto al eje de coordenadas inicial, va a poder referenciarse con respecto al eje rotado con una sencilla transformación afin (fig.2c). Con esta nueva posición  $\{x',y'\}$  de cada avatar el algoritmo crea dos árboles binarios de búsqueda con las claves  $x'$  e  $y'$  de cada avatar. Una vez creados, se eligen para ambos los  $N/M$  avatares con mayor y menor clave ([19]) en los cuales se evalúa su  $C_p^L$ . Dado que la cardinalidad de los 4 conjuntos creados es la misma, e igual a  $N/M$ , no tiene sentido calcular su  $C_p^W$  ya que éste evalúa la desviación típica del conjunto de avatares asignados con respecto al equilibrado perfecto. De entre estos conjuntos de avatares se elige aquel con menor  $C_p^L$  y se le asignan directamente sus clientes al servidor  $S_j$  (fig.2d). En este estado el servidor  $S_0$  tiene actualmente asignados  $\frac{N(M-1)}{M}$  de forma que se repite el proceso

anterior para elegir, de entre ellos,  $N/M$  avatares a asignar a servidor  $S_2$ . Tal y como está planteado el proceso, cuando no queden servidores por asignar y se haya completado el alojamiento de clientes a  $S_{M-1}$  todos los servidores tendrán un número de avatares muy cercano a  $N/M$ .



**Fig. 2.** Vista aérea de la ejecución en fases de la creación de una solución inicial basada en proyecciones en un DVE de 2500 avatares a asignar a 8 servidores: (a) asignación directa a  $S_0$  (b) nuevo eje de coordenadas generado aleatoriamente (c) proyección de cada avatar sobre el nuevo eje (d) creación y evaluación de los 4 conjuntos de asignación asociados a los 2 diferentes árboles de búsqueda. Se asigna a  $S_1$  aquel de los 4 con menor coste  $C_p^L$

Cabe destacar que el uso de 4 árboles de búsqueda binarios se realiza para elevar la diversidad en la elección de conjuntos de avatares a asignar al servidor en curso. Debido a la generación aleatoria de los ángulos de rotación se garantizan soluciones estructuralmente independientes en la generación de individuos de la población inicial. Además, la aceleración inicial de su  $C_p$  vendrá garantizada por el equilibrio de cargas y la asignación al mismo servidor de avatares cercanos en el mundo virtual.

### 3.2 Aleatorización en la elección del cruce de cromosomas

El algoritmo inicialmente publicado basaba su mecanismo de generación de nuevos individuos en un cruce de auto-fertilización simple de un solo corte y probabilidad estática igual a uno. Este tipo de cruce es excesivamente generalista y ha sido utilizado frecuentemente en la resolución de problemas mediante algoritmos genéticos. La técnica que proponemos en este trabajo se basa en la elección aleatoria para cada derivación de un tipo de cruce de entre los detallados en la Tabla 1:

**Tabla 1.** Tipos de cruce a elegir aleatoriamente en la producción de hijos

- 1 Intercambio aleatorio de la asignación de servidores de 2 avatares frontera. Un avatar A *no* es frontera si todos los avatares que se encuentran dentro de su AOI están actualmente asignados al mismo servidor que el avatar A.
- 2 Dado un avatar frontera, elegido aleatoriamente, se construye la lista de servidores con los cuales es frontera. De esta lista se elige un servidor al azar de y se le asigna directamente
- 3 Además de lo realizado en el apartado anterior busca, si existiera, un avatar de la partición receptora con enlaces a vecinos en la partición dadora cambiándole su asignación y provocando, con ello, un intercambio de avatares entre servidores
- 4 Dado que cada avatar genera una carga en el sistema según [7], se escoge aleatoriamente un avatar frontera de aquella partición que tenga una mayor carga soportada global y se le asigna a la partición con menor carga
- 5 Se permuta la asignación de servidores de dos avatares elegidos aleatoriamente. El primero de ellos pertenece al servidor con mayor carga, el segundo al menor

## 4 Evaluación de prestaciones

Para validar el algoritmo presentado en la sección anterior proponemos la evaluación de prestaciones mediante simulación. Tal y como se comenta en la sección II, [8] recoge una importante contribución en el campo de los DVE. En este artículo se ofrece a la comunidad investigadora en la materia una especificación común de creación de mundos virtuales. Mediante ella, cualquier solución al problema del particionado pueda ser evaluada y comparada. Básicamente esta especificación define dos mundos virtuales básicos distintos en los cuales un conjunto fijo de avatares, localizados en una escena virtual cuadrada, han de ser asignados a los servidores de la simulación. Este trabajo denomina mundos SMALL a aquellos que simulan 13 avatares sobre 3 servidores y mundos LARGE a aquellos en los que 2500 avatares se alojan en 13 servidores de simulación distintos. Con el fin de evaluar el comportamiento dinámico de las aplicaciones 3D en todo tipo de situaciones, se definen para ambos mundos tres tipos distintos de distribuciones de localización de avatares denominadas *uniforme*, *sesgada* y *agrupada*.

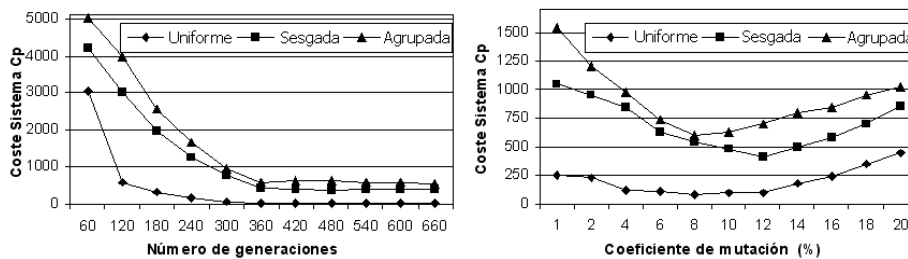
Para recrear ambos mundos en un modelo real de DVE se han utilizado las instalaciones de las aulas del Departamento de Informática de la UV. Dichas aulas están equipadas con computadores Pentium IV a 1.5Ghz con 256 MB de memoria RAM, tarjeta gráfica nVidia Geforce2 MX-400 y conexión a LAN de 10Mbps. Sobre esta gran instalación, y a efectos de simulación, cada uno de los servidores del DVE corría en exclusiva en un equipo de la red. Para la recreación de todos los avatares en la escena 3D se han conseguido alojar satisfactoriamente 50 clientes por computador. Para ello, se ha eliminado la interfaz gráfica de los clientes y se ha sustituido por una interfaz modo texto necesaria en la recogida de datos de la simulación. Los avatares robotizados, siguiendo las diferentes distribuciones comentadas, fueron localizados aleatoriamente en la escena. A cada uno de ellos se le asignó una carga computacional modelada como un valor real entre 0 y 5. A su vez, para el cálculo de  $C_p$ , se cuantifico la relación de comunicaciones que caracteriza a dos avatares como un valor real (entre 0 y 5) inversamente proporcional a las distancia que les une (desde 5 en avatares muy cercanos y hasta casi 0 cuando las distancia que les separa está ya cercana a  $D/2$ ).

Cabe destacar que los resultados detallados a continuación valoran tanto la calidad de las soluciones obtenidas por los algoritmos, en términos de  $C_p$ , así como el tiempo necesario para calcularlas. Lógicamente, soluciones obtenidas de bajo  $C_p$  provocarán excelentes particiones clientes/servidores en el sistema, creando un DVE de baja latencia de comunicaciones y alta interactividad de uso. Por el contrario, debido a que la ejecución de este algoritmo se realiza por uno de los servidores del DVE, el tiempo empleado para ello debe ser el menor posible intentando no entorpecer las labores del control de avatares para las cuales el servidor está inicialmente diseñado.

### 4.1 Parametrización del nuevo algoritmo genético

De igual forma que se realiza en [13] y [15] es necesario calibrar los parámetros básicos que hacen variar el comportamiento del algoritmo genético para el problema; típicamente estos son tamaño de la población, número de iteraciones y tasa de mutación.

La figura 3 muestra el historial de convergencia para el algoritmo presentado del número de generaciones y del coeficiente de evaporación con respecto al “fitness”. Éste viene descrito mediante la función de coste  $C_p$ . Por problemas de espacio no se muestran los datos asociados a la variación del tamaño de la población cuya incidencia en el sistema es similar a la creada por la modificación del número de generaciones.



**Fig. 3.** Parametrización del comportamiento del algoritmo para un DVE LARGE. Historial de convergencia para el número de generaciones y el coeficiente de mutación

A la vista de los resultados, en mundos LARGE se puede afirmar que para el conjunto de patrones de DVE pasados al algoritmo carece de sentido iterar por encima de las 400 generaciones. De forma análoga, se encuentran valores razonablemente buenos del tamaño de población cercanos a los 20 individuos. El comportamiento del algoritmo es algo distinto con respecto a la variación del coeficiente de mutación. El algoritmo logra sus mejores prestaciones para una variación del coeficiente entre el 8-10%. Valores situados por encima de este rango de mutaciones hacen al algoritmo caer en posibles óptimos locales de los cuales le cuesta mucho salir. Por el contrario, una excesiva tasa de mutación no permite una especialización o evolución de los individuos que paulatinamente van siendo generados.

## 4.2 Resultados en mundo virtuales SMALL y LARGE

Siguiendo los patrones de pruebas Lui-Chan para DVE, se ha evaluado el comportamiento de la solución propuesta en el presente artículo. Tanto para mundos SMALL como para mundos LARGE se va a comparar la calidad de las soluciones obtenidas por la implementación propuesta por Lui-Chan en [8] como con la genética básica descrita en [15]. La Tabla 2 muestra los resultados obtenidos para mundos SMALL y para mundos LARGE, tanto en tiempo de ejecución del algoritmo como en la calidad de la solución obtenida. Debido a que son dos las mejoras aportadas al algoritmo inicial, se muestran las mejoras provocadas solamente por la sustitución del algoritmo inicial por el basado en proyecciones (fila V1) y las aportadas con nuestra propuesta, que consiste en esta sustitución y además en la modificación del tipo de cruce propuesto en la sección 3.2 (fila V2).

Con el objetivo de detallar al máximo el tamaño del problema fue posible, solamente en mundos virtuales SMALL, reproducir una búsqueda exhaustiva que permitiera una visión global en la comparativa de métodos. Debido al reducido



tamaño del modelo para estos mundos, una búsqueda completa requiere la exploración de  $3^{13}$  (1.594.323) diferentes soluciones. La implementación de búsquedas exhaustivas es útil, en la medida que sea factible, como medida de evaluación a la eficiencia de las heurísticas implementadas para el problema.

**Tabla 2.** Comparativa de prestaciones del algoritmo presentado frente a propuestas en  
(a) Mundos Virtuales SMALL

	Distrb. Uniforme		Distrb. Sesgada		Distrb. Agrupada	
	Tiempo (s)	Cp	Tiempo (s)	Cp	Tiempo (s)	Cp
<b>B.Exhaustiva</b>	3.411	6.54	3.843	7.04	4.783	7.91
<b>Lui-Chan</b>	0.0009	6.56	0.001	8.41	0.0011	8.89
<b>Genético Básico</b>	0.002	6.54	0.003	7.04	0.005	7.91
<b>V1</b>	0.002	6.54	0.003	7.04	0.006	7.91
<b>V2</b>	0.003	6.54	0.003	7.04	0.006	7.91

(b) Mundos Virtuales LARGE

	Distrb. Uniforme		Distrb. Sesgada		Distrb. Agrupada	
	Tiempo (s)	Cp	Tiempo (s)	Cp	Tiempo (s)	Cp
<b>Lui-Chan</b>	30.9	1637.0	32.1	3460.5	43.3	5903.8
<b>Genético Básico</b>	6.6	1832.2	14.6	2825.6	29.2	4905.93
<b>V1</b>	6.2	547.2	14.7	612.9	28.7	1002.5
<b>V2</b>	6.4	321.3	15.0	450.8	28.6	791.9

Estos resultados muestran que al igual que lo hacia la aproximación genética inicial, el esquema propuesto (recogido en V2) obtiene rápidas soluciones de particionado alcanzando en la mayoría de los casos valores iguales a los obtenidos mediante una técnica exhaustiva. Sin embargo, a la hora de diseñar DVE escalables el mecanismo de particionado debe de ofrecer soluciones muy eficientes cuando el número de avatares del DVE aumenta. Tal y como puede apreciarse en los resultados para grandes mundos LARGE, la técnica presentada obtiene soluciones de mayor calidad que las obtenidas por la aproximación genética básica. Por otra parte a la vista de los resultados, cabe destacar las mejoras ofrecidas solamente por la modificación de la partición inicial, mostrada en V1.

## 5 Conclusiones

En este artículo hemos presentado dos mejoras al algoritmo genético básico utilizado para la solución del problema del particionado en mundos virtuales distribuidos (DVE). Estas mejoras están basadas en la maximización de la diversidad estructural de la población inicial y en la creación de una base aleatoria de cruces dirigidos.

Comparado con las técnicas actuales de particionado, las pruebas obtenidas sobre una plataforma de DVE recogida en la literatura de la materia destacan un excelente

comportamiento del nuevo algoritmo genético en grandes mundos virtuales. Esto hace que los sistemas que incorporen esta solución mejoren su escalabilidad y puedan gestionar simultáneamente y de forma eficiente un volumen mayor de avatares.

## Referencias

1. M. Abrash, "Quake's game engine: The big picture", Dr. Dobbs's Journal, Spring, 1997
2. T.A. Funkhouser, "Network Topologies for Scalable Multi-User Virtual Environments", Technical Report Bell Laboratories, 1996
3. J.H. Holland and D.E. Goldberg, "Genetic algorithms and machine learning: Introduction to the special issue on genetic algorithms", Machine Learning, 3, 1998
4. K.E. Kinnear, "Alternatives to Automatically Function Definition", In Advances in Genetic Programming 1994, MIT Press, pp. 119-141
5. E. Kirshenbaum, "Genetic Programming With Statically Scoped Local Variables", Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2000, July, 2000, pp. 459-468
6. M. Lewis and J. Jacobson, "Game Engines in Scientific Research ", Communications of the ACM., Vol. 45, N°1 January 2002
7. J.C.S. Lui, M.F. Chan and K. Oldfield, "Dynamic Partitioning for a Distributed Virtual Environment", Dpt. of Computer Science, The Chinese University of Hong Kong, 1998
8. J.C.S. Lui and M.F. Chan, "An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems", IEEE, TPDS Journal, Vol. 13, N° 3, March 2002
9. M.I. R. Macedonia, "A Taxonomy for Networked Virtual Environments", IEEE Multimedia, 4(1) 48-56, January-March 1997
10. Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Springer-Verlag, Second Edition, 1992
11. D.C. Miller, J.A. Thorpe, "SIMNET: The advent of simulator networking ", Proceedings of the IEEE, 83(8):1114-1123, August 1995
12. P. Morillo and M. Fernández, "A GRASP-based algorithm for solving DVE partitioning problem", Proceedings of 2003 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPS-03), Nice, France, April, 2003
13. P. Morillo, M. Fernández y V. Arnau, "Unas primeras aproximaciones evolutivas al problema del particionado en mundos virtuales distribuidos", MAEB 03, Gijón, Febrero 2003
14. P. Morillo, M. Fernández and J.M. Orduña, "An ACS-Based Partitioning Method for Distributed Virtual Environment Systems", Proceedings of 2003 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPS-03), Nice, France, April, 2003
15. P. Morillo, M. Fernández and J.M. Orduña, "A Comparison Study of Modern Heuristics for Solving the Partitioning Problem in Distributed Virtual Environment Systems", Proc. of Int. Conference on Computational Science and its applications (ICCSA' 2003), May 2003
16. P. Morillo, M. Fernández and N. Pelechano "A grid representation for Distributed Virtual Environments" Proceedings of 2003 1st European Across Grids Conference, Santiago de Compostela, Spain, February, 2003
17. T. Nitta, K. Fujita, S. Cono, "An Application Of Distributed Virtual Environment To Foreign Language", Kansas City, Missouri, IEEE Education Society, October 2000
18. J.M. Salles, R. Galli, A.C. Almeida, C. Belo and J.M. Rebordão, "mWorld: A Multiuser 3D Virtual Environment", IEEE Computer Graphics, March-April, 1997, Vol. 17, No. 2
19. R. Sedgewick, "Algorithms in C", 3<sup>rd</sup> ed., Addison-Wesley, 1998
20. S. Singhal and M. Zyda, "Networked Virtual Environments", ACM Press, New York, 1999