

Unas primeras aproximaciones evolutivas al problema del particionado en mundos virtuales distribuidos

Pedro Morillo[♦], Marcos Fernández[♦], Vicente Arnau[♠]

Resumen—El incremento, tanto del rendimiento de las tarjetas gráficas de última generación, así como el de las rápidas conexiones a Internet, está haciendo popular a los **Mundos Virtuales Distribuidos**. Esta incipiente tecnología ha de solucionar de forma eficiente diferentes problemas asociados a la heterogeneidad en: hardware de los clientes, requisitos de las aplicaciones gráficas y diversidad del tipo de conexión a la red. La tarea de asignar clientes de una simulación a los servidores de ésta, en una arquitectura red-servidor, es una de ellos. En este trabajo se describe un nuevo enfoque de éste problema NP-completo, denominado **problema del particionado**, basado en metaheurísticas evolutivas. Los primeros resultados obtenidos, comparados con la principal referencia en el área, parecen indicar que el enfoque evolutivo obtiene soluciones más eficientes empleando menores tiempos de ejecución.

Palabras Clave—**Mundos Virtuales Distribuidos**, **enfriamiento simulado**, **algoritmos genéticos**, **búsqueda heurística**, **equilibrado de cargas**.

I. INTRODUCCIÓN

El aumento de la potencia y rendimiento de los ordenadores personales ha hecho que en los últimos años se popularicen las aplicaciones de entornos virtuales en 3D. Los primeros entornos, o mundos virtuales creados, han estado siempre basados en arquitecturas monousuario. El paso del tiempo ha hecho que las tradicionales aplicaciones basadas en gráficos 3D tiempo real evolucionen en la denominada Realidad Virtual Distribuida. Estos modernos sistemas permiten a diferentes usuarios interactuar en un escenario virtual común. Cada uno de ellos es cliente de la simulación y puede estar localizado físicamente en un punto remoto, completamente alejado del resto de integrantes del sistema. Esta interacción simultánea de usuarios en estos sistemas colaborativos abre nuevas posibilidades en el campo de la simulación. El ámbito de aplicación de este tipo de sistemas, descrito en [1], es bastante amplio. Se han realizado aproximaciones enfocadas al diseño colaborativo ([2]), a las simulaciones distribuidas para el entrenamiento civil o militar ([3]), el aprendizaje interactivo a distancia ([4]) o los conocidos juegos en red multiusuario ([5][6]).

Básicamente, dos conceptos relevantes deben de ser tratados a la hora de diseñar un mundo virtual

distribuido (DVE, del inglés “Distributed Virtual Environment”).

En primer lugar, la **heterogeneidad múltiple** que caracteriza a ese tipo de sistemas. Esta aparece en:

- Cada *cliente*, que representado usualmente en la escena por una figura humanoide o avatar, suele poseer unas características hardware (velocidad procesamiento, tamaño memoria, tecnología de tarjeta gráfica) totalmente distintas de los integrantes de la simulación.
- El *tipo de conexión* a Internet entre los nodos de la simulación. Desde instalaciones sobre medios compartidos como Ethernet o Fast-Ethernet en LAN, hasta RDSI, ADSL o ATM en WAN, pueden encontrarse como medio de transferencia de datos en algunos DVEs.
- *Tasa de comunicación* entre avatares. Estas aplicaciones suelen basarse en un esquema de paso por mensaje. La frecuencia de envío y recepción de estos no es constante y puede variar bastante de entornos 3D colaborativos a batallas militares virtuales en tres dimensiones.

El otro concepto a solucionar está asociado a los **problemas intrínsecos** que estas tecnologías han de resolver. Hoy en día, cada una de ellas, supone una línea de investigación totalmente abierta:

- *Modelo de datos*. Detallados in [7], describen las diferentes posibilidades en la que el conjunto de datos persistentes o semi-persistentes de la simulación pueden ser gestionados. Principalmente se emplean metodologías de replica, compartición o distribución de datos entre clientes.
- *Modelo de comunicaciones*. El ancho de banda determina el tamaño y la eficiencia de un DVE; el comportamiento de estos integrantes del mundo virtual va a depender de la forma en la que los clientes se conecten. Esquemas broadcast, P2P o unicast definen diferentes valores de latencia de red para el intercambio de información entre avatares.
- *Vistas consistentes*. Problema abordado en otros campos de la computación como la gestión de bases de datos ([8]). En los sistemas DVE se ha garantizado que, avatares que comparten una misma zona virtual han de tener una visión local común de los objetos contenidos en ésta.
- *Reducción del tráfico de mensajes*. El mantenimiento de un bajo volumen de mensajes en la red permite a los DVE escalar con cierta facilidad. Tradicionalmente, técnicas como “deadreckoning” descrita en [1] ofrecían cierto nivel de

[♦] Grupo ARTEC. Instituto de Robótica. Universidad de Valencia.
Correo Electrónico: {Pedro.Morillo,Marcos.Fernandez}@uv.es.

[♠] Departamento Informática. Universidad de Valencia.
Correo Electrónico: Vicente.Arnau@uv.es.

independencia a los avatares. A su vez, con ayuda de la red, soluciones basadas en broadcast ([9],[10]) hacen descender el mínimo número de mensajes necesarios para mantener un estado consistente del sistema.

Gran parte de la problemática básica que caracteriza a este tipo de sistemas está asociado a un aspecto clave en su diseño, denominado el *problema del particionado*. Definido en [11] por Lui-Chan para los mundos virtuales distribuidos, este problema establece una relación lógica entre la asignación de clientes a servidores y el rendimiento de la red del sistema en arquitecturas red-servidor. Este problema, así como todas las aproximaciones previas más importantes, son descritos en la sección II. La sección III presenta una solución basada en algoritmos evolutivos, caso del enfriamiento simulado y algoritmo genético simple. En la sección IV, los resultados obtenidos sobre un sistema de evaluación de estas arquitecturas indican la idoneidad de las técnicas presentadas que, comparadas con la máxima referencia en la materia, consiguen configuraciones del DVE de mayor calidad, empleando a su vez para su computo menor tiempo. Por último, las contribuciones del trabajo son resumidas en la sección V, así como el planteamiento del trabajo futuro.

II. PROBLEMA PARTICIONADO.

A pesar de la heterogeneidad del modelo descrito anteriormente, existe un conjunto de características comunes a los modernos DVEs. Se pueden encontrar numerosas referencias en la literatura de la materia, ([1][7][11]) en las cuales la implementación de los DVE's sobre arquitecturas red-servidor se está convirtiendo en un standard de facto. Tecnologías de red basadas en el procesamiento paralelo, como la gestión de servidores web de alto rendimiento, están llegando a este nivel a las mismas conclusiones ([14]). En estas arquitecturas, denominadas también jerárquicas por algunos investigadores ([16]), la simulación es controlada por un conjunto de servidores interconectados. Cuando un cliente desea introducirse en una simulación éste obligatoriamente es asignado, a uno y sólo uno, de estos servidores coordinados. Así, para cada actualización (ej.: cambio de posición) de su avatar asociado, el cliente envía un mensaje al servidor, el cual, atendiendo a la configuración del sistema en ese momento, dirige la comunicación hacia otro servidor de la simulación o, si es el caso, a sus propios clientes. Para evitar un desbordamiento del volumen de mensajes circulando en la red se suelen definir áreas de influencia (AOI en [1], locales en [17] y aura en [18]) para los avatares. Esta propiedad permite que un avatar sólo propague mensajes, hacia otros avatares que pertenezcan a su AOI (del inglés "area of influence"). Dependiendo de la localización del receptor de un mensaje pueden definirse en un DVE (fig.1) dos esquemas diferentes de comunicación. Si éste es alojado en el mismo servidor que el emisor se realizarán rápidas transferencia intra-servidor; en caso contrario, serán

necesarias costosas comunicaciones inter-servidor para que dos avatares puedan intercambiar información.



Fig. 1. Modelo comunicaciones básico en un DVE

Sobre esta arquitectura, Lui y Chan en [11] redefinen el *problema del particionado*. Este consiste en la búsqueda de la mejor asignación, del conjunto de clientes a los servidores de la simulación, que permita una óptima interactividad a los usuarios de aplicaciones (a nivel de *frame-rate*) mediante la minimización del tráfico de red. Se han de considerar dos factores básicos a la hora de crear una eficiente asignación. En primer lugar, C_p^W , el número de avatares asignados a cada servidor ha de ser lo más similar posible, creando el denominado balanceo o equilibrado de cargas. En segundo lugar, para minimizar el volumen total de comunicaciones inter-servidor, C_p^L , avatares localizados en emplazamientos vecinos deberían de ser asignados al mismo servidor. Dado que simultáneamente para todos los avatares de la escena esto no es posible, es entonces necesario encontrar un adecuado agrupamiento topológico. De acuerdo a estos dos parámetros, se ofrece una función de evaluación del coste para estimar la bondad de una solución de agrupación obtenida:

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

W_1 y W_2 denotan la importancia relativa del factor de carga computacional y de las comunicaciones interservidor 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 sea mucho mayor que uno. Por el contrario, en redes lentas de medio compartido, o sobre Internet, este factor estará cercano a cero.

Estos mismos autores demuestran en [13] la NP-completitud del problema además de ofrecer a los investigadores una plataforma común de pruebas para evaluar sus soluciones y contrastar resultados.

Además de esto, se describe un refinamiento de su algoritmo inicial, cuyas bases son publicadas en [12], así como una paralelización sencilla basada en criterios de volumen de avatares.

Existen otras aproximaciones al problema del particionado empleando diferente terminología ([18] y [19]). En [18] se detalla una solución que agrupa avatares en distribuciones topológicamente regulares. Para asegurar buenas prestaciones gráficas el número de distribuciones creadas se iguala al de servidores con lo que la correlación es directa. Esta solución no obtiene buenos resultados cuando los avatares no se encuentran distribuidos de manera uniforme.

Otra aproximación publicada ([19]) no basa el particionado en conceptos dinámicos como AOI, aura o locales. Para ello divide la escena 3D virtual en una malla regular. Sobre cada celda crea un grupo multicast ahorrando muchos mensajes. A pesar de ser una solución rápida y determinista, carga en exceso el rendimiento cuando los avatares se encuentran agrupados en zonas concéntricas.

III. APROXIMACIONES EVOLUTIVAS

Todas las soluciones comentadas anteriormente para el problema del particionado, clasificado como NP-completo, comparten la misma característica. Son soluciones a medida definidas como heurísticas *ad-hoc* para resolver el problema.

En la literatura reciente de la materia se pueden encontrar problemas similares al de particionado en DVEs, tales como el balanceo de cargas ([27]) y la asignación de procesos a procesadores en redes de estaciones de trabajo ([20]), abordados mediante técnicas evolutivas. Esto hizo que se considerara interesante aplicar esta base de conocimiento en lugar de construir una solución desde el comienzo.

Dentro de estas técnicas, denominadas meta-heurísticas modernas, se encuentran los algoritmos evolutivos. Descritos en [29], son algoritmos de búsqueda no exhaustivos que resuelven problemas basándose en la evolución de procesos en la naturaleza. Los algoritmos presentados en este trabajo, fundamentados en técnicas de programación genética (AG) y enfriamiento simulado (SA), son de este tipo. Dado que la eficiencia de una clasificación de avatares, para un DVE dado, es medida en términos de C_p , encontrar un mínimo en su espacio de soluciones equivale a incrementar las prestaciones del sistema a nivel de interactividad. Es aquí donde van a intervenir estos algoritmos, los cuales una vez adaptados, van a ofrecer de forma autónoma soluciones de agrupamiento de clientes (avatares) en servidores en el mundo virtual 3D.

A. Elección de las configuraciones iniciales

La mayoría de metaheurísticas modernas basan su modo de funcionamiento en la generación de una o varias configuraciones (o presoluciones) iniciales. Durante la ejecución del procedimiento heurístico éstas van siendo transformadas en la solución final a través de una sucesión controlada de iteraciones. Tal y como se publica en trabajos que describen las metodologías tratadas en el presente artículo ([22] y [24]), el conjunto de configuraciones iniciales de partida, para la resolución del problema, ha de ser generado de forma

aleatoria. Esto conlleva, para el problema del particionado en DVE's, que en el instante inicial de comienzo del algoritmo, todo avatar va a estar preasignado a un servidor de forma totalmente aleatoria. Con ello, su valor hacia soluciones más óptimas, que reduzcan el coste del particionado global, será tanto más eficaz según sea la efectividad del método de clasificación. En la realidad, a pesar de estas consideraciones, la efectividad de una ejecución para la resolución de un problema en un conjunto de iteraciones dado, va a ser mucho mayor si el conjunto inicial de soluciones ha sido cuidadosamente elegido. La afirmación es peligrosa, ya que estas metodologías, que buscan buenas particiones del sistema, basan su funcionamiento en una derivación intensiva de un conjunto de soluciones parciales. En toda la secuencia de iteración se busca que el coste de la última solución precalculada sea reducido al máximo evitando la caída en mínimos locales. Por ello, podría siempre parecer que cuanto más evolucionada se encuentre esta configuración inicial, menor será el coste de particionado de las soluciones que pueda obtener en un tiempo dado el AG. Realmente, la convergencia a un óptimo global no se asegura en muchos casos si la población inicial no es totalmente aleatoria. Esto se debe a que favorecerla de forma inicial puede llevar al algoritmo a prematuras convergencias a mínimos locales.

Con esta precaución y para no adular la llegada al óptimo global, se realizó un análisis previo de la forma en que el AG presentado evoluciona sus propias soluciones intermedias. Con este análisis y basándose en la teoría de grafos para la clasificación de patrones ([21]), se ha diseñado un algoritmo inicial muy rápido que particiona en grano grueso de forma equilibrada los avatares de una escena virtual. La figura 2 muestra los resultados de este algoritmo inicial en la reclasificación de un DVE formado por 2500 elementos a repartir entre 8 servidores. Estos se localizan en el mundo virtual siguiendo tres diferentes distribuciones de probabilidad.

En toda simulación presentada en este artículo se comprobó realmente que las soluciones calculadas realizando la configuración inicial introducida obtenían soluciones de la misma calidad que las que parten de poblaciones aleatorias. Para ello lógicamente, empleando menor tiempo.

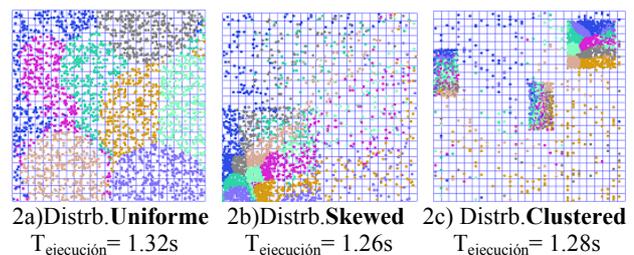


Fig. 2. Proceso de pre-asignación

B. Enfriamiento Simulado (SA)

Método originalmente publicado en [22] que adopta conceptos físicos de la termodinámica. Simulated Annealing (SA) es un procedimiento basado en búsqueda local donde todo movimiento de mejora es aceptado y se permiten movimientos de empeoramiento de acuerdo con unas probabilidades. Aunque existen referencias en la literatura de la materia concernientes a diferentes problemas resueltos mediante SA ([23]), ha sido necesaria una adaptación y configuración previa para abordar, mediante esta técnica, el problema del particionado. Esta se basa en la evolución de una pre-solución inicial como la descrita anteriormente. Para ello, de forma iterativa, SA, durante un número de pasos dados (intentos), va a elegir 2 avatares de forma aleatoria entre los participantes en el DVE con la característica de que pertenezcan a servidores distintos. Una vez elegidos se intercambiarán entre ellos los servidores a los que pertenecen, computando tras esta permutación el coste C_p de la nueva partición obtenida. Si esta nueva solución reduce el coste de la anterior configuración, o a pesar de ser mayor lo es tan solo por debajo de un cierto umbral, esta permutación se mantiene. En caso contrario, como indica el algoritmo, se ha de deshacer el cambio y continuar con otra iteración.

A pesar de la aparente sencillez del proceso, son muchos los parámetros a configurar para la obtención de buenos resultados. Después de un análisis riguroso, se destacan 4 características básicas en la obtención de soluciones para el problema del particionado:

a) *Avatares frontera*. La elección de un avatar, de entre los que pertenecen a un mismo servidor no ha de realizarse aleatoriamente. Esta búsqueda no ha de ser libre, sino guiada, y ha realizarse sólo entre los *avatares frontera* de esa agrupación. Se entiende por avatar frontera a aquel que contiene en su AOI al menos un avatar alojado en un servidor diferentes al suyo. La justificación se basa en la forma en la que se realiza la búsqueda de la reducción iterativa de C_p . Debido a que un intercambio de avatares no modifica el factor equilibrado de cargas C_p^W , la reducción del coste total mediante permutaciones se consigue haciendo disminuir el coste de intercomunicación C_p^L . Por la forma en la que este sub-coste se calcula (evaluando las comunicaciones entre elementos cercanos), si un avatar, que estaba totalmente rodeado en un espacio virtual por vecinos asignados a su propio servidor, es asignado a otro servidor cualquiera, entonces, los costes de comunicación interservidor, transcurrido este cambio, se incrementarán.

b) *Elementos intercambiables*. A pesar de lo especificado por el método general, cabe la posibilidad de intercambiar más de dos elementos a la hora realizar cada iteración del método en la búsqueda del descenso de coste. La tabla 1 confirma que esto no debe realizarse:

| | Intercambio 2a2 | | Intercambio 3a3 | | Intercambio 4a4 | |
|-----------|-----------------|-------|-----------------|-------|-----------------|-------|
| SA | Cp | T(s) | Cp | T(s) | Cp | T(s) |
| Uniforme | 1707,62 | 6,35 | 1808,38 | 6,51 | 1817,90 | 7,02 |
| Skewed | 2628,46 | 13,79 | 2826,44 | 14,32 | 2992,30 | 15,82 |
| Clustered | 4697,61 | 29,62 | 5046,27 | 30,25 | 5283,70 | 33,98 |

Tabla 1. Análisis del número de elementos intercambiables

c) *Número de iteraciones*. Factor clave en la aplicación eficiente de la técnica. Lógicamente, el aumento del número de iteraciones va a ser directamente proporcional al tiempo requerido para la ejecución e inversamente al coste final de la solución obtenida. El problema radica en que el factor de ambos procesos lineales es bastante diferente, por lo que se ha de detectar un valor que a pesar de explorar exhaustivamente el árbol de decisiones, consiguiendo una solución de bajo coste, no requiera demasiado tiempo para ello. A su vez, debido a exigencias del tipo de pruebas a someter a las soluciones en el problema del particionado en DVEs ([13]), se ha de intentar que el valor para el parámetro elegido favorezca de forma ecuánime a las tres diferentes distribuciones sin perjuicio, en particular, para ninguna de ellas.

d) *Coficiente de enfriamiento*. Determina la tolerancia del algoritmo a la aceptación de cambios desfavorables. Su variación no va repercutir en el incremento de los tiempos de ejecución del algoritmo ya que su descenso gradual, con el número de iteraciones, no interviene directamente en el coste temporal de la obtención de soluciones, aunque si en la calidad de estas al permitir al algoritmo la salida de posibles mínimos locales. Debido a que la elección de estos parámetros, en concreto los dos últimos, va muy ligado a la batería de pruebas, se mostrará en la sección IV una configuración calibrada de estos.

C. Algoritmos Genéticos (AG)

Son algoritmos heurísticos basados en el concepto darvinista de evolución. Es muy abundante la literatura en la materia y sus principales aportaciones se basan en la solución de problemas complejos de diversa naturaleza. Para ello, crean poblaciones ([24]) de pre-soluciones que dejan evolucionar, por generaciones, hasta que obtienen post-soluciones de calidad aceptable. En publicaciones más recientes ([25]) se describen implementaciones y comparativas de la optimización de estos algoritmos mediante la paralelización de sus tareas.

Abordando el problema del particionado, la implementación propuesta en este trabajo se basa en partir de una población compuesta por un conjunto de individuos o cromosomas. Cada uno de estos individuos vendrá representado por un vector de genes indicador, en un instante del tiempo, de una de las posibles configuraciones en la que el DVE a simular va a poder particionarse. Por lo tanto, cada uno de los elementos de este vector (descriptor de la asignación de un avatar

para una solución) tomará un rango de valores igual al número de servidores que componen la simulación del DVE. Sobre esta población se aplica un algoritmo simple con probabilidad de cruce igual a uno, mutando posteriormente todos los descendientes. Como selección entre los padres e hijos se seleccionan los mejores de esa población intermedia con el fin de dejar fijo el tamaño de la población.

A la hora de generar soluciones para un DVE, siguiendo técnicas genéticas simples como la propuesta, el número de individuos de la población, el total de iteraciones y la tasa de mutación son los parámetros básicos a calibrar en este tipo de algoritmos.

IV. RESULTADOS

Tal y como se comenta en la sección II, [13] recoge una importante contribución en el campo de los DVEs. 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 o rejilla virtual cuadrada, han de ser asignados a los servidores de la simulación. Así se denominan mundos SMALL a aquellos que simulan 13 avatares sobre 3 servidores (rejilla 4x4) y mundos LARGE a aquellos en los que 2500 avatares se alojan en 13 servidores de simulación distintos (rejilla 25x25). 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 (uniforme, sesgada-skewed, y agrupada-clustered).

Después de implementar ambos patrones de evaluación es necesario destacar 3 características inherentes de este test: a) ancho de banda uniforme entre todos los nodos de la arquitectura, ya sea entre tramos servidor-servidor o cliente-servidor; b) las características técnicas se consideran comunes a todos los servidores de las simulación; c) no es necesario ofrecer el tamaño de los escenarios virtuales en Km. o m. ya que existe una relación que establece que si un avatar tiene un AOI de diámetro igual a D unidades, entonces las celdas regulares lógicas que forman la rejilla de la escena tendrán un área de D^2 unidades.

Manteniendo estas especificaciones una batería de test, formada por cientos de diferentes DVEs, fue pasada a los dos algoritmos en una aplicación prototipo. En ella, los avatares, siguiendo las diferentes distribuciones comentadas, fueron localizados aleatoriamente en la escena. A cada uno de estos avatares 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 une a dos avatares como un 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 esta 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.

La obtención de estos resultados se ha realizado mediante una red de computadores formada por equipos PC equipados con procesadores Pentium IV a 1.5Ghz, 256 MB de RAM y tarjeta gráfica nVidia Geforce2 MX- 400.

A. Parametrización de SA y AG en el problema

Tal y como se comenta anteriormente, la versión de SA y AG presentadas para el problema del particionado han de ser calibradas a la hora de atacar los dos diferentes conjuntos de prueba. Mientras que para SA se han de elegir valores adecuados del número de iteraciones, y del coeficiente de enfriamiento, para AG se han de calibrar número de generaciones, número de individuos en la población y tasa de mutación. Debido a razones en la extensión del artículo, carece de sentido mostrar el historial de convergencia en la elección de todo el rango de valores combinados en los diferentes parámetros. A pesar de ello se destaca que, las mejores soluciones cuyos resultados numéricos en tiempo de ejecución y calidad C_p se han obtenido en SA con 2750 iteraciones y 0.75% en coeficiente de enfriamiento. Para AG se han elegido, a la vista de los historiales, poblaciones de 15 individuos en 300 generaciones con tasas de mutación de 1%.

Como ejemplo representativo en AG, se muestra en la figura 3 la variación de la calidad de soluciones C_p frente al número de generaciones, manteniendo constante el tamaño de la población a 15 individuos con tasa de mutación del 1%.

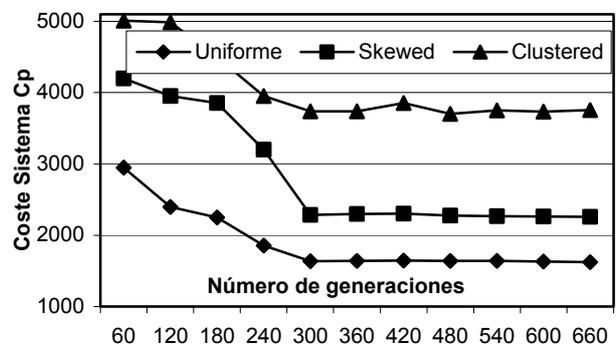


Fig. 3. Ejemplo representativo historial convergencia AG

A la vista de los resultados carece de sentido, para el caso propuesto, iterar AG por encima de las 300

generaciones donde no se obtendrán soluciones de mejor calidad malgastando tiempo en su computo.

B. Resultados en mundos virtuales SMALL

Con el objetivo de detallar al máximo el tamaño del problema, y sólo para este tipo de mundos virtuales, fue posible 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. Tal y como muestra la tabla número 2, es necesario gastar más de 4 segundos, en algunos casos, para asegurar que el algoritmo exhaustivo alcanza la solución óptima. A pesar de que estos tiempos de ejecución están lejos de las especificaciones temporales en tiempo real, la implementación de búsquedas exhaustivas es útil, en la medida que sea factible, como medida de evaluación a la eficiencia de las heurística implementadas para el problema.

| | Computación t(s) | Coste Sistema Cp |
|--------------------------|------------------|------------------|
| Exhaustivo Alg. | 3.411 | 6.54 |
| Lui-Chan Alg. | 0.0009 | 6.56 |
| Enfriamiento Alg. | 0.002 | 6.82 |
| Genéticos Alg. | 0.002 | 6.54 |

(a) Avatares en Distribución Uniforme

| | Computación t(s) | Coste Sistema Cp |
|--------------------------|------------------|------------------|
| Exhaustivo Alg. | 3.843 | 7.04 |
| Lui-Chan Alg. | 0.0010 | 8.41 |
| Enfriamiento Alg. | 0.005 | 7.46 |
| Genéticos Alg. | 0.003 | 7.04 |

(b) Avatares en Distribución Skewed

| | Computación t(s) | Coste Sistema Cp |
|--------------------------|------------------|------------------|
| Exhaustivo Alg. | 4.783 | 7.91 |
| Lui-Chan Alg. | 0.0011 | 10.56 |
| Enfriamiento Alg. | 0.006 | 7.91 |
| Genéticos Alg. | 0.005 | 7.91 |

(c) Avatares en Distribución Clustered

Tabla 2. Resultados en mundos virtuales SMALL

Al comparar ambas soluciones evolutivas con el algoritmo Lui-Chan, implementado para el mismo sistema según se detalla en [13], se puede observar claramente que se obtienen unas soluciones de mayor calidad a nivel de costes C_p del sistema. Del total de configuraciones evaluadas, tan solo Lui-Chan consigue rebajar las 6.82 unidades de coste con sus 6.56. A pesar de ello, ambas son superadas en eficiencia por la solución genética que consigue soluciones de coste 6.54 para esos patrones de prueba. A nivel de tiempo en la obtención de soluciones, ninguna solución evolutiva consigue, para mundos virtuales pequeños, ejecuciones más rápidas que las realizadas por el algoritmo Lui-Chan. Esto no plantea ningún problema ya que es necesario recordar que se está trabajando en un sistema 3D en tiempo real, donde supuesto un “frame-rate” exigido de 25fps se han de completar dibujos de escena a razón de 40ms. De estos 40ms, tras diferentes pruebas, una máquina con las características hardware

descritas anteriormente necesita del orden de 12-15ms para realizar las tareas de procesamiento de polígonos y procesado de píxel con el fin de dibujar una escena en un “frame”. Aun tomando decisiones pesimistas como ejecutar el algoritmo de particionado en los 25-28ms. restantes, o etapa de aplicación, para completar agrupaciones por cada frame existe tiempo más que suficiente para llevar su cálculo a cabo. Estos últimos datos dependen en gran medida del tamaño de la escena en número de polígonos, tipo de representación a utilizar, texturizados, efectos y luces. Este hecho ya se contempla en [13] donde se afirma que para mundos virtuales SMALL soluciones por debajo de los 10ms son clasificadas como excelentes.

C. Resultados en mundos virtuales LARGE

En el caso de grandes DVEs (LARGE) la evaluación de los resultados obtenidos ha sido un poco diferente. En primer lugar, no existe un valor óptimo de coste Cp con el que comparar las diferentes metaheurísticas. El motivo de esta inexistencia se basa en el hecho de la imposibilidad de evaluar las 82500 diferentes soluciones de una escena virtual de ese tipo. Además por [12], para esta configuración no se va a exigir que para cada uno de los “frames” se ejecute el algoritmo de partición, hecho que como se comprobará sería inviable.

La siguiente tabla muestra los resultados:

| | Computación t(s) | Coste Sistema Cp |
|--------------------------|------------------|------------------|
| Lui-Chan Alg. | 30.93 | 1637.04 |
| Enfriamiento Alg. | 6.35 | 1707.62 |
| Genéticos Alg. | 6.60 | 1832.21 |

(a) Avatares en Distribución Uniforme

| | Computación t(s) | Coste Sistema Cp |
|--------------------------|------------------|------------------|
| Lui-Chan Alg. | 32.17 | 3460.52 |
| Enfriamiento Alg. | 13.79 | 2628.46 |
| Genéticos Alg. | 14.59 | 2825.64 |

(b) Avatares en Distribución Skewed

| | Computación t(s) | Coste Sistema Cp |
|--------------------------|------------------|------------------|
| Lui-Chan Alg. | 43.31 | 5903.80 |
| Enfriamiento Alg. | 29.62 | 4697.61 |
| Genéticos Alg. | 29.20 | 4905.93 |

(c) Avatares en Distribución Clustered

Tabla 3. Resultados en mundos virtuales LARGE

Tal y como muestra la tabla 3, excepto para distribuciones uniforme de avatares, las soluciones evolutivas presentadas en este trabajo han logrado, en las diferentes pruebas a que se ha visto sometidas, soluciones con menor coste que las obtenidas por la heurística Lui-Chan. Para este caso de excepción, la solución basada en SA obtiene soluciones con un exceso de simplemente un 4.31% con respecto a las alcanzadas con la implementación Lui-Chan. Además de esta evidente mejora en cuanto a calidad de las soluciones obtenidas, las aproximaciones evolutivas son más rápidas, en el cálculo de agrupaciones, que las análogas Lui-Chan. Así consiguen reducir en un

79.5%, 57.13% y 32.58% los tiempos de ejecución para las distribuciones uniforme, sesgada (skewed) y focalizada (clustered), respectivamente.

Ante esto, se han de destacar dos importantes hechos. El primero de ellos se basa en que en grandes mundos virtuales, al contrario con lo que ocurría en mundos SMALL, el tiempo de ejecución en la aplicación del procedimiento de asignación de clientes-servidores es el parámetro más importante para mantener una eficiencia media en el sistema, ya que se ha de evitar mantener a un servidor (usualmente uno de los de la simulación) permanentemente calculando particiones. Por lo cual y debido a que todas las posibles soluciones son aproximaciones a la óptima, no se busca la mejor de todas sino aquellas que siendo suficientemente buenas se ejecuten más rápidamente.

Por otra parte, el hecho de que la distribución de avatares en un DVE convencional, por el comportamiento de los avatares en la escena, se realice usualmente en patrones sesgados (skewed) o focalizados (clustered) inclina, si cabe, todavía más la balanza hacia la eficacia de las soluciones presentadas en el este trabajo.

V. CONCLUSIONES Y TRABAJO FUTURO

Los entornos virtuales distribuidos (DVE) se están convirtiendo en una importante área de interés dentro del campo del procesamiento paralelo y distribuido. Estos sistemas se caracterizan por una heterogeneidad inherente que aparece en numerosos puntos de su arquitectura. El control óptimo de sus clientes remotos (avatares) interactuando en un mundo 3D compartido no es una tarea trivial.

En este trabajo, se han presentado nuevas soluciones al problema de la asignación eficiente de clientes a los servidores de la simulación.

En lugar de desarrollar una técnica ad-hoc para este problema NP-completo, tal y como han realizado otros grupos de investigación, se han adaptado a las especificaciones de éste dos técnicas basadas en computación evolutiva.

Los experimentos realizados, sobre una plataforma de evaluación de mundos virtuales, parecen indicar que las iniciales aproximaciones evolutivas presentadas obtienen soluciones de mejor calidad y en menor tiempo que las implementadas siguiendo el paradigma Lui-Chan, máxima referencia en la literatura de la materia.

Con ello, las técnicas presentadas logran que los grandes mundos virtuales que las incorporen, como mecanismo base de particionado, presenten mejor escalabilidad global y latencia en las comunicaciones.

Los resultados prometedores de estas primeras experiencias evolutivas dentro de este campo plantean numerosas líneas de futuro a acometer.

Dentro de los algoritmos genéticos y en grandes DVEs, donde el espacio de búsqueda es sumamente amplio, se van a utilizar estrategias de cruce CHC ([26]) así como el cruce propuesto por Whitley ([27]) para problemas altamente combinatorios.

A su vez, debido a la naturaleza multiobjetivo del problema del particionado sería de interés acometer sus soluciones mediante algoritmos evolutivos multiobjetivo.

Por último, siguiendo líneas de grupos de trabajo en el campo de la computación evolutiva, se está trabajando en una implementación paralela a la solución genética ([25]). En su aproximación inicial aprovechará los servidores de la simulación al basarse en un esquema maestro-esclavo con comunicaciones asíncronas. La plataforma completa a desarrollar será portada para la implementación futura de un multi-simulador de conducción distribuido para la Dirección General de Tráfico en el que el grupo investigador se encuentra actualmente trabajando.

REFERENCIAS

- [1] S.Singhal, M.Zyda. "Networked Virtual Environments" ACM Press, New York, 1999.
- [2] J.M Salles, R.Galli, A.C.Almeida, C.A.C. Belo, J. Rebordão "mWorld: A Multiuser 3D Virtual Environment" IEEE CGraphics 97 (Vol. 17, No. 2)
- [3] D.C.Miller, J.A. Thorpe. "SIMNET: The advent of simulator networking ". Proceedings of the IEEE, 83(8):1114-1123. August 1995
- [4] T.Nitta, K. Fujita, S.Cono "An Application Of Distributed Virtual Environment To Foreign Language". Kansas . IEEE October 18 - 21, 2000
- [5] Michael Lewis and Jeffrey Jacobson " Game Engines in Scientific Research ". Communications of the ACM., Vol 45. N°1 January 2002.
- [6] M.Abrash "Quake's game engine: The big picture"Dr. Dobb's. Journal.Spring. Year 1997.
- [7] Michael R. Macedonia. "A Taxonomy for Networked Virtual Environments". IEEE Multimedia, 4(1) 48-56. January-March 1997.
- [8] P.A.Berstein, V.Hadzilacos and N.Goodman." Concurrency, Control and Recovery in Database Systems". Addison-Wesley. 1997.
- [9] Frecon, M. Stenius. "DIVE: A scaleable network architecture for distributed virtual environments". Distributed Systems Eng., vol- 5. 1998.
- [10] Didier Verna, Yoann Fabre. "Urbi et Orbi: "Unusual Design and Implementation Choices for Distributed Virtual Environments" EPITA/LRDE 14-16. Kremlin. 2000.
- [11] 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.
- [12] John C.S. Lui, W.K. Lam, "General Methodology in Analysing the Performance of Parallel/Distributed Simulation under General Computational Graphs". ICSM 1999.
- [13] Jonh C.S. Lui, M.F. Chan. "An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems". IEEE TPDS,13-3, 03/2002.
- [14] J.C.Hu, I.Pyarali, D.C.Schmidt, "Measuring the Impact of Event Dispatching and Concurrency Models on Web Server Performance Over High-Speed Networks," IEEE GIC, November.1997.

- [15] Tomas A. Funkhouser. "Network Topologies for Scalable Multi-User Virtual Environments". Technical Report Bell Laboratories. 1996.
- [16] D.B. Anderson, J.W. Barrus, J.H. Howard, "Building multi-user interactive multimedia environments at MERL". IEEE Multimedia, 2(4):77-82, 1995.
- [17] F.C. Greenhagh "Awareness-based communication management in MASSIVE systems" Distributed Systems Engineering, vol 5, 1998.
- [18] Tam TszFirst. "Term Paper New Architect. for DVE Object Based Dynamic Group Multicast Framework". November 1997.
- [19] Peter T.S. Tam "Communication cost optimiz. and analysis in Distributed Virtual Environment", Technical report RM1026-TR98-0412. 1998.
- [20] V. Llorens, J. M. Orduña y J. Duato. "A Comparative Study of Heuristic Search Methods for Communication-Aware Task Mapping Techniques". XII Jornadas de Paralelismo. Valencia, 2001.
- [21] R. Duda, P. Hart, D. Stork. "Pattern Classification". Ed. Wiley Intescience Publication. 2000. p:567-580.
- [22] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, "Optimization by simulated annealing," Science, vol. 220, pp. 671-680, 1983.
- [23] P.V. Laarhoven, E. Aarts. "Simulated annealing: Theory and applications". Reidel Pub., Dordrecht, Holland. Año 1987.
- [24] Zbigniew Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Springer-Verlag, Second Edition, 1992.
- [25] Enrique Alba, José M. Troya "Analyzing synchronous and asynchronous parallel distributed genetic algorithms". Future Generation Computer Systems 17 (2001) 451-456.
- [26] L. J. Eshelman, "The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination," en Foundations of Genetic Algorithms (G. J. E. Rawlins, ed.), pp. 265-283, Morgan Kaufman, 1991.
- [27] D. Whitley, "The Genitor Algorithm and Selection Pressure: Why Ranked-Based Allocation of Reproductive Trail is Best," en Proceedings of the Third International Conference of Genetic algorithms (J. D. Schaffer, ed.), pp. 116-121, Morgan Kaufmann, Junio 1989.
- [28] Naohiko Yamaguchi, Masaharu Munetomo, Yoshiharu Sato "Load Balancing Routing with Genetic Algorithm Based on Link Load Metric". IPSJ JOURNAL Vol.43 No.07-038.
- [29] S.H. Zanakis, J.R. Evans, "Heuristic Optimization: Why, when and how to use it". Interfaces, Vol. 11, nº 5, October. 1981.