Un Sistema Híbrido MPI/OpenMP Basado en el Algoritmo NSGA-II para la Inferencia de Árboles Filogenéticos

Sergio Santander-Jiménez[1], Miguel A. Vega-Rodríguez[1], Juan A. Gómez-Pulido[1]y Juan M. Sánchez-Pérez[1].

Resumen— **Las metaheurísticas que hacen uso de técnicas de optimización multiobjetivo suelen presentar una mayor complejidad a nivel computacional al requerir mayores tiempos de ejecución para la generación y evaluación de aquellas soluciones Pareto que representen un consenso entre criterios en conflicto. Los modelos híbridos basados en la explotación de memoria distribuida y memoria compartida en clusters de computadores representan una alternativa interesante para llevar a cabo estas búsquedas en tiempos reducidos. En este artículo proponemos un estudio de un sistema híbrido maestro-esclavo basado en las tecnologías MPI y OpenMP para paralelizar el algoritmo Fast Nondominated Sorting Genetic Algorithm (NSGA-II), aplicado a uno de los problemas más importantes en la disciplina de la Bioinformática, la inferencia filogenética. Los experimentos realizados sobre un clúster compuesto por hasta 64 cores muestran que el sistema puede alcanzar prometedores factores de aceleración y eficiencias al realizar análisis filogenéticos sobre bases de datos de nucleótidos complejas.**

Palabras clave— **Inferencia Filogenética, Nondominated Sorting Genetic Algorithm, Computación con Clusters, Optimización Multiobjetivo.**

I. Introducción

El aprovechamiento de los recursos hardware disponibles resulta de vital importancia para afrontar los complejos retos computacionales que supone el procesamiento y análisis de problemas de optimización reales en diversos campos como las Telecomunicaciones o la Biología. Esta necesidad aumenta cuando se considera la resolución de problemas conforme a objetivos conflictivos entre sí, lo que implica el uso de técnicas de optimización multiobjetivo para obtener no una, sino un conjunto de soluciones conforme a los objetivos planteados [1]. Por ello, las metaheurísticas multiobjetivo paralelas que permitan la resolución de estos problemas deben ser diseñadas para explotar las características de la estructura hardware. La computación con clusters es uno de los paradigmas de computación paralela más empleados al permitir al investigador la búsqueda y obtención de soluciones a estos problemas en tiempos razonables por un coste monetario medio. En este sentido, un diseño algorítmico híbrido que permita hacer uso conjuntamente de las posibilidades que ofrecen los esquemas de memoria distribuida (entre nodos) y memoria compartida (entre cores de un mismo nodo) puede resultar de gran utilidad a la hora de llevar cabo este tipo de análisis. Según Capello y Etiemble [2], se pueden distinguir varios factores que determinan el éxito de una propuesta híbrida:

1. El nivel de paralelismo sobre memoria compartida que ofrece el diseño de los bucles principales del algoritmo.
2. El coste por comunicaciones entre procesos situados en diferentes nodos, de acuerdo a la arquitectura de comunicaciones del clúster.
3. Los patrones de acceso a memoria.
4. Equilibrio en el rendimiento de los diferentes componentes de la arquitectura.

En este estudio presentamos un esquema híbrido MPI/OpenMP para paralelizar uno de los algoritmos de referencia en optimización multiobjetivo, el NSGA-II [3], aplicándolo a uno de los problemas reales más importantes del ámbito de la Bioinformática: la inferencia filogenética [4]. Los árboles filogenéticos representan hipótesis sobre la evolución de las especies consideradas en la entrada del procedimiento, definiéndose diferentes criterios para determinar la calidad de la topología inferida. La obtención de árboles de calidad precisa el uso de complejos operadores y heurísticas, por lo que las aproximaciones clásicas basadas en la paralelización de los bucles de evaluación deben ser extendidas a un nivel superior y permitir a los procesos asignar varios hilos de ejecución para repartir la carga algorítmica que supone tanto la generación como la evaluación de soluciones. Nuestra propuesta se basa en un esquema maestro esclavo para reconstrucción filogenética multiobjetivo conforme a los criterios de máxima parsimonia y máxima verosimilitud [4]. Esta aproximación ha sido probada sobre cuatro bases de datos reales de nucleótidos y los resultados obtenidos, tanto a nivel de soluciones como a nivel de aceleración y eficiencia, comparadas con las propuestas de otros autores.

Este artículo se organiza del siguiente modo. En la siguiente sección, se describe una breve introducción al problema y su complejidad. La Sección III describe la aproximación híbrida propuesta, aplicándola a la inferencia de árboles filogenéticos. En la Sección IV presentamos y razonamos los resultados experimentales obtenidos, comparándolos con los publicados por otros autores. Finalmente, la Sección V presenta las conclusiones que arroja el estudio y define líneas de trabajo futuro.

[1]Universidad de Extremadura, Dept. Tecnología de los Computadores y de las Comunicaciones, Escuela Politécnica. Campus Universitario s/n, 10003, Cáceres, España. {sesaji, mavega, jangomez, sanperez}@unex.es

II. Introducción a la Inferencia Filogenética

La Filogenética es la disciplina de la Biología que tiene por objeto la descripción de las relaciones ancestro-descendiente que se han dado entre las especies que han poblado la tierra a lo largo de la historia evolutiva [5]. Con este propósito, se definen técnicas de estimación que toman en su entrada un conjunto de n secuencias de N caracteres que representan características moleculares de los organismos a procesar, construyendo en su salida una hipótesis de la evolución de las especies. Esta hipótesis viene dada por una estructura arborescente en la que las hojas representan las especies resultantes del proceso evolutivo y los nodos internos posibles ancestros comunes: el *árbol filogenético*.

La cuestión de determinar la calidad de los árboles filogenéticos inferidos dio lugar a la propuesta de métodos basados en criterios de optimalidad, que definen una función objetivo que gobierna el proceso de búsqueda de soluciones. Estos métodos presentan una complejidad de tipo NP-Completo [6] debido a dos factores fundamentales [7]:

1. Se produce un crecimiento exponencial del espacio de búsqueda de topologías filogenéticas a medida que aumenta el número de especies consideradas. Cuantitativamente, para un número n de organismos a procesar, el número de topologías filogenéticas posibles viene definido por la siguiente ecuación [8]:

$$\frac{(2n-5)!}{(n-3)!2^{n-3}} \tag{1}$$

   Actualmente, el principal interés investigador en Filogenética es el tratamiento de grandes conjuntos de datos que consideren un amplio número de organismos, por lo que es menester considerar este factor a la hora de diseñar estrategias de búsqueda eficientes que permitan explotar el hardware disponible.
2. Los bucles de evaluación de topologías precisarán mayores tiempos de procesamiento y consumo de memoria al considerar cadenas moleculares de mayor longitud. Si además se consideran diferentes criterios conflictivos que deben satisfacerse simultáneamente, como ocurre al aplicar técnicas de optimización multiobjetivo, la complejidad del proceso de evaluación se verá afectada, precisando mayores recursos computacionales para su cálculo.

Para afrontar el reto computacional que suponen estos métodos, es necesaria la definición de metaheurísticas paralelas que permitan obtener soluciones de calidad en tiempos adecuados. En este estudio presentamos una propuesta híbrida MPI/OpenMP para paralelizar una aproximación basada en el algoritmo NSGA-II para reconstrucción filogenética conforme a los criterios que se introducen a continuación: máxima parsimonia y máxima verosimilitud.

A. Máxima Parsimonia

El principio de máxima parsimonia se inspira en el razonamiento de Occam, por el cual se considera que la hipótesis más sencilla para explicar un determinado fenómeno será más plausible que otras posibles explicaciones. Por ello, los métodos basados en este criterio tratan de encontrar aquellas hipótesis evolutivas que minimicen el número de cambios a nivel molecular que se precisan para explicar la historia evolutiva [7]. Matemáticamente, dado un árbol filogenético τ, construido a partir de un conjunto de n secuencias de N nucleótidos alineados, el valor de parsimonia para τ viene dado por [9]:

$$P(\tau) = \sum_{i=1}^{N} \sum_{(a,b)\in B(\tau)} C(a_i, b_i) \tag{2}$$

donde (a, b) es la rama del conjunto B que relaciona los nodos a y b, a_i y b_i el valor del i-ésimo carácter de las secuencias asociadas a a y b, y $C(a_i, b_i)$ el coste de evolucionar del estado a_i al b_i. Desde esta perspectiva, se prefieren aquellas topologías que *minimicen* la parsimonia, al representar una hipótesis más sencilla de cómo transcurrió el proceso evolutivo que dio origen a las especies de entrada.

B. Máxima Verosimilitud

Los métodos basados en la función estadística de verosimilitud se definieron con objeto de obtener aquel árbol filogenético que represente la hipótesis más probable de la historia evolutiva de las especies de entrada [7]. Se trata de un método cuya complejidad radica en la cantidad de parámetros que considera, a saber: la topología del árbol filogenético, los tiempos evolutivos entre especies emparentadas (representados por los valores de longitud de ramas) y el modelo de evolución molecular, que indica las probabilidades de cambio a nivel de nucleótido en las secuencias de entrada [4].

Considérese τ una topología filogenética, D el conjunto de secuencias de N caracteres, D_i el estado del carácter i-ésimo de las secuencias y m el modelo evolutivo. La verosimilitud se define como la probabilidad condicional de los datos dados τ y m [4]:

$$L[D, \tau, m] = \Pr[D|\tau, m] = \prod_{i=1}^{N} \prod_{j=1}^{B} (r_i t_j)^{n_{ij}} \tag{3}$$

donde B es el conjunto de ramas de τ, r_i la probabilidad de cambio del estado i, t_j la longitud de la rama j y n_{ij} el número de cambios de estado en j para el carácter i. Se prefieren aquellas topologías que *maximicen* la función de verosimilitud, pues representarán una hipótesis más probable de la evolución de las especies.

III. Fast Nondominated Sorting Genetic Algorithm (NSGA-II)

Uno de los algoritmos más empleados al aplicar optimización multiobjetivo para la resolución de problemas es el NSGA-II, propuesto por Deb et al. [3].

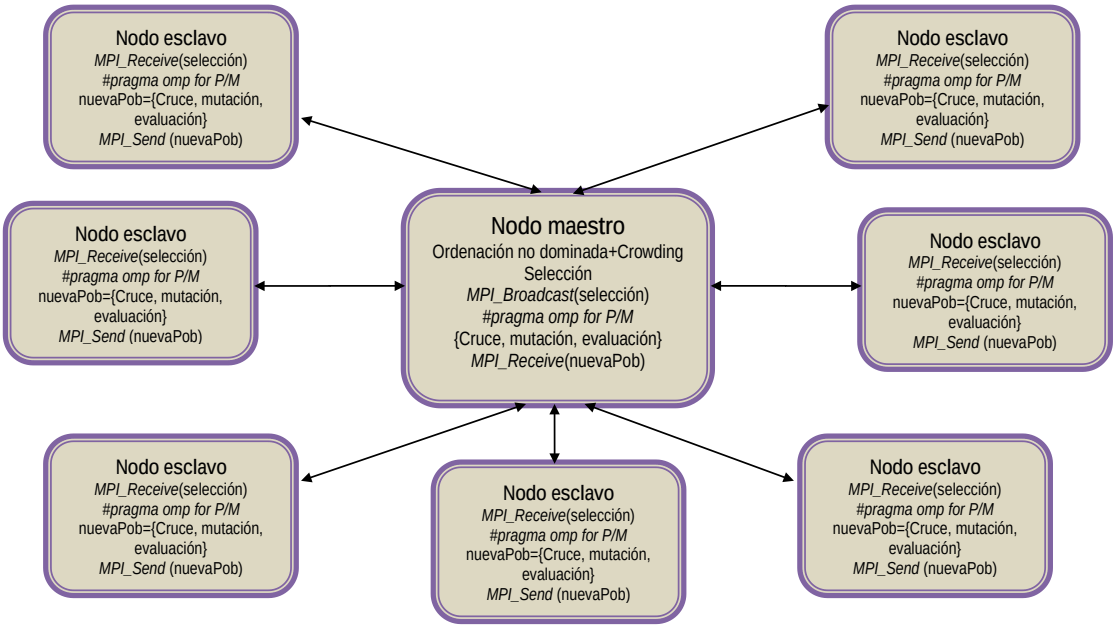

Fig. 1: Esquema híbrido MPI-OpenMP para un clúster de 8 nodos.

Estos autores extendieron la definición clásica de algoritmo genético y aplicaron conceptos multiobjetivo como el de dominancia para distinguir la calidad de las soluciones. Diremos que una solución x domina a una y si x presenta mejores o iguales valores que y en todos los objetivos y, al menos, es mejor en uno de ellos. Este algoritmo define una población *offspring* aplicando los operadores clásicos de selección, cruce y mutación, que se combinará con la población padre. Esta población combinada será clasificada en función de la calidad de las soluciones desde un punto de vista multiobjetivo.

La principal novedad que aporta el NSGA-II es la introducción de tres propuestas que permiten mejorar la eficacia del algoritmo: ordenación rápida no dominada, ordenación por distancia de crowding y operador de comparación de soluciones por rango y crowding [3]. La ordenación no dominada permite asignar rangos a las soluciones en virtud del número de soluciones que la dominan. La ordenación interna de soluciones del mismo rango se realiza mediante el cálculo de la distancia de crowding para cada una de ellas. Este operador determina una estimación de la densidad de soluciones que rodean a la solución tratada. En base a estas dos propuestas se define el operador de comparación, que distingue soluciones según sus valores de rango y, en caso de pertenecer al mismo rango, según sus distancias de crowding.

A. Un Sistema Maestro-Esclavo Híbrido NSGA-II para la Inferencia Filogenética

Al ser uno de los algoritmos más empleados al aplicar técnicas de optimización multiobjetivo, existen en la literatura distintas propuestas para la paralelización del NSGA-II. Pueden distinguirse diversas aproximaciones maestro-esclavo síncronas y asíncronas planteadas con objeto de explotar arquitecturas clúster o grid [10] para la reducción de los tiempos de ejecución y la mejora de las soluciones. A su vez, existen otras propuestas que emplean modelo de islas para especializar un conjunto de procesos en la exploración de regiones diferentes del espacio de búsqueda y, de este modo, generar soluciones de mayor calidad [11] [12]. En el ámbito de la Filogenética pueden encontrarse distintas aproximaciones basadas en la primera estrategia, ejecutando el proceso maestro el código principal del algoritmo y apoyándose en los esclavos para reducir los tiempos de cómputo que precisa la evaluación de soluciones [7] [13].

En este estudio proponemos una aproximación híbrida MPI/OpenMP basada en el modelo maestro-esclavo para paralelizar una versión del algoritmo NSGA-II adaptada al problema de la inferencia filogenética. Debido a los dos factores de complejidad presentados en la Sección II que gobiernan el proceso de reconstrucción filogenética, resulta de utilidad plantear un escenario en que se aprovechen las características de las arquitecturas de tipo clúster para implementar un modelo de grano grueso que permita paralelizar no solo el proceso de evaluación de soluciones, sino también los pasos de reconstrucción de nuevas topologías filogenéticas. Esta necesidad radica en el hecho de que el proceso de generación de nuevos árboles filogenéticos implica el empleo de operadores de cruce y mutación específicos del problema y que presentan complejidades adicionales para mantener la coherencia de la hipótesis evolutiva descrita por la topología resultante. En la Figura 1 se puede encontrar un esquema representativo de la aproximación híbrida propuesta.

Considérese una arquitectura hardware compuesta por M nodos interconectados mediante una red de

comunicaciones, donde cada nodo contiene un máximo de N cores. Una ejecución del algoritmo considerando un tamaño de población P comenzará con la asignación de hasta M procesos MPI, de los cuales uno tomará el rol de maestro, siendo el resto esclavos que esperarán la asignación de tareas por parte del maestro. La inicialización de la población es llevada a cabo por el proceso maestro, asignando aleatoriamente árboles de un repositorio generado mediante técnicas de bootstrap [4], tras lo cual comienzan las tareas iniciales de una iteración típica del algoritmo. Estas tareas consisten en la ordenación rápida no dominada de la población y el cálculo de distancias de crowding. El proceso de selección de las topologías más prometedoras para generar la población offspring se realiza mediante torneo binario según los valores de rango y distancia de crowding calculados.

Las soluciones seleccionadas son enviadas mediante *MPI_Broadcast* a los procesos esclavos, que se encargan junto con el maestro de la generación y evaluación de P/M nuevos individuos cada uno. Este proceso de broadcasting comunica los valores de parsimonia y verosimilitud de los individuos, así como una representación de la topología en forma de cadena de caracteres que delimita las relaciones evolutivas entre especies mediante el uso de paréntesis (formato Newick) [4]. Mediante la definición de cláusulas *#pragma omp for*, se paralelizan usando de 1 a N hilos los pasos de cruce, mutación y evaluación de las topologías filogenéticas generadas conforme a los criterios de parsimonia y verosimilitud. El proceso de cruce se realiza aplicando el operador Prune-Delete-Graft Recombination [14] y la mutación mediante cambios topológicos Nearest Neighbour Interchange [4] y modificación aleatoria del valor de ramas siguiendo una distribución gamma.

Una vez concluidas estas tareas, los procesos esclavos comunican al maestro las nuevas topologías Newick y sus valores de parsimonia y verosimilitud. El maestro combinará la población actual con la offspring para generar la población combinada que será considerada en la siguiente iteración del algoritmo, actualizándose además la estructura de datos que contiene las soluciones Pareto más prometedoras encontradas. Bajo este modelo, se puede apreciar que el rendimiento del sistema se verá afectado por el impacto que suponen las sincronizaciones de hilos OpenMP y las comunicaciones entre procesos MPI.

IV. Resultados Experimentales

Para evaluar el rendimiento de la propuesta, se ha llevado a cabo una serie de experimentos sobre una arquitectura clúster compuesta por ocho nodos de cómputo, conteniendo cada uno procesadores Intel Xeon E5410 a 2,33GHz con 8 cores, 8GB de RAM y 6MB de caché por nodo. La aplicación fue compilada usando GCC 4.4.5 y utilizando las librerías MPICH2 1.4.1. Para incluir resultados estadísticamente relevantes, se han realizado cinco ejecuciones del algoritmo para cada conjunto de datos considerado usando configuraciones de 1, 2, 4, 8 (1 proceso MPI), 16 (2 procesos MPI con 8 hilos OpenMP cada uno), 32 (4 procesos MPI con 8 hilos OpenMP) y 64 (8 procesos MPI con 8 hilos OpenMP) hilos de ejecución.

La experimentación se ha realizado sobre cuatro bases de datos reales de nucleótidos [9]: $rbcL_55$, 55 secuencias de 1314 nucleótidos cada una del gen rbcL de diversas especies de plantas verdes. $mtDNA_186$, 186 secuencias de 16608 nucleótidos de ADN mitocondrial humano. $RDPII_218$, 218 secuencias de 4182 nucleótidos de ARN procariota. Y $ZILLA_500$, 500 secuencias de 759 nucleótidos del gen plástido rbcL. Una ejecución completa constará de 50 generaciones con un tamaño de población de 100 individuos, usando un 80 % de ratio de cruce y un 5 % de ratio de mutación, de acuerdo a las propuestas de otros autores [9] [13].

En la Tabla I se muestran los tiempos secuenciales (en valor promedio) de ejecución necesarios para cada una de las bases de datos. Los factores de aceleración y los valores de eficiencia obtenidos en función del número de hilos usados pueden encontrarse en la Tabla II. La representación gráfica de estos valores de aceleración con respecto al valor lineal teórico viene dado por la Figura 2.

TABLA I: Tiempos secuenciales.

Instancia	Tiempo de ejecución
$rbcL_55$	7654,09 segundos
$mtDNA_186$	71220,00 segundos
$RDPII_218$	87512,70 segundos
$ZILLA_500$	101955,00 segundos

TABLA II: Valores de aceleración y eficiencia.

	$rbcL_55$		$mtDNA_186$	
Hilos	Aceleración	Eficiencia	Aceleración	Eficiencia
1	1	100 %	1	100 %
2	1,82892	91,44580 %	1,84901	92,45057 %
4	3,28804	82,20101 %	3,63694	90,92339 %
8	6,06132	75,76646 %	6,16537	77,06711 %
16	10,16768	63,54797 %	10,71588	66,97422 %
32	16,54304	51,69700 %	19,71820	61,61937 %
64	28,19760	44,05875 %	36,22691	56,60454 %
	$RDPII_218$		$ZILLA_500$	
Hilos	Aceleración	Eficiencia	Aceleración	Eficiencia
1	1	100 %	1	100 %
2	1,80457	90,22864 %	1,84260	92,13020 %
4	3,53321	88,33030 %	3,63629	90,90723 %
8	6,72627	84,07833 %	7,05561	88,19515 %
16	12,65055	79,06592 %	13,25900	82,86875 %
32	22,81127	71,28522 %	23,55544	73,61074 %
64	42,75756	66,80869 %	43,50206	67,97197 %

De acuerdo a los valores indicados en la Tabla II, se puede apreciar un aumento de la eficiencia de la propuesta en virtud de la complejidad de la base de datos empleada. Para razonar los valores obtenidos, debemos considerar el impacto que suponen en la escalabilidad de la propuesta los factores sugeridos por Capello y Etiemble a la hora de analizar el rendimiento de sistemas híbridos MPI/OpenMP: los patrones de comunicación entre procesos MPI, la sincronización entre hilos OpenMP y el tiempo serie no paralelizable [2]. La Tabla III presenta un resumen del porcentaje que suponen estos factores en el tiempo total para cada una de las instancias en función

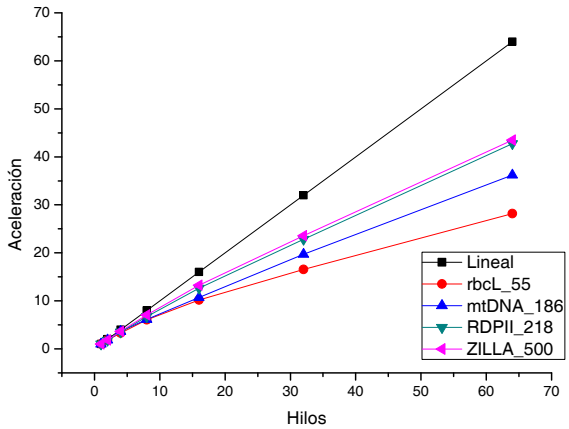

Fig. 2: Factores de aceleración logrados para cada conjunto de datos.

del número de hilos usados.

TABLA III: Impacto tiempos serie y comunicaciones.

	rbcL_55		*mtDNA_186*	
Hilos	$T_{serie} + T_{com}$	Porcentaje Tiempo total	$T_{serie} + T_{com}$	Porcentaje tiempo total
2	86,30514s	2,06223 %	92,15264s	0,23925 %
4	86,46489s	3,71435 %	92,33157s	0,47150 %
8	87,30726s	6,91390 %	92,61316s	0,80173 %
16	91,20175s	12,11521 %	133,05004s	2,00189 %
32	94,06098s	20,32969 %	134,55269s	3,72527 %
64	98,99273s	36,46880 %	135,97292s	6,91643 %
	RDPII_218		*ZILLA_500*	
Hilos	$T_{serie} + T_{com}$	Porcentaje Tiempo total	$T_{serie} + T_{com}$	Porcentaje Tiempo total
2	105,90772s	0,21839 %	95,83395s	0,17320 %
4	106,09321s	0,42833 %	96,12998s	0,34285 %
8	106,50273s	0,81858 %	96,43396s	0,66735 %
16	154,92396s	2,23950 %	175,93862s	2,28804 %
32	170,86701s	4,45381 %	236,82939s	5,47165 %
64	177,34783s	8,66488 %	246,98584s	10,53837 %

Para el caso de la instancia *rbcL_55*, estos factores suponen porcentajes mayores del 12 % del tiempo total de cómputo cuando se deben utilizar varios procesos MPI situados en diferentes nodos. En el caso de utilizar todo el hardware disponible, el impacto total es superior a 36 %, motivando una eficiencia del 44 %. Para el resto de instancias, el impacto que suponen estos factores se ve reducido debido a la ganancia que se obtiene al aplicar paralelismo. Conjuntos de datos más complejos motivan mejores valores de aceleración al aumentar el tamaño de la fracción paralelizable de la aplicación, de acuerdo a lo dispuesto en la ley de Amdalh [15]. Esto explica la obtención de valores de eficiencia superiores al 65 % al utilizar toda la estructura hardware sobre las instancias *RDPII_218* y *ZILLA_500*.

Analizando la Tabla III, se puede observar que, al aumentar el número de especies en el conjunto de entrada, se produce un nuevo incremento del porcentaje de tiempo que suponen los factores limitantes con respecto al tiempo total. A mayor número de especies, la caracterización de una topología filogenética mediante formato Newick implicará cadenas de caracteres de mayor tamaño que deben comunicarse a los procesos esclavos. Esto se traduce en un aumento del tiempo por comunicaciones que debe considerarse al realizar análisis filogenéticos sobre grandes conjuntos de datos. Con todo ello, la escalabilidad de la propuesta viene determinada por el tamaño del conjunto de datos de entrada (número de especies), el tiempo empleado en sincronizaciones y secciones críticas y el tiempo por comunicaciones a nivel de proceso. Para los conjuntos de datos analizados con más de 100 especies, el nivel de paralelismo obtenido compensa el impacto de los factores limitantes planteados, dando lugar a valores significativos de speedup y eficiencia para 16, 32 y 64 hilos de ejecución.

A. Comparación con Otros Autores

Cancino et al. desarrollaron en el año 2010 dos aproximaciones paralelas basadas en Computación con Clusters para minimizar los tiempos de ejecución requeridos por su software multiobjetivo para reconstrucción filogenética, PhyloMOEA [13]. Para ello hicieron uso del modelo maestro esclavo para distribuir las tareas de evaluación de soluciones entre diferentes nodos de cómputo, desarrollando dos versiones: una basada puramente en MPI y otra híbrida que implementaba paralelismo de grano fino con OpenMP para reducir los tiempos de cómputo de la función de verosimilitud (véase la Ecuación 3). La Tabla IV presenta una comparación de los factores de aceleración obtenidos por nuestra propuesta con los conseguidos por Cancino et al. usando 16 hilos de ejecución.

TABLA IV: NSGA-II - PhyloMOEA. Aceleración.

Instancia	NSGA-II	PhyloMOEA MPI	PhyloMOEA Híbrido
rbcL_55	**10,16**	7,3	8,3
mtDNA_186	**10,71**	7,4	8,5
RDPII_218	**12,65**	9,8	10,2
ZILLA_500	**13,25**	6,7	6,3

Los resultados mostrados en la Tabla IV indican que los valores de aceleración de nuestra aproximación híbrida resultan mejores que los obtenidos por las dos versiones de PhyloMOEA para los cuatro conjuntos de datos empleados. Se puede apreciar además cómo en el caso de la instancia más compleja, *ZILLA_500*, nuestra propuesta es capaz de doblar los factores de aceleración obtenidos por estos autores, lo que pone de manifiesto la bondad de aplicar el esquema paralelo estudiado en este artículo al problema de la inferencia filogenética, extendiendo la aplicación clásica de OpenMP para paralelizar las funciones de evaluación a un nivel superior.

B. Resultados Filogenéticos

En este apartado mostramos los resultados filogenéticos obtenidos por la propuesta para cada base de datos haciendo uso del modelo evolutivo $HKY85 + \Gamma$ [16]. La Tabla V muestra los valores de árboles más parsimoniosos y verosímiles generados en una ejecución media del algoritmo, así como una comparativa con los mejores valores de parsimonia y verosimilitud obtenidos usando PhyloMOEA [17]. Nuestra aproximación genera árboles de mejor verosimilitud en todas las instancias, así como árboles más parsimoniosos para el caso de las instancias

$mtDNA_186$ y $RDPII_218$.

TABLA V: NSGA-II - PhyloMOEA. Resultados Filogenéticos.

NSGA-II $HKY85+\Gamma$				
	Árbol más parsimonioso		Árbol más verosímil	
Instancia	Parsimonia	Verosimilitud	Parsimonia	Verosimilitud
$rbcL_55$	**4874**	-21927,73	4889	**-21864,07**
$mtDNA_186$	**2434**	-40005,96	2451	**-39895,34**
$RDPII_218$	**41525**	-137821.50	42825	**-134500.70**
$ZILLA_500$	**16219**	-81614,40	16313	**-80985,58**
PhyloMOEA $HKY85+\Gamma$				
Instancia	Mejor valor de parsimonia		Mejor valor de verosimilitud	
$rbcL_55$	**4874**		-21889,84	
$mtDNA_186$	2437		-39896,44	
$RDPII_218$	41534		-134696,53	
$ZILLA_500$	**16219**		-81018,06	

La evaluación del rendimiento multiobjetivo del algoritmo se ha llevado a cabo mediante la métrica de hipervolumen, que indica el porcentaje del espacio de soluciones dominado por las soluciones Pareto halladas. La Tabla VI presenta los valores de hipervolumen en promedio junto con los puntos de referencia necesarios para su cálculo. Las soluciones encontradas cubren un porcentaje significativo del espacio de soluciones, concretamente, más del 68 % para $rbcL_55$, $mtDNA_186$ y $ZILLA_500$ y 70 % para $RDPII_218$.

TABLA VI: Métrica de Hipervolumen.

Puntos de referencia				
	Mínimo		Máximo	
Instancia	Punto de referencia		Punto de referencia	
	Pars.	Verosim.	Pars.	Verosim.
$rbcL_55$	4774	-21569,7	5279	-23551,4
$mtDNA_186$	2376	-39272,2	2656	-43923,9
$RDPII_218$	40658	-132739,9	45841	-147224,6
$ZILLA_500$	15893	-79798,0	17588	-87876,4
Hipervolumen				
Instancia	Media(%)		Desv. estándar	
$rbcL_55$	68,75473		0,00708	
$mtDNA_186$	68,77644		0,00196	
$RDPII_218$	70,38929		0,00067	
$ZILLA_500$	68,71378		0,00016	

V. Conclusiones

En este artículo hemos presentado una aproximación híbrida MPI/OpenMP basada en uno de los algoritmos de referencia en optimización multiobjetivo, NSGA-II, para reconstrucción filogenética de acuerdo a los principios de parsimonia y verosimilitud. La principal característica que diferencia nuestra propuesta de otras es el aprovechamiento de la tecnología OpenMP para integrar un paralelismo de grano grueso que englobe los procesos de cruce, mutación y evaluación de soluciones. Los experimentos realizados sobre cuatro bases de datos reales de nucleótidos han puesto de manifiesto los factores que determinan la escalabilidad de los sistemas híbridos, consiguiendo unos valores de aceleración y eficiencia significativos para hasta 64 hilos de ejecución, conforme el nivel de paralelismo obtenido compensa el impacto por comunicaciones entre procesos y sincronización entre hilos.

Como trabajo futuro, se abordarán estrategias paralelas que hagan uso de otras arquitecturas hardware, como multiprocesadores many-cores y GPUs. Asimismo, se planteará el diseño de equipos paralelos de algoritmos bioinspirados con el propósito de aplicar paralelismo para la obtención de mejores soluciones al problema de la inferencia filogenética multiobjetivo.

Agradecimientos

Agradecemos al Gobierno de Extremadura por la ayuda GR10025 proporcionada al grupo de investigación TIC015, así como a la Fundación Valhondo por el apoyo económico ofrecido a Sergio Santander-Jiménez para hacer posible este trabajo.

Referencias

[1] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. New York: John Wiley & Sons, ISBN: 978-0-471-87339-6, 2001.

[2] F. Capello y D. Etiemble, "MPI versus MPI+OpenMP on the IBM SP for the NAS Benchmarks", en *Proc. of the 2000 ACM/IEEE conference on Supercomputing*. IEEE Computer Society, 2000, Artículo no. 12.

[3] K. Deb, A. Pratap, S. Agarwal y T. Meyarivan, "A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II", *IEEE Trans. Evol. Comput*, vol. 6, pp. 182-197, 2002.

[4] J. Felsenstein, *Inferring Phylogenies*. Sunderland, Massachusetts: Sinauer, ISBN: 0-87893-177-5, 2004.

[5] D. Swofford, G. Olsen, P. Waddell, P y D. Hillis, "Phylogenetic Inference", *Molecular Systematics*. Sinauer, 1996, vol. 2, pp. 407-514.

[6] D.A. Bader, U. Roshan, A. Stamatakis y C.-W. Tseng, "Computational Grand Challenges in Assembling the Tree of Life: Problems and Solutions", *Advances in Computers*, vol. 68, pp. 127-176, 2006.

[7] Q. Snell, M. Whiting, M. Clement y D. McLaughlin, "Parallel phylogenetic inference", en *Proc. of the 2000 ACM/IEEE conference on Supercomputing*. IEEE Computer Society, 2000, Artículo no. 35.

[8] L. Poladian y L. Jermiin, "Multi-Objective Evolutionary Algorithms and Phylogenetic Inference with Multiple Data Sets", Soft Computing, vol. 10, no. 4, pp. 359–368, 2006.

[9] W. Cancino y A.C.D. Delbem, "A Multi-Objective Evolutionary Approach for Phylogenetic Inference", en *Evolutionary Multi-Criterion Optimization*. LNCS, Berlin / Heidelberg: Springer, 2007, vol. 4403, pp. 428-442.

[10] J.J. Durillo, A.J. Nebro, F. Luna y E. Alba, "A Study of Master-Slave Approaches to Parallelize NSGA-II", en *11th International Workshop on Nature Inspired Distributed Computing*. IEEE Computer Society, 2008, pp. 1-8.

[11] K. Deb, P. Zope y A. Jain, "Distributed Computing of Pareto-Optimal Solutions Using Multi-Objective Evolutionary Algorithms", en *Proc. of the Second International Conference on Evolutionary Multi-Criterion Optimization*. LNCS, Berlin / Heidelberg: Springer, 2003, vol. 2632, pp. 534-549.

[12] J. Branke, H. Schmeck, K. Deb, y M. Reddy S, "Parallelizing Multi-Objective Evolutionary Algorithms: Cone Separation", en *Congress on Evolutionary Computation, CEC2004*. IEEE Computer Society, 2004, vol. 2, pp. 1952-1957.

[13] W. Cancino, L. Jourdan, E.G. Talbi y A.C.D. Delbem, "Parallel Multi-Objective Approaches for Inferring Phylogenies", en *Proc. of EVOBIO'2010*. LNCS, Berlin / Heidelberg: Springer, 2010, vol. 6023, pp. 26-37.

[14] C. Cotta y P. Moscato, "Inferring Phylogenetic Trees Using Evolutionary Algorithms", en *Parallel Problem Solving From Nature VII*. LNCS, Berlin / Heidelberg: Springer, 2002, vol. 2439, pp. 720-729.

[15] M. D. Hill y M. R. Marty, "Amdahl's Law in the Multicore Era", *IEEE Computer*, vol. 41, no. 7, pp. 33-38, July 2008.

[16] M. Hasegawa, H. Kishino y T. Yano, "Dating of the Human-Ape Splitting by a Molecular Clock of Mitochondrial DNA", Journal of Molecular Evolution, vol. 22, pp. 160-174, 1985.

[17] W. Cancino y A.C.D. Delbem, "A Multi-Criterion Evolutionary Approach Applied to Phylogenetic Reconstruction", en *New Achievements in Evolutionary Computation*. InTech, February 2010, pp. 135-156, ISBN: 978-953-307-053-7.