

ALGORITMOS EVOLUTIVOS DISTRIBUIDOS: PROPUESTAS QUE COMBINAN PREVENCIÓN DE INCESTO Y MULTIPLICIDAD

Alfonso H., Minetti G., Salto C.
Facultad de Ingeniería - Universidad Nacional de La Pampa
Calle 110 esq. 9 – General Pico (6360) – La Pampa – Rep. Argentina
e-mail: {alfonsoh, minettig, saltoc}@ing.unlpam.edu.ar

Resumen

En la actualidad los *algoritmos evolutivos* (AEs) se usan para buscar soluciones a problemas complejos para los cuales otras técnicas pueden insumir mucho tiempo y que, por lo general, proveen una única solución óptima.

Distintos avances sobre los AEs incluyen técnicas para prevenir el incesto y nuevos métodos de multirecombinación. Algunos de los métodos de multirecombinación permiten múltiples crossovers sobre una pareja seleccionada para la cruce, y otros los aplican sobre un conjunto de más de dos padres.

La inclusión del paralelismo y la distribución en el diseño de los AEs ha sido muy importante dando lugar a mecanismos de búsqueda y optimización mejorados: *algoritmos evolutivos paralelos*. Esto permite disponer de una mayor cantidad de recursos computacionales para alcanzar los resultados de forma más rápida por medio de un trabajo cooperativo.

El presente trabajo muestra cómo estas tendencias pueden resolver un problema de optimización complejo, en este caso el 3-SAT, de forma más eficiente y eficaz.

Palabras claves—algoritmos evolutivos, multiplicidad de crossover y padres, paralelismo, distribución.

Evento: Workshop de Agentes y Sistemas Inteligentes

I. INTRODUCCIÓN

Los AEs son utilizados frecuentemente como mecanismos de optimización. Trabajan sobre una población de individuos en lugar de hacerlo con una única solución. En los últimos años, se han introducido algunas mejoras en estos algoritmos que incluyen características de multiplicidad y prevención de incesto, como también la posibilidad de ejecutar en forma distribuida y/o paralela.

La opción de multiplicidad permite recombinar una pareja de padres o múltiples padres [1, 12, 13, 14]. Su aplicación contribuye a que la población final se encuentre más centrada alrededor del óptimo. Esto es importante cuando se requiere de múltiples alternativas cercanas a la mejor solución.

La prevención de incesto fue propuesta inicialmente por Eshelman y Shaffer [10] y mostró sus beneficios para evitar la convergencia prematura. Este método evita la cruce de pares de soluciones similares basado en la distancia de Hamming. La prevención de incesto fue extendida (EIP - Extended Incest Prevention) en trabajos previos [4, 15] al mantener información sobre los ancestros y modificar la selección para la reproducción y, de esta manera, prevenir la cruce de individuos pertenecientes a la misma “familia”, por un número predefinido de generaciones.

La implementación de los AEs en entornos paralelos y/o distribuidos ofrecen ventajas cuando los AEs trabajan con poblaciones de gran tamaño y/o con cromosomas de considerable longitud, lo cual supone una significativa utilización de recursos computacionales, como memoria física y tiempo de procesador. El tiempo de ejecución se puede disminuir al reducir el número de evaluaciones para alcanzar una solución o al ejecutar el algoritmo en un ambiente distribuido y paralelo. Los modelos paralelos de estos algoritmos, llamados *algoritmos evolutivos paralelos* (AEP), son interesantes porque consiguen ambos objetivos, ya que modifican el comportamiento típico del algoritmo secuencial equivalente mediante el uso de una población estructurada –una distribución espacial de individuos ya sea en la forma de un conjunto de islas [22] o de una grilla de difusión [18, 21].

El presente trabajo trata con los AEs que incorporan prevención de incesto y multiplicidad, ejecutados de forma secuencial y paralela, con el objetivo de resolver un problema NP-completo tradicional: el problema del 3-SAT.

La organización de este trabajo se detalla a continuación. Primero, se define el problema del MaxSat. En las siguientes tres secciones se introducen los conceptos relacionados a prevención de incesto, multirecombinación, paralelismo y distribución. Luego, la especificación de varios experimentos que utilizan un paquete que implementa un modelo unificado de AE paralelo, con la correspondiente descripción de los experimentos y análisis de los resultados alcanzados. Finalmente, un delineamiento de las principales conclusiones extraídas.

II. PROBLEMA DE MAXSAT

El problema de satisfacción SAT es uno de los problemas NP-completos más paradigmáticos sobre el cual se han realizado diversos estudios teóricos y sobre el que se fundamentan diversas aplicaciones [5]. SAT se basa sobre un conjunto de variables booleanas x_1, x_2, \dots, x_n y una fórmula booleana, $f : \mathbf{B}^n \rightarrow \mathbf{B}$, $\mathbf{B} = \{0,1\}$ siendo la pregunta a resolver si existe una asignación de valores posibles a la variable $x = (x_1, x_2, \dots, x_n) \in \mathbf{B}^n$ de modo que satisfaga f , o sea que $f(x)=1$. Una instancia SAT es llamada *satisfecha* si existe tal valoración para x , caso contrario se dice que la instancia es *insatisfecha*. La fórmula f se puede expresar como una *fórmula normal conjuntiva* (CNF) sin pérdida de generalidad [23] de la siguiente forma:

$$f(x) = c_1(x) \wedge c_2(x) \wedge \dots \wedge c_m(x)$$

donde cada cláusula c_i es una disyunción de literales, y un literal es una variable o su negación.

La clase k -SAT agrupa aquellas instancias cuyas cláusulas contienen exactamente k literales distintos. Los problemas pertenecientes a la clase 2-SAT son resolubles en tiempo polinomial, pero aquellos problemas que pertenecen a clases k -SAT con $k \geq 3$ son NP-completos [16].

En este trabajo se ha centrado el estudio sobre la clase 3-SAT, asumiendo por lo tanto que las cláusulas consisten de exactamente tres literales ($k=3$) y el tamaño del problema variará acorde al incremento de la cantidad de variables que se considere ($n=30, 40, 50, \dots$). Por otra parte, es necesario aclarar que las instancias seleccionadas son todas instancias satisfacibles cuyo óptimo es conocido, de esta forma será posible medir la performance del algoritmo por el porcentaje de casos donde la solución es alcanzada. Estas instancias se han generado usando el generador `mkcnf.c` desarrollado por Allen van Gelder que se basó sobre el `mwff.c` de Bart Selman [20]. El generador `mkcnf.c` puede forzar fórmulas satisfacibles y mucho más duras que las producidas por el generador `mwff.c`. Mitchell et al. [19] encontraron que la fase de transición más dura para las instancias del problema se encuentran cuando la siguiente relación se mantiene entre el número de cláusulas y el número de variables, $k = 4.3 \cdot n$.

Varios AEs han sido desarrollados para resolver el problema SAT, algunos de ellos figuran en [17] y [8]. La representación más ampliamente usada en estos AEs es la binaria, donde cada variable de la fórmula SAT puede ser representada por un gen que puede tomar uno de dos valores: 0 (falso) o 1 (verdadero). En este caso la función de fitness más comúnmente usada es contar el número de cláusulas satisfechas.

III. PREVENCIÓN DE INCESTO EXTENDIDA

La extensión del concepto de incesto está fuertemente vinculada con la relación familiar de los miembros afectados a la reproducción. Para prevenir incesto, EIP sólo permite la recombinación de individuos sin ancestros comunes. Para construir el conjunto de padres bajo EIP, se eligen individuos aleatoriamente desde la población, pero sólo se cruzarán de no mantener relación de parentesco, por lo tanto se modifica el efecto del mecanismo de selección sobre la población. De esta manera, se reduce el intercambio de material genético similar y se mantiene la diversidad poblacional en un grado conveniente, ya que una alta diversidad tiene un efecto negativo: disminuir la velocidad del proceso de búsqueda. Consecuentemente, cada individuo mantiene información de sus ancestros.

En el proceso evolutivo existen dos tópicos fuertemente relacionados: diversidad poblacional y presión selectiva forzada por el mecanismo. Ambos inciden en la exploración del espacio de búsqueda y en la explotación de la información. La selección juega un rol importante debido a que una fuerte presión selectiva puede producir una convergencia prematura en tanto que una presión selectiva suave puede provocar una búsqueda poco efectiva.

IV. MULTIRECOMBINACIÓN

Tradicionalmente los AEs realizan una única recombinación sobre cada pareja después de la selección. Dicho procedimiento se lo conoce como el SCPC (Single Crossover Per Couple). Pero en la naturaleza cuando se lleva a cabo el proceso de reproducción se aplica muchas veces el crossover y la consecuencia es un número múltiple y variable de hijos. Multiple Crossover Per Couple (MCPC) [11] es un método de crossover que permite un número variable de hijos. De estos últimos es posible elegir uno o varios individuos para insertarlos en la próxima generación.

En la opción de Multiple Parents de Eiben (MP) [7, 9] la creación de hijos está basada sobre un muestreo más grande del espacio de búsqueda y consecuentemente se suministra mayor diversidad. Esto puede evitar la convergencia prematura. Eiben usó tres mecanismos de Scanning Crossover (SX) [6] generando un único hijo. Ellos son: Uniform Scanning Crossover (USX), Occurrence Based Scanning (OBSX) y Fitness Based Scanning (FBSX).

Combinando las ideas anteriores surge la opción Multiple Crossover on Multiple Parents (MCMP), el cual permite la repetida recombinación de múltiples padres usando SX. Es decir, se realizan n_1 crossovers sobre los n_2 padres seleccionados; de los n_1 hijos generados se seleccionan n_3 , en función a algún criterio, para insertarlos en la próxima generación. Es de esperar que la exploración y explotación del espacio del problema esté adecuadamente balanceada [14].

Cuando se eligen múltiples padres para realizar la recombinación también es posible aplicar EIP. De esta manera se da origen a una nueva opción evolutiva EIP-MCMP [15].

V. PARALELISMO Y DISTRIBUCIÓN

Las características interesantes que incluyen los AEPs son: (a) disminución del tiempo para ubicar una solución (algoritmos más rápidos), (b) reducción en el número de evaluaciones (costo de búsqueda), (c) posibilidad de tener poblaciones de mayor tamaño, (d) aumento de la calidad en las soluciones halladas y (e) la posibilidad de atacar problemas de mayor tamaño.

Las versiones de AEPs son menos propensas a la convergencia prematura con lo cual se mejora el proceso de búsqueda [1]. Los AEPs son interesantes porque no son simplemente “versiones más rápidas” de AEs secuenciales sino que además proporcionan un mecanismo de búsqueda distinto, frecuentemente mejor.

Un AEP extiende la versión secuencial incluyendo una fase de comunicación con un vecindario formado por otros AEs. Una forma de lograr la paralelización es considerar un modelo de selección descentralizado donde los individuos se agrupan espacialmente. Este agrupamiento resulta de particionar una única población en varias y da origen a islas de AEs que se ejecutan realizando intercambios esporádicos de individuos. Dichas islas pueden utilizar: (a) algoritmos cuya evolución básica es secuencial y panmíctica (todos los individuos pertenecen a una misma población y se pueden cruzar entre sí) [24], (b) algoritmos celulares o de otra clase. El tipo de AE distribuido a utilizar lo determina la política de migración, la cual define: la topología de las islas, los individuos a intercambiar, la sincronización entre las subpoblaciones y la forma en que se integran los individuos en la subpoblación destino. Existe un consenso respecto a utilizar una topología anillo debido a su simplicidad, fácil implementación y ganancia en tiempo de comunicación.

VI. DESCRIPCIÓN DE LOS EXPERIMENTOS

En esta sección se detallan las distintas opciones algorítmicas desarrolladas para resolver el problema del 3-SAT, como así también su parametrización.

Estas opciones algorítmicas se ejecutan en forma secuencial y paralela. En ambos casos, además del AE simple (Simple), se aplica el control de incesto extendido (EIP), la característica de multiplicidad (MCMP- n_1 - n_2) y la combinación de los dos últimos (EIP-MCMP- n_1 - n_2).

Para llevar a cabo este trabajo se utilizó una versión modificada del paquete de software MALLBA que pertenece a un proyecto homónimo de las universidades españolas de Málaga, La Laguna y Barcelona [2, 3].

Para llevar a cabo esta experimentación, al paquete MALLBA se introdujeron las modificaciones necesarias para implementar la prevención extendida de incesto y características de multiplicidad. A cada individuo se le asoció un identificador unívoco y una lista con sus padres para corroborar, antes del crossover, que no exista relación familiar. Para aplicar la característica de multiplicidad, además de incorporar un nuevo tipo de crossover (el scanning crossover), se modificó el funcionamiento del operador de crossover de modo de generar n_1 hijos por pareja y luego seleccionar una cantidad n_3 de ellos. Por último, se alteró el control de terminación del algoritmo para que el mismo pare cuando se halle el valor óptimo o se alcance el número máximo de generaciones.

Instancia	Nro. de Var. (n)	Nro. de Cláusulas (k)
1	30	129
4	40	172
5	40	172
7	50	215

Tabla 1. Instancias usadas para la experimentación

Se consideraron las 4 instancias mostradas en la Tabla 1 y se realizaron ejecuciones para las distintas combinaciones (n_1 , n_2) de números de crossovers y de padres consistentes de 30 ejecuciones tanto para las versiones secuenciales como paralelas. Cada experimento se realizó con un máximo de 5.000 generaciones, una población de 300 individuos y otra de igual tamaño para los hijos. La población de la siguiente generación se obtiene al combinar los grupos de padres e hijos (reemplazo $m+1$) haciendo uso del método de la ruleta. EIP fue implementado considerando los ancestros en común de las dos últimas generaciones consecutivas. Se usa: selección por torneo binario para conformar el conjunto de padres a recombinar, crossover uniforme con una probabilidad de 0.65 y mutación Big-Creep con una probabilidad de 0.01. El número de crossover n_1 fue establecido entre 2 y 4, mientras que el de padres n_2 , entre 3 y 4. Se eligió el mejor de los vástagos para insertarlo en la población de hijos ($n_3=1$).

La implementación de los AEPs se realizó mediante un modelo isla, distribuyendo los 300 individuos en 3 islas con poblaciones de igual tamaño. En estas la frecuencia de migración se establece en 25 generaciones, migrando 5 individuos seleccionados, mediante un torneo de 3 individuos, a la próxima isla del anillo unidireccional estático. El control sobre la recepción de inmigrantes desde la isla previa se hace en cada generación. Las 5 soluciones a ser reemplazadas se seleccionan mediante un torneo de 5. Los individuos recibidos pasarán a formar parte de la población sólo en caso de ser mejores a los seleccionados para el reemplazo.

VII. RESULTADOS DE LOS EXPERIMENTOS

En esta sección se presenta un análisis comparativo del comportamiento de los AEs secuenciales y paralelos, en ambos casos se realiza control de incesto y se usa multiplicidad. El objetivo de la experimentación es identificar aquellas opciones algorítmicas que brinden soluciones de forma más eficiente y eficaz.

Las medidas de performance que se usan para poder realizar la evaluación de los experimentos son:

- PE (Porcentaje de Éxito). Indica el porcentaje de todas las ejecuciones donde una solución fue encontrada.
- NPE (Número Promedio de Evaluaciones). Indica el número promedio de evaluaciones necesarias para encontrar la solución óptima. Es válido aclarar que si PE=0, NPE no está definido.
- TpoEj (Tiempo de Ejecución). Indica el tiempo de ejecución, expresado en segundos, para encontrar el mejor individuo (no necesariamente el óptimo).

Las Tablas 2, 3 y 4 presentan los valores de las variables de performance obtenidos al ejecutar diferentes combinaciones algorítmicas en un ambiente secuencial.

Los valores remarcados en la Tabla 2 indican que los correspondientes algoritmos alcanzan en todas sus ejecuciones el valor óptimo para esa instancia. En la Tabla 3 se presentan los valores de NPE para cada una de las instancias. Al vincular esta tabla con la Tabla 2 se evidencia, en general, que las opciones que realizan un mayor número de evaluaciones son aquellas con los valores de PE más bajos. Al comparar las alternativas algorítmicas entre sí, no se encuentran diferencias significativas que permitan identificar cuál de ellas presenta un comportamiento que brinde los mejores resultados.

AE/Instancia	1	4	5	7
Simple	0.97	0.80	0.63	1.00
EIP	1.00	0.63	0.60	0.90
MCMP-2-3	0.97	0.63	0.73	0.93
MCMP-2-4	0.93	0.57	0.83	0.93
MCMP-3-3	0.97	0.50	0.70	0.90
MCMP-3-4	0.97	0.60	0.60	0.93
MCMP-4-3	1.00	0.53	0.63	1.00
MCMP-4-4	0.90	0.77	0.70	0.93
EIP-MCMP-2-3	0.90	0.57	0.73	1.00
EIP-MCMP-2-4	0.93	0.73	0.67	0.87
EIP-MCMP-3-3	0.93	0.57	0.73	0.93
EIP-MCMP-3-4	0.97	0.77	0.77	0.97
EIP-MCMP-4-3	1.00	0.67	0.77	1.00
EIP-MCMP-4-4	0.97	0.67	0.70	1.00

Tabla 2: Resultados de PE para AEs secuenciales en las distintas instancias

AE/Instancia	1	4	5	7
Simple	211069	999528	1944035	457671
EIP	43959	1897467	1925860	765471
MCMP-2-3	198701	1745193	1234167	722562
MCMP-2-4	358959	2131612	842231	1127068
MCMP-3-3	203114	2344854	1462941	1019213
MCMP-3-4	213159	1856473	1793938	679652
MCMP-4-3	49765	2176554	1739357	501975
MCMP-4-4	499765	1304135	1487067	874778
EIP-MCMP-2-3	517475	1965518	1470112	372781
EIP-MCMP-2-4	346736	1461954	1757909	1125123
EIP-MCMP-3-3	355330	2061180	1315080	706188
EIP-MCMP-3-4	240304	1173693	1136328	665281
EIP-MCMP-4-3	62104	1666428	1355928	498491
EIP-MCMP-4-4	219488	1876070	1484831	451865

Tabla 3: Resultados de NPE para AEs secuenciales en las distintas instancias

AE/Instancia	1	4	5	7
Simple	2.27	8.66	22.12	24.27
EIP	1.56	17.87	13.13	18.44
MCMP-2-3	2.33	15.49	4.46	29.17
MCMP-2-4	3.33	21.55	7.64	62.16
MCMP-3-3	2.54	12.25	9.44	39.26
MCMP-3-4	3.27	13.97	4.32	32.38
MCMP-4-3	2.14	16.28	8.73	33.50
MCMP-4-4	3.14	20.80	17.74	43.01
EIP-MCMP-2-3	3.82	6.52	18.83	26.18
EIP-MCMP-2-4	3.34	22.81	22.93	46.65
EIP-MCMP-3-3	3.07	18.65	10.23	29.68
EIP-MCMP-3-4	5.55	12.05	9.37	44.72
EIP-MCMP-4-3	2.87	20.38	20.48	35.14
EIP-MCMP-4-4	4.38	40.59	14.26	38.14

Tabla 4: Resultados de TpoEj en AEs secuenciales en las distintas instancias

En la Tabla 4 se indican los tiempos promedios, expresados en segundos, para hallar la mejor solución. Se puede observar un mayor costo computacional para alcanzar el óptimo cuando se combina prevención de incesto con multiplicidad. Este incremento surge dado que durante la evolución se produce un aumento en el número de individuos que mantienen una relación ancestral. En consecuencia, al avanzar en las generaciones, se torna complejo conseguir un conjunto factible de padres para la recombinación. En general, también se observa un incremento en el tiempo cuando el número de padres pasa de 3 a 4 para una determinada cantidad de hijos. En la mayoría de los casos, el hecho de crear 4 hijos hace que cualquiera de los algoritmos tarde más tiempo que con sólo la generación de 2 vástagos; mientras que la creación de 3 hijos puede insumir más tiempo que la de 4 y menos que la de 2.

Las Tablas 5, 6 y 7 presentan los valores de las variables de performance obtenidos al ejecutar los mismos algoritmos en un ambiente paralelo.

En la Tabla 5 se presentan los valores de PE para cada una de las instancias, remarcándose aquellos valores cuyas opciones algorítmicas alcanzaron el valor óptimo en el 100% de los casos. En la misma se observa un mayor número de aciertos para todas las opciones cuando se las compara con sus pares secuenciales. Para las instancias 1 y 7 se encuentra el valor óptimo en todas las ejecuciones, excepto el algoritmo EIP que lo obtiene en 29 de las 30 ejecuciones. Dentro de las instancias consideradas, la 4 y la 5 presentan mayor dificultad en su resolución a pesar de que sólo utilizan 40 variables; esto se evidencia al obtener solamente con algunas opciones algorítmicas el 100% de éxito. Para estas mismas instancias aumenta el porcentaje de éxito en un promedio del 30% con respecto a las opciones secuenciales. Comparando las opciones algorítmicas paralelas entre sí no se encuentran diferencias importantes que permitan identificar un algoritmo cuyo comportamiento brinde los mejores resultados.

Los valores de la Tabla 6 reflejan el objetivo inicial de disminuir el número de evaluaciones mediante algoritmos paralelos al contrastarlos con los secuenciales (Tabla 3). Esta diferencia es más notoria cuando se evalúan las instancias 4 y 5 en el ambiente paralelo, ya que se realizan alrededor de 1 millón menos de evaluaciones.

En la Tabla 7 se indican los tiempos promedios en los algoritmos paralelos para hallar la mejor solución. En dicha tabla se observa una reducción del costo computacional (expresado en segundos) en la mayoría de las opciones cuando son comparadas con sus pares secuenciales. Además, dentro de las alternativas ejecutadas en paralelo, es posible identificar a EIP-MCMP como la combinación que requiere menor tiempo para conseguir el óptimo. Esta es una ventaja en contraposición al

AE/Instancia	1	4	5	7
Simple	1.00	0.87	0.97	1.00
EIP	0.97	0.87	1.00	0.97
MCMP-2-3	1.00	1.00	0.93	1.00
MCMP-2-4	1.00	0.90	1.00	1.00
MCMP-3-3	1.00	0.93	0.97	1.00
MCMP-3-4	1.00	1.00	1.00	1.00
MCMP-4-3	1.00	0.93	0.93	1.00
MCMP-4-4	1.00	0.97	1.00	1.00
EIP-MCMP-2-3	1.00	0.93	1.00	1.00
EIP-MCMP-2-4	1.00	0.97	0.93	1.00
EIP-MCMP-3-3	1.00	0.97	1.00	1.00
EIP-MCMP-3-4	1.00	0.97	0.97	1.00
EIP-MCMP-4-3	1.00	0.87	1.00	1.00
EIP-MCMP-4-4	1.00	0.93	1.00	1.00

Tabla 5: Resultados de PE para AEs paralelos en las distintas instancias

AE/Instancia	1	4	5	7
Simple	605370	1257450	1038900	684040
EIP	494920	1265360	1059800	661300
MCMP-2-3	298871	1113070	1164850	760920
MCMP-2-4	468650	1262090	1193390	745250
MCMP-3-3	244360	1120940	1133750	549480
MCMP-3-4	220550	1396460	1113350	656850
MCMP-4-3	384530	1049060	973340	541690
MCMP-4-4	449000	1309710	1055670	622650
EIP-MCMP-2-3	350940	1116270	1184580	616670
EIP-MCMP-2-4	331070	1157300	949040	513520
EIP-MCMP-3-3	331580	1203090	1211350	554430
EIP-MCMP-3-4	256100	1311870	1088350	684050
EIP-MCMP-4-3	472060	1225450	1052080	539550
EIP-MCMP-4-4	411340	1282960	930060	468730

Tabla 6: Resultados de NPE para AEs paralelos en las distintas instancias

AE/Instancia	1	4	5	7
Simple	2.91	14.66	16.29	11.02
EIP	2.80	22.03	16.17	15.59
MCMP-2-3	1.06	13.11	3.08	3.69
MCMP-2-4	0.66	12.49	5.97	6.26
MCMP-3-3	0.89	13.56	11.59	12.02
MCMP-3-4	0.69	6.22	4.33	3.07
MCMP-4-3	0.78	15.29	19.99	7.98
MCMP-4-4	0.47	9.88	12.92	8.07
EIP-MCMP-2-3	0.65	2.32	4.39	4.16
EIP-MCMP-2-4	1.03	8.66	5.01	2.88
EIP-MCMP-3-3	0.51	3.88	8.97	6.09
EIP-MCMP-3-4	0.69	3.24	4.52	6.68
EIP-MCMP-4-3	0.48	4.17	1.58	8.88
EIP-MCMP-4-4	0.57	8.62	3.97	2.77

Tabla 7: Resultados de TpoEj en AEs paralelos

secuencial, ya que periódicamente las poblaciones intercambian individuos debilitando así, la relación ancestral.

Tanto las opciones ejecutadas en secuencial como en paralelo proveen el valor óptimo en todas las instancias, pero las últimas lo hacen en un mayor porcentaje de veces y en general en menor tiempo.

VIII. CONCLUSIONES

En este trabajo se examinaron los beneficios de la incorporación a un algoritmo evolutivo de las características de prevención de incesto extendida y de multiplicidad (tanto de padres como de hijos). Los algoritmos evolutivos se ejecutaron en versiones secuenciales y paralelas. Siendo el problema utilizado para la prueba fue el 3-SAT.

A partir del análisis de los resultados alcanzados se infiere que los algoritmos evolutivos paralelos que incorporan las características mencionadas previamente: obtienen mayor porcentaje de éxito, realizan un menor número de evaluaciones e insumen menos tiempo que los secuenciales. Por lo

tanto la paralelización del algoritmo evolutivo logra, en general, una mayor eficacia y eficiencia para resolver un problema.

En posibles trabajos futuros se aplicarán las mejores opciones algorítmicas paralelas, aquí detectadas, para resolver otros problemas de optimización compleja. Como así también, la variación del número de hijos y de padres a fin de obtener una combinación adecuada.

AGRADECIMIENTOS

Agradecemos la cooperación del grupo de proyecto, por proveer nuevas ideas y críticas constructivas. También a la Universidad Nacional de La Pampa de la que recibimos un aporte continuo.

REFERENCIAS

- [1] Alba Torres, E., Troya, J. M., "A Survey of Parallel Distributed Genetic Algorithms". Complexity. Volumen 4. Número 4. pp. 31-52. 1999.
- [2] Alba Torres, E., et. al., "MALLBA: A Library of Skeletons for Combinatorial Optimisation". Proceedings of Eighth International Euro-Par Conference. pp. 927-932. 2002.
- [3] Alba Torres, E., Cotta C., "Optimización en Entornos Geográficamente Distribuidos". Proyecto MALLBA. 2002.
- [4] Alfonso H., Cesan P., Fernandez N., Minetti G., Salto C., Velazco L., Gallard R., "Improving Evolutionary Algorithms Performance by Extending Incest Prevention". Proceedings del Cuarto Congreso Argentino de Ciencias de la Computación (CACIC'98). pp. 323-334. 1998.
- [5] Cook S.A., "The complexity of theorem-proving procedures". Proceedings of Third Annual ACM Symposium on the Theory of Computing. pp. 151-158. 1971.
- [6] Eiben, A.E., A Method for Designing Decision Support Systems for Operational Planning, PhD Thesis, Eindhoven University of Technology, 1991.
- [7] Eiben A.E., van Kemenade C.H.M., and Kok J.N., "Orgy in the computer: Multi-parent reproduction in genetic algorithms". Proceedings of the Third European Conference on Artificial Life, Numero 929. pp. 934-945. 1995.
- [8] Eiben, A.E., van der Hauw, J.K., "Solving 3-SAT with adaptive Genetic Algorithms". Proceedings of the Fourth IEEE Conference on Evolutionary Computation. pp. 81-86. 1997.
- [9]. Eiben A.E., Bäck Th. "An empirical investigation of multi-parent recombination operators in evolution strategies". Evolutionary Computation, Volumen 5. Número 3. pp. 347-365. 1997.
- [10] Eshelman L. J, Schaffer J.D., "Preventing Premature Convergence in Genetic Algorithms by Preventing Incest". Proceedings of the Fourth International Conference on Genetic Algorithms. pp. 115-122. 1991.
- [11] Esquivel S., Leiva A., Gallard R., "Multiple crossover per couple in genetic algorithms". Proceedings. of the Fourth IEEE International Conference on Evolutionary Computation. pp. 103-106. 1997.
- [12] Esquivel S., Leiva A., Gallard R., "Couple fitness Based Selection with Multiple Crossover per Couple in Genetic Algorithms". Proceedings of the International Symposium on Engineering of Intelligent Systems. Volumen 1. pp. 235-241. 1998.
- [13] Esquivel S., Leiva H.,Gallard R., "Multiplicity in genetic algorithms to face multicriteria optimization". Proceedings of the 1999 Congress on Evolutionary Computation (IEEE). pp. 85-90. 1999.
- [14] Esquivel S., Leiva H.,Gallard R., "Multiple crossovers between multiple parents to improve search in evolutionary algorithms". Proceedings of the 1999 Congress on Evolutionary Computation (IEEE). pp. 1589-1594. 1999.

- [15] Esquivel, S.C., Ferrero, S.W., Gallard, R.H., Salto, C., Alfonso, H, Schütz, M “Enhanced Evolutionary Algorithms for Single and Multiobjective Optimization in the Job Shop Scheduling Problem”. Knowledge-Based Systems. Volumen 15. pp. 13-25. 2002.
- [16] Garey M., Johnson D., *Computers and Intractability: a Guide to the Theory of NP-completeness*. Freeman. 1979.
- [17] Hao, J.K., “A Clausal Genetic Representation and its Evolutionary Procedures for Satisfiability Problems”. Artificial Neural Nets and Genetic Algorithms: Proceedings of the International Conference in Ales. pp. 289-292. 1995.
- [18] Manderick, B. Spiessens, R., “Fine-Grained Parallel Genetic Algorithms”. Proceedings of the Third International Conference on Genetic Algorithms. pp. 428-433. 1989.
- [19] Mitchell, D.G., Selman, B., Levesque, H.J., “Hard and Easy Distributions for (SAT) Problems”. Proceedings of the Tenth National Conference on Artificial Intelligence. AAAI Press. pp. 459-465. 1992.
- [20] Selman, B., Levesque, H. Mitchell, D., “A new method for solving hard satisfiability problems”. Proceeding Tenth National Conference on Artificial Intelligence. pp. 440-446. 1992.
- [21] Spiessens, P., Manderick, B., “A Massively Parallel Genetic Algorithm”. Proceedings of the Fourth International Conference on Genetic Algorithms. pp. 279-286. 1991.
- [22] Tanese, R., “Distributed Genetic Algorithms”. Proceeding of the Third International Conference on Genetic Algorithms, pp. 434-439. 1989.
- [23] Tseitin G., “On the complexity of derivation in propositional calculus”. Studies in Constructive Mathematics and Mathematical Logic. Volumen 2. pp. 115-125. 1968.
- [24] D. Whitley, T. Starkweather. *GENITOR II: a Distributed Genetic Algorithm*. Journal of Experimental and Theoretical Artificial Intelligence. Volumen 2. pp. 189-214. 1990.