

Capítulo V

ALGORITMOS EVOLUTIVOS AVANZADOS

5.1. INTRODUCCIÓN

En este capítulo se presentan algunas modificaciones al diseño de un algoritmo evolutivo para mejorar su performance. El hecho de innovar en alguna de las características de los algoritmos evolutivos, a fin de acrecentar su calidad y aplicabilidad, los transforman en algoritmos evolutivos avanzados. Estas modificaciones pueden afectar tanto a los operadores genéticos como al mecanismo de selección y tienen por objetivo balancear la explotación y la exploración a fin de evitar la convergencia prematura durante el proceso de búsqueda.

En los siguientes apartados se presentan tres técnicas que combinadas minimizan el riesgo de la convergencia prematura, a lo cual se le suma que la población final se encuentre centrada alrededor de la solución óptima al problema en cuestión. Las técnicas son:

- ✓ Múltiples padres.
- ✓ Múltiples crossovers.
- ✓ Prevención de incesto.

5.2. ALGORITMOS EVOLUTIVOS AVANZADOS CON MÚLTIPLES PADRES Y MÚLTIPLES CROSSOVERS

En la evolución simulada, es decir con algoritmos evolutivos, muchas características técnicas se inspiran por los mecanismos naturales. En particular, variantes abstractas de la reproducción sexual y asexual se implementan como operadores de búsqueda. Algunas técnicas evolutivas, por ejemplo programación evolutiva, trabajan exclusivamente con mutación (es decir, implementan una simplificación de la reproducción asexual), mientras otras, por ejemplo algoritmos genéticos y estrategias

evolutivas, usan recombinación (implementan una simplificación de la reproducción sexual) y mutación. Hay varios trabajos que investigan las ventajas y desventajas de la mutación con respecto al crossover [41, 60, 63, 86, 125,131].

Técnicamente, la cuestión se centra en la aridad de los operadores de reproducción. La mutación y el crossover tienen aridad uno y dos, respectivamente, y la pregunta es si los operadores unarios o binarios son preferibles para instancias típicas de problemas de optimización de relevancia práctica. Desde el punto de vista técnico, no hay necesidad de restringir la aridad de los operadores de reproducción en uno o dos. En general, un operador de reproducción puede tener una aridad desde uno al tamaño de la población (o aún más, si se permite la repetición entre los padres). Durante el desarrollo de la computación evolutiva se han propuesto varios operadores de recombinación, algunos alrededor de los sesenta [19]. En estrategias evolutivas, la recombinación global puede mezclar información proveniente de más de dos padres en un hijo [8, 126]. En algoritmos genéticos, la recombinación *p-sexual voting* de Mühlenbein [106], *scanning crossover* y *diagonal crossover* [37] y *recombinación gene-pool* [107, 141] son ejemplos de operadores *multi-parents*.

5.2.1. EVOLUCIÓN DE LA OPCIÓN MULTIPARENT

Un proceso de evolución mantiene una población de cromosomas a efectos de buscar a través del espacio de búsqueda de soluciones potenciales. Se debe lograr un balance entre [18]: la *explotación* de la mejor solución y la *exploración* del espacio de búsqueda.

Un factor a tener en cuenta, a la hora de mantener un buen balance, es el efecto causado por los mecanismos de selección: una fuerte presión selectiva puede producir una convergencia prematura a un óptimo local, mientras que una presión selectiva suave puede llevar a que la búsqueda se vuelva inefectiva.

El crossover contribuye en gran medida al intercambio de material genético durante la ejecución de un algoritmo evolutivo. Este operador combina las características de dos cromosomas para formar dos hijos similares al intercambiar los segmentos correspondientes de los padres. La idea detrás de la aplicación del operador de crossover es el intercambio de información entre diferentes soluciones potenciales. Muchos investigadores estudiaron el efecto de distintos tipos de crossover para mejorar la

performance de un GA en cuanto a exactitud y velocidad. Esas investigaciones abarcan varios tipos de crossover (crossover de un punto, crossover de dos puntos, crossover multi punto y crossover uniforme) y crossover especializados los cuales dependen de la estructura de datos particular usada para representar un cromosoma. Por ejemplo, Davis [25], Goldberg [74] y Oliver [111] crearon diversas variantes de crossover tal como OX, PMX y CX cuando trataban el problema del viajante (TSP) usando algoritmos genéticos. Se propusieron diversos operadores de crossover los cuales investigan el espacio de soluciones de diferentes maneras. Frantz [65], Syswerda [134] y Davis [27] propusieron el crossover *multi punto*, *uniforme* y *guaranted-uniform crossover*, respectivamente. Más recientemente Eshelman y Schaffer [39] estudiaron las características de algoritmos genéticos probando diversas variantes de crossover.

Independientemente del tipo de crossover aplicado, con la opción de crossover convencional se aplica el operador sólo una vez sobre la pareja de padres seleccionada para crear dos hijos. Por ejemplo, crossover de un punto, de dos puntos, uniforme y otras variaciones producen un par de hijos por pareja. Tal procedimiento se denomina como *single crossover per couple* (SCPC).

5.2.1.1.MULTIPLE CROSSOVER PER COUPLE

En [44] y [45] se presenta una opción diferente: *permitir múltiples hijos por pareja*, como frecuentemente ocurre en la naturaleza. La cantidad de hijos varía desde cero a algún número máximo predefinido. Este método de crossover se conoce como *multiple crossover per couple* (MCPC).

Esta opción se probó con un conjunto de funciones de testeo bien conocidas (funciones de De Jong F_1 , F_2 , F_3 [29], Schaffer F_1 [124] y otras funciones). El hecho de realizar múltiples crossover por pareja sobre un par de padres seleccionados provee beneficios extras en cuanto a tiempo de procesamiento y calidades similares en las soluciones halladas cuando se contrasta contra la opción convencional SCPC. Con la opción MCPC se observa que [49]:

- ✓ En algunos casos MCPC halla resultados que son mejores que los hallados por SCPC.
- ✓ MCMP halla una reducción del tiempo de procesamiento cuando se incrementa el número de crossover realizados por pareja.

- ✓ Se obtiene una mejora en la calidad de los resultados permitiendo entre 2 y 4 crossover por pareja.

MCPC se aplicó para optimizar funciones de testeo clásicas y algunas funciones duras (no lineales, no separables). Las bondades de esta opción prevalecen bajo todos los testeos y revelan que, cuando MCPC se aplica con 2, 3 y 4 crossovers por pareja, los resultados son tan buenos como se pueden esperar bajo SCPC, pero con un beneficio adicional en tiempo de procesamiento. La performance se obtiene a través de la habilidad mostrada por MCPC para explotar la recombinación de las soluciones buenas halladas.

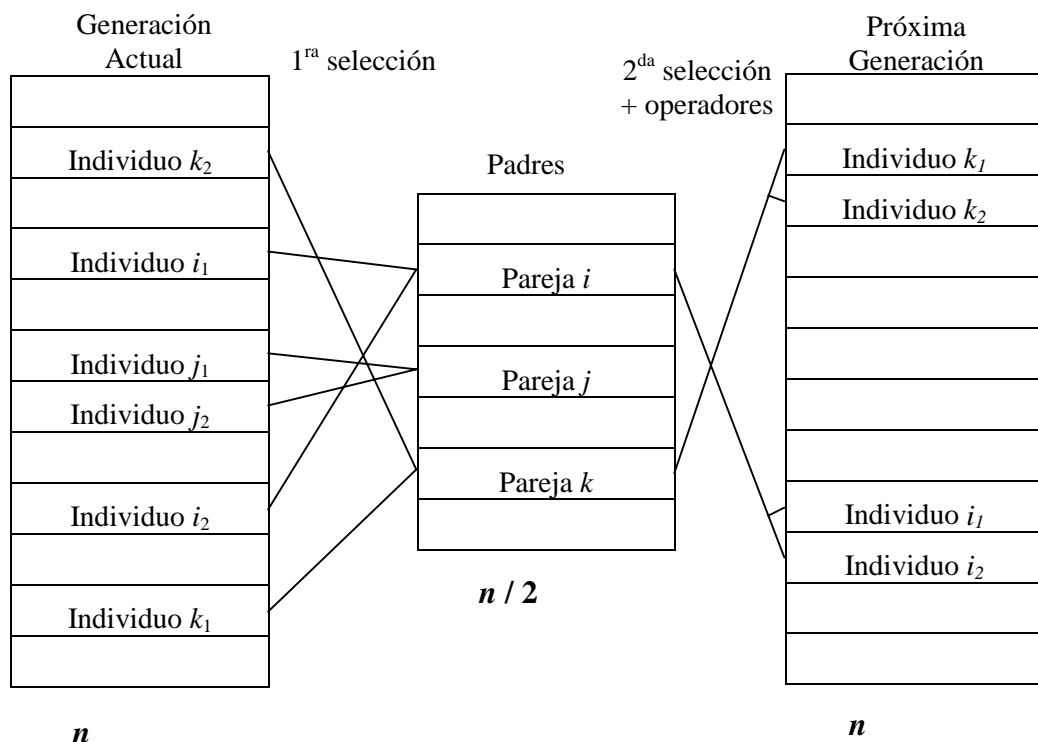
Los efectos de MCPC son: una economía en el esfuerzo computacional y una gran explotación en la recombinación de las soluciones buenas previamente encontradas. Pero por otra parte, esos experimentos mostraron que, en algunos casos, el método incrementa el riesgo de convergencia prematura dada la pérdida de diversidad genética. Para solucionar este problema se presenta una estrategia para combinar la explotación introducida por MCPC con la exploración de nuevas soluciones en el espacio del problema. Esto se logra al introducir un nuevo operador de selección [48]: *fitness proportional couple selection* (FPCS).

FPCS divide el proceso de selección total en dos etapas [51]. La primera selecciona los individuos de la población actual para crear un grupo intermedio de parejas, y posteriormente, la segunda etapa somete a esa población de parejas a la selección para cruzar aquellos pares con el fitness más alto (figura 5.1) [46]. Ambos pasos de selección se basan en el esquema de selección proporcional.

El operador de selección se puede modificar para favorecer, con más hijos, a aquellas parejas que tienen características específicas. No hay problema respecto del fitness individual, pero se han elegido dos criterios para asignar el fitness a las parejas: una basada sobre miembros con fitness disímiles ($FPCS_{DIFF}$) y el otro, sobre fitness promedio ($FPCS_{AVG}$).

El método se puede describir de la siguiente manera:

1. seleccionar individuos usando selección proporcional para construir la población intermedia de padres.
2. asignar a cada pareja un valor de fitness de pareja, calculado de acuerdo al criterio de fitness de la pareja.



1^{ra} Selección: selección proporcional al fitness del individuo
2^{da} Selección + operadores: selección proporcional al fitness de la pareja más operadores genéticos clásicos.
Fitness de la Pareja = $|\text{fitness.padre1} - \text{fitness.padre2}|$
 n : tamaño de la población.

Figura 5.1 Esquema del proceso de selección de parejas.

3. seleccionar las parejas para reproducción por medio de selección proporcional. Por cada pareja, se controla el proceso de producción de hijos para no sobrepasar el tamaño de la población.

Combinando las dos técnicas, MCPC y FPCS, se intenta disminuir la presión selectiva introducida por MCPC, buscando alta calidad de soluciones. Además se intenta mejorar la diversidad genética en forma indirecta al asignar un fitness alto a parejas disímiles esperando consecuentemente que los padres se encuentren a gran distancia en el *fitness landscape*.

Se obtienen buenos resultados para todas las funciones de optimización cuando se contrastan con los resultados obtenidos solamente bajo la opción MCPC [48]. Estos buenos resultados se logran permitiendo entre 2 y 3 crossover por pareja; pero determinar el valor adecuado de crossover por pareja es una tarea bastante tediosa.

Como una alternativa para hallar un número óptimo, se incorpora a cada individuo de la población el número de crossovers [47]. Así con el EA se realiza una auto-

adaptación de este parámetro dando la cantidad más apropiada de la repetición de crossovers.

La auto-adaptación es un nuevo campo en computación evolutiva la cual aconseja la actualización de parámetros del algoritmo al evolucionarlos como parte de la estructura del cromosoma. La auto-adaptación de los parámetros es una de las tres técnicas mencionadas en [36] para cambiar dinámicamente el valor de un parámetro mientras el algoritmo se está ejecutando. En [36] se dan las siguientes categorías de control de parámetros:

- ✓ *Control de parámetros determinístico*: es el caso en el cual los valores de los parámetros se modifican acorde con una regla determinística, sin retroalimentación alguna del proceso de búsqueda realizado por la estrategia.
- ✓ *Control de parámetros adaptables*: en este caso, se usa alguna información de retroalimentación del proceso de búsqueda para determinar la dirección y magnitud del cambio en los parámetros.
- ✓ *Control de parámetros auto-adaptables*: aquí los parámetros a ser adaptados se codifican dentro del cromosoma y se someten a los operadores genéticos. Los mejores individuos de la población tienen mejores chances de supervivencia y reproducción. Aquí es de esperar que los mejores valores de parámetros se propaguen intensivamente.

Como el número de crossovers a ser aplicado a una pareja en MCPC es uno de los parámetros del algoritmo, incluidos como parte del individuo, la opción pertenece a la última categoría mencionada.

Como en [47] se usa una representación binaria del cromosoma, el número de crossovers permitido por un individuo se codifica en un campo el cual ocupa los últimos $\log_2(max_cross + 1)$ bits de cada individuo para hallar el óptimo esperado. Esos últimos bits se denominan *ncross_field*. En algunos experimentos se permite un máximo de tres y en otros un máximo de siete crossovers por pareja. De modo que dos o tres bits extras son suficientes para tal propósito.

De esta forma se tienen dos espacios de búsqueda: uno correspondiente a la función objetivo, y otro asociado al número de crossovers a aplicar.

Cada individuo preserva información respecto del número de crossover originalmente aplicados a sus padres. De esta forma es de esperar que, basados en el

principio de *supervivencia por fitness*, las buenas soluciones lleven información del número de crossover aplicados a sus ancestros y que este número sea apropiado.

De acuerdo a Spears [132], se usa una técnica de auto-adaptación. Una vez que se selecciona la pareja, se chequea el correspondiente número de crossovers arrastrado por cada padre y:

- ✓ si son iguales, entonces se aplica el operador de crossover tantas veces como lo especificado por *ncross_field*,
- ✓ en caso contrario, se usa un número en el rango permitido en lugar del número *ncross_field*.

En la segunda situación, y siguiendo la opción de Spears, cuando el número decodificado de crossover es diferente, se viola la intención de preservar información porque los hijos no podrán tomar el número de crossovers por el cual fueron creados. Si el crossover y la mutación no alteran el *ncross_field* (y este evento tiene una baja probabilidad de ocurrencia) entonces el hijo retiene información de sus padres, pero no preserva información respecto de cómo fueron creados.

Para retener la información respecto de cuantos individuos fueron creados o cómo los padres fueron creados, en [47] se presentan dos opciones:

- ✓ *E1*: en cualquier situación, el intercambio de información de los padres a los hijos se hace en la forma tradicional aplicando el operador genético con sus correspondientes probabilidades. En el caso de que los valores del campo *ncross_field* no fueran iguales, esta opción, preservando la información de los padres, fuerza la diversidad poblacional en el espacio de búsqueda del parámetro, porque la mayoría de las veces un hijo hereda características desde un padre y el otro hijo hereda característica desde el otro padre.
- ✓ *E2*: si los valores especificados en *ncross_field* no fueran coincidentes, entonces un nuevo valor aleatorio para el número de crossover se inserta primero en el campo *ncross_field* del padre, y luego se realiza el crossover por el número de veces especificado por ese valor generado al azar. Al preservar la información, esta opción crea individuos más similares en el espacio de búsqueda del parámetro e incrementa la pérdida de diversidad genética.

La auto-adaptación se analiza desde los resultados obtenidos por dos funciones de testeo bastante duras: función Easom y la función 7 de Schwefel (función multimodal).

Para contrastar los resultados se usa un EA canónico con los mismos valores para los parámetros.

Las variables con las cuales se mide la performance son [47]:

- ✓ Cr_Avg que mide el número de crossover medio por pareja.
- ✓ *Calidad* de las soluciones encontradas.

En ambos casos, la *calidad* alcanza el valor 1 (al menos una solución fue óptima) luego de 80 a 152 generaciones respectivamente y los valores de $CrAvg$ varían entre 2.5 y 2.8 luego de unas pocas generaciones bajo E1 y entre 2.3 y 2.6 bajo E2.

Aquí el comportamiento del mecanismo de control de parámetro adaptable es sencillo: cuando la diversidad genética en el espacio de búsqueda del parámetro es baja entonces se permiten menos cantidad de crossovers y viceversa. Este comportamiento favorece el proceso evolutivo.

5.2.1.2. MULTIPLE CROSSOVERS PER MATING ACTION

Como una consecuencia de la investigación relacionada a problemas de optimización multicriterio [50], MCPC se extiende a *multiple crossovers per mating action* (MCPMA) para aplicar el método a un conjunto formado por padres. Esto sirve para determinar que el *multiparent approach* (propuesto por Eiben) en un estudio de optimalidad Pareto [38, 34, 33] mitiga la posible pérdida de diversidad generada por MCPC y no requiere ajustes extras. Consecuentemente, se puede balancear la explotación y la exploración del espacio del problema.

MCPMA provee un medio para explotar buenas características de más de dos padres seleccionados en función de su fitness al aplicar repetidamente el método de crossover seleccionado. Una vez seleccionados, los padres se someten al crossover n_1 veces (especificado como argumento), y genera n_1 hijos.

Eiben usa, en su opción de múltiple padres, operadores genéticos multi-parent [37]. Esos operadores usan dos o más padres para generar hijos. El mecanismo básico usado para definir operadores multi-parent es *gene-scanning*. Las técnicas de scanning generan un hijo desde n_2 padres con $n_2 \geq 2$. Sobre diferentes funciones de optimización, diferentes versiones de scanning muestran diferentes comportamientos. Usando un gran número de padres n_2 , la creación de hijos está basada sobre un mayor muestreo del espacio de búsqueda y consecuentemente se suministra una mayor diversidad. Esto

puede ayudar a evitar la convergencia prematura. Como ha reportado en su trabajo, es difícil determinar el número óptimo de padres, pero ha concluido que la mejor performance se obtiene cuando se trabaja entre 2 y 4 padres, e incrementado el número de padres deteriora la performance.

5.2.1.3. MULTIPLE CROSSOVERS ON MULTIPLE PARENTS (MCMP)

Multiple crossovers on multiple parents (MCMP) presentado en [49], permite la múltiple recombinación de múltiples padres bajo crossover uniforme, esperando que la explotación y la exploración del espacio del problema estén adecuadamente balanceadas.

MCMP provee un medio para explotar buenas características de más de dos padres seleccionados acorde a su fitness y aplicando repetidamente una de las variantes de SX: un número n_1 de crossovers se aplica sobre un número n_2 de padres seleccionados. De los n_1 hijos producidos se seleccionan n_3 , acorde a algún criterio, para ser insertados en la próxima generación.

Bajo esta opción la velocidad de convergencia, medida en el número de generaciones, se aumenta sin incrementar el riesgo de convergencia prematura. Consecuentemente, la calidad de los resultados es muy buena.

Adicionalmente, cuando se observa la población final, se detecta que todos los individuos están mucho más concentrados alrededor del óptimo. Esta propiedad se detecta en la optimización multimodal. Esto es un aspecto importante para aplicaciones que necesitan soluciones alternativas cercanas al óptimo.

Aunque no se puede ser concluyente, por medio de esta asociación, múltiples crossovers sobre múltiples padres, el espacio de búsqueda se explota eficientemente por la aplicación de múltiples crossovers y se explora eficientemente por el gran número de muestreos provistos por múltiples padres.

5.2.2. OPERADORES GENÉTICOS MULTI-PARENT

En esa sección se describen varios operadores multi-parents. Esos operadores usan dos o más padres para generar un hijo.

5.2.2.1. TÉCNICAS DE GENE-SCANNING

El mecanismo general para scanning [35] es asignar un marcador tanto a cada padre como al hijo. El marcador del hijo atraviesa todas sus posiciones de izquierda a derecha, una por vez. En cada paso, el marcador de los padres se actualiza de modo que cuando se debe elegir un valor para el gen marcado en el hijo, los marcadores de los padres muestren las posibles elecciones. El algoritmo correspondiente se muestra en la figura 5.2.

Las dos características más importantes de todos los procedimientos de gene scanning son:

- ✓ el mecanismo de actualización del marcador del padre, y
- ✓ la forma en la cual se elige un valor de los genes marcados. Si se varía la forma en la cual se eligen los valores a ser insertados en el hijo se tienen tres técnicas de scanning específicas: *uniform*, *ocurrence based* y *fitness based*.
- ✓

5.2.2.1.1 UNIFORM SCANNING

Uniform scanning (USX) [35] es una extensión natural del número de padres para el crossover uniforme. El crossover uniforme se define de la siguiente manera:

- ✓ recorrer los dos padres y los dos hijos de izquierda a derecha,
- ✓ por cada posición en el hijo 1, elegir en forma aleatoria si se hereda del padre 1 o del padre 2.

procedure gene-scanning

begin

Iniciar el marcador de los padres

for *marcador_hijo* = 1 **to** *l* **do**

Actualizar el marcador de los padres

child.allele[*marcador_hijo*] = elegir un valor desde los indicados por los marcadores de los padres

end for

end procedure

Figura 5.2 Procedimiento de gene-scanning

Padre 1	1	0	1	1	0	1	0	1	1	1
Padre 2	1	0	0	0	0	1	0	1	0	1
Padre 3	0	1	0	1	1	0	1	0	0	0
Padre 4	0	1	0	1	0	1	0	1	0	1
Hijo	1	0	0	1	0	1	0	1	0	1

Figura 5.3 OSX sobre un patrón de bits.

Para USX, se crea un único hijo. Cada alelo en el hijo se elige por un mecanismo aleatorio uniforme, cada padre tiene igual chance de ser elegido para aportar un valor. Esto significa que la posibilidad de heredar desde el padre i , $P(i)$, se define como:

$$P(i) = \frac{1}{\text{número_de_padres}}$$

y el número esperado de genes para heredar desde el padre como:

$$E(i) := P(i) \times \text{CHROMOSOME_LENGTH}$$

5.2.2.1.2 OCURRENCE BASED SCANNING

Occurrence based scanning (OSX) [35] está basado sobre la premisa de que los valores que se repiten con mayor ocurrencia en la mayoría de los padres (los cuales se seleccionan en base a su fitness), en una posición particular, es el mejor valor a elegir. La elección entre los valores marcados se hace por una función de mayoría. Si no hay mayoría entre los valores marcados, se elige el valor del primer padre. La figura 5.3 muestra el funcionamiento del OSX para una representación binaria.

5.2.2.1.3 FITNESS BASED SCANNING

Fitness based scanning (FSX) [35] elige el valor a ser heredado en forma proporcional al valor de fitness de los padres. Por ejemplo, para un problema de maximización donde el padre i tiene un fitness $f(i)$, la probabilidad $P(i)$ de elegir un valor desde esos padres es (selección por ruleta):

$$P(i) = \frac{f(i)}{\sum f(i)}$$

Como para OSX, esto significa que el número esperado de genes heredado del padre i es:

$$E(i) = P(i) \times CHROMOSOME_LENGHT$$

5.2.2.1.4 ADAPTANDO EL SCANNING A DIFERENTES TIPOS DE REPRESENTACIONES

Es posible definir procedimientos de scanning para diferentes tipos de representaciones al cambiar el mecanismo de actualización de los marcadores [35]. Para representaciones donde todas las posiciones son independientes (no hay epistasis), el mecanismo de actualización del marcador es simple, los marcadores de los padres se inician en la primera posición de cada uno de ellos, y a cada paso todos se incrementan en uno (así recorren los padres de izquierda a derecha). En problemas donde está presente la epistasis se necesitan mecanismos de actualización de los marcadores más sofisticados.

5.2.2.2 ADJACENCY BASED CROSSOVER

Adjacency based crossover (ABC) [35] es un caso especial de scanning, diseñado para representaciones basadas en permutaciones donde las posiciones relativas de los valores es importante, por ejemplo en TSP. La principal diferencia entre los procedimientos de scanning ya descritos y ABC es la forma en la cual se selecciona el primer valor y el mecanismo de actualización del marcador. El primer valor del gen del hijo se hereda siempre del primer valor del gen del primer padre. El mecanismo de actualización es el siguiente: por cada padre, se setea su marcador al primer sucesor del valor previamente seleccionado, el cual no está en el hijo.

Un crossover similar a ABC se propone en [146]. La principal diferencia se produce cuando todos los sucesores inmediatos a la ciudad se han heredado en el hijo.

Se han definido dos tipos de ABC: *occurrence based* (OB-ABC) y *fitness based* (FB-ABC). Ambos usan el mismo mecanismo de selección de los valores a ser heredados por el hijo como *occurrence based* (figura 5.4) y *fitness based scanning*, la diferencia es el mecanismo de actualización del marcador.

Padre 1	1 3 2 7 5 4 8 6	1 3 2 7 5 4 8 6	1 3 2 7 5 4 8 6	1 3 2 7 5 4 8 6
Padre 2	2 3 8 5 6 4 1 7	2 3 8 5 6 4 1 7	2 3 8 5 6 4 1 7	2 3 8 5 6 4 1 7
Padre 3	3 7 2 4 8 1 6 5	3 7 2 4 8 1 6 5	3 7 2 4 8 1 6 5	3 7 2 4 8 1 6 5
Padre 4	1 5 2 7 6 3 8 4	1 5 2 7 6 3 8 4	1 5 2 7 6 3 8 4	1 5 2 7 6 3 8 4
Hijo	1	1 3	1 3 8	1 3 8 6
Padre 1	1 3 2 7 5 4 8 6	1 3 2 7 5 4 8 6	1 3 2 7 5 4 8 6	1 3 2 7 5 4 8 6
Padre 2	2 3 8 5 6 4 1 7	2 3 8 5 6 4 1 7	2 3 8 5 6 4 1 7	2 3 8 5 6 4 1 7
Padre 3	3 7 2 4 8 1 6 5	3 7 2 4 8 1 6 5	3 7 2 4 8 1 6 5	3 7 2 4 8 1 6 5
Padre 4	1 5 2 7 6 3 8 4	1 5 2 7 6 3 8 4	1 5 2 7 6 3 8 4	1 5 2 7 6 3 8 4
Hijo	1 3 8 6 4	1 3 8 6 4 5	1 3 8 6 4 5 7	1 3 8 6 4 5 7 2

Figura 5.4 Ejemplo del crossover OB-ABC.

5.2.2.3 CROSSOVER DIAGONAL

El crossover tradicional crea dos hijos desde dos padres al empalmar los padres entre el único punto de crossover e intercambiando los substrings. La idea básica detrás del crossover diagonal es generalizar este mecanismo para un crossover n -ario de $(n-1)$ -point. El crossover diagonal selecciona $(n-1)$ puntos de crossover produciendo n segmentos del cromosoma y formando n hijos al tomar las piezas desde los padres a través de las diagonales. Por ejemplo, el primer hijo se compone tomando el primer *substring*₁ del *padre*₁, el segundo *substring*₂ del *padre*₂, etc. Mientras el segundo hijo deberá tener el primer *substring*₁ del *padre*₂, primer *substring*₂ del *padre*₃, etc. La figura 5.5 ejemplifica esta idea.

5.3. ALGORITMOS EVOLUTIVOS AVANZADOS CON PREVENCIÓN DE INCESTO

Se han sugerido varios mecanismos para combatir la convergencia prematura en algoritmos genéticos. Muchos de ellos, están sujetos a la objeción de que tienden a socavar el paralelismo implícito del algoritmo, la propiedad que distingue a los algoritmos genéticos de otros mecanismos de búsqueda.

Padre 1	1a	1b	1c
Padre 2	2a	2b	2c
Padre 3	3a	3b	3c
Hijo 1	1a	2b	3c
Hijo 2	2a	3b	1c
Hijo 3	3a	1b	2c

Figura 5.5 Crossover diagonal para tres padres.

La convergencia prematura –pérdida de diversidad poblacional antes de alcanzar el óptimo o al menos un valor satisfactorio- se reconoce como una falla seria de los algoritmos genéticos. En [40] se examinan tres estrategias para combatir la convergencia prematura:

- ✓ Una estrategia de mating, *prevención de incesto*,
- ✓ Una estrategia de crossover, *uniform crossover*, y
- ✓ Una estrategia de manejo de la población, *eliminación de duplicados*.

Todas esas estrategias, pero especialmente las dos primeras, están sujetas a la objeción de volver al operador de crossover más destructor de esquemas, debilitando el paralelismo implícito de los algoritmos genéticos.

La idea de prevención de incesto, fue inicialmente propuesta por Eshelman y Schaffer [40] y muestra sus beneficios para evitar la convergencia prematura. Este método evita el cruzamiento de padres que muestren similitudes, las cuales se determinan en función de la distancia de Hamming de los padres. La prevención de incesto fue extendida en [2] al mantener información de los ancestros dentro del cromosoma. Modifica el mecanismo de selección para reproducción al prevenir el cruzamiento de individuos pertenecientes a la misma familia, por un número predefinido de generaciones. Esta opción fue testeada sobre un conjunto de funciones multimodales.

5.3.1. OPCIÓN DE EHELMAN Y SCHAFFER

Los padres se pueden seleccionar de modo de mantener la diversidad de la población. Usualmente, las estrategias de mating se consideran en el contexto de especialización, donde el objetivo es prevenir que individuos disímiles se crucen [17,

147]. Todo lo demás sigue igual, los hijos producidos por crossover de diferentes padres tenderán a ser más diversos. Las *funciones de sharing* de Goldberg, al reducir el fitness de los individuos como una función de cuán similares son a los otros individuos en la población, se pueden pensar como una estrategia de mating indirecta [77]. Aunque Goldberg presente funciones de sharing en el contexto de especialización, tienen el efecto de reprimir más que promover el mating dentro de especies, al menos para la especie dominante. En lugar de evitar el mating de individuos diversos, los individuos similares pertenecientes a la especie más grande tienen menor probabilidad de ser seleccionados para reproducción con los demás. El mecanismo de prevención de incesto de Eshelman es una opción más directa para prevenir el mating de individuos similares [42]. Los individuos se recombinan aleatoriamente, pero sólo se cruzan si su distancia Hamming es superior a un cierto umbral. De este modo sólo una parte de la población se cruza para producir nuevos hijos en cualquier generación. El umbral se establece inicialmente a una distancia Hamming promedio esperada de la población inicial, y entonces se disminuye a medida que la población converge. Aunque este mecanismo no previene explícitamente que se crucen hermanos o ancestros cercanos, este mecanismo tiene su efecto en la medida en que tales individuos tiendan a ser similares.

5.3.2. PREVENCIÓN DE INCESTO EXTENDIDA

La extensión del concepto de incesto está fuertemente relacionada al cruzamiento de miembros de la misma familia. Para prevenir el incesto, la *prevención de incesto extendida* (EIP) [2] permite sólo la recombinación de individuos que no compartan ancestros. Para construir la nueva población, los individuos se eligen de la previa en función a la selección proporcional al fitness, pero se permite su cruzamiento si no se detectan ancestros comunes en las generaciones cercanas. De esta forma, el intercambio de material genético se reduce y la diversidad de la población se mantiene en algún grado conveniente. Por lo tanto cada individuo mantiene información respecto de sus ancestros.

Permitiendo el crossover sólo entre individuos que no son familiares, se modifica el efecto del mecanismo de selección en la población. Sin embargo, la selección es el único operador de un EA donde el fitness del individuo afecta el proceso de evolución. En tal proceso hay dos puntos fuertemente relacionados: diversidad poblacional y

presión selectiva forzadas por el mecanismo. La selección juega un rol importante porque una fuerte presión selectiva puede producir convergencia prematura y una presión selectiva suave puede provocar que la búsqueda se torne inefectiva [84].

El pseudo-código que se muestra en la figura 5.6 delinea un procedimiento para prevenir el incesto entre miembros de una misma generación o consecutivas, es decir entre hermanos y entre padres e hijos.

En lugar de medir la similitud entre individuos a través de su distancia Hamming para prevenir la recombinación, EIP evita el cruzamiento entre individuos pertenecientes a la misma familia en un número limitado de generaciones.

Esta opción muestra una buena performance cuando se la contrasta con un EA tradicional para la optimización de funciones multimodales de variada dificultad. Los valores óptimos se alcanzan en la mayoría de las corridas de cada una de las series.

Esta variante presenta también un criterio alternativo para prevenir el incesto que es independiente de la representación del individuo y consecuentemente, se puede aplicar a representaciones con enteros, reales, vectores y otras sin necesidad de realizar cambios. Esta capacidad no está presente en el enfoque de Eshelman.

```

procedure selección de padres
begin
  for 1 to pop_size
    seleccionar indiv-1  $C(t)$ 
    seleccionar indiv-2  $C(t)$ 
    while ((padre(indiv-1) = padre(indiv-2)) or
      (indiv-1 = padre(indiv-2)) or
      (indiv-2 = padre(indiv-1)))
      seleccionar indiv-2  $C(t)$ 
    end while
    recombinar y mutar individuos en  $C(t)$  construyendo  $C'(t)$ 
  end for
end procedure

```

Figura 5.6 Procedimiento para EIP

En [2] se reportan resultados para dos variantes de EIP, se realiza la prevención de incesto durante dos (EIP2) y tres generaciones (EIP3) sucesivas. Cuando se usa la opción EIP3, se presentan algunas dificultades y se deben a que a medida que la población evoluciona se vuelve más difícil encontrar individuos para formar una pareja que no pertenezcan a la misma familia, para salvar esta dificultad es necesario considerar un tamaño de la población mayor al tomado para EIP2.

Se realizaron experimentos para un gran número de generaciones consecutivas, pero el proceso muestra un alto overhead cuando se buscan individuos que posean ancestros distintos cuando se realiza la búsqueda localizada. La prevención de incesto manteniendo la información ancestral por tres generaciones consecutivas presenta mejores resultados que cuando se aplica a sólo dos generaciones sucesivas.

EIP ha sido puesta a prueba sobre un amplio conjunto de funciones de testeo y aunque adiciona algún overhead dado el control, específicamente, cuando la diversidad genética de la población inicial no es alta, los resultados son prometedores.

5.4. EA AVANZADOS CON MULTIPLICIDAD Y PREVENCIÓN DE INCESTO

En [104] y [105] se presenta un algoritmo evolutivo el cual combina múltiples crossovers, múltiples padres y prevención de incesto. Esos trabajos mostraron que se mejora considerablemente el proceso de búsqueda de un algoritmo evolutivo.

Cuando se usa una cantidad de padres $n_2 > 2$, el pseudo código para la opción EIP se muestra en la figura 5.7.

Como se puede observar, cuando se aplica MCMP y mutación bit flip se obtienen n_1 hijos, una selección siguiente elige el hijo con el fitness más alto para insertarlo en la próxima generación.

Esta opción se ha analizado para un conjunto de funciones de testeo. Los resultados indican que esta opción mitiga la posible pérdidas de diversidad generada por la aplicación de múltiples crossovers sobre un par de padres. Consecuentemente la calidad de los resultados es al menos tan buena como las opciones previas más complejas. Adicionalmente, cuando se observa la población final se detecta que todos los individuos están mucho más centrados alrededor del óptimo. Es una característica importante cuando la aplicación necesita de múltiples soluciones alternativas cercanas al

```

procedure múltiple selección de padres
begin
  int mating_pool[cant_parents] //vector para almacenar los padres seleccionados
  int children_pool[cant_cross] //vector para almacenar los hijos creados
  for 1 to pop_size
    seleccionar indiv-1  $C(t)$ 
    mating_pool[1] = indiv-1
    i=2
    while (i <= cant_parent)
      repeat
        seleccionar indiv-i  $C(t)$ 
      until(is_relative(mating_pool, indiv-i)) //control de no existencia de
        ancestros comunes y unicidad de padres en el mating pool
      matting_pool[i] = indiv-i
      i=i+1
    end while
    recombinar usando MCMP y mutar individuos del mating_pool a
      children_pool
    seleccionar los mejores individuos de children_pool construyendo  $C'(t)$ 
  end for
end procedure

```

Figura 5.7 Modificación del procedimiento EIP para incorporar MCMP

óptimo para hacer frente a sistemas dinámicos. También la velocidad de convergencia, medida en número de generaciones, se aumenta sin incrementar el riesgo de convergencia prematura.