CAPÍTULO 3: ALGORITMOS GENÉTICOS

3.1. INTRODUCCIÓN

Un algoritmo genético (AG) es una forma de evolución [133] que ocurre en una computadora. Un algoritmo es una descripción general de un procedimiento, y un programa es su realización como una secuencia de instrucciones a una computadora. Los Algoritmos Genéticos son un método de búsqueda que puede usarse para: resolver problemas y modelar sistemas evolutivos. La idea básica de un algoritmo genético es muy simple. Primero se crea una población de individuos en una computadora (típicamente almacenada como un string binario en memoria), y entonces la población evoluciona usando los principios de variación, selección y herencia.

Los algoritmos genéticos, a diferencia de otras técnicas de búsqueda, realizan una búsqueda multidireccional al mantener una población de múltiples soluciones potenciales. Intentando así, escapar de los óptimos locales. La población sufre una evolución simulada: en cada generación las soluciones relativamente buenas se reproducen, mientras que las soluciones relativamente malas mueren. Los AGs usan reglas de transición probabilísticas para seleccionar individuos a reproducir o a morir.

Con distintas técnicas de transferencia y medidas apropiadas de fitness los Algoritmos Genéticos pueden hacerse a la medida de muchos tipos de problemas, incluyendo optimización de una función o determinación del orden apropiado de una secuencia [81, 130]. Los Algoritmos Genéticos son capaces de explorar un rango mucho más amplio de soluciones potenciales que los convencionales.

Los AGs han dado resultados satisfactorios en la resolución de problemas de optimización como ruteo de redes, scheduling, control adaptativo, game playing, modelado cognitivo, problemas de transporte, problemas del viajente, problemas de control óptimo, optimización de consultas de base de datos [7, 13, 14, 31, 42, 55, 59, 69, 73, 75, 76, 95, 105, 122, 143, 144].

3.2. DESCRIPCIÓN GENERAL DE LOS ALGORITMOS GENÉTICOS (AGS)

Goldberg define a los Algoritmos Genéticos en [69]. Los Algoritmos Genéticos son técnicas estocásticas de búsqueda ciega de soluciones cuasi-óptimas. Ellos mantienen una población que representa a un conjunto de posibles soluciones, llamadas individuos. Tal población es sometida a ciertas transformaciones con las que se trata de obtener nuevos candidatos, y a un proceso de selección sesgado a favor de los mejores individuos.

La principal innovación de los AGs en el dominio de los métodos de búsqueda es la adición de un mecanismo de *selección* de soluciones [113]. La selección presenta dos características: a corto plazo los mejores individuos tienen más probabilidad de sobrevivir, mientras que a largo plazo estos poseen más posibilidades de tener descendencia. A causa de lo anterior, el mecanismo de selección se desdobla en dos operadores:

1. El operador de *selección* elige los elementos a transformarse posteriormente (selección para la reproducción),
2. El operador *reemplazo* elige los elementos que sobreviven (selección para el reemplazo).

En cuanto a las restantes características, cabe destacar que los AGs canónicos o simples son métodos de búsqueda:

- *Ciegos*. Es decir, no disponen de ningún conocimiento específico del problema; de manera que la búsqueda se basa exclusivamente en los valores de la función objetivo.
- *Codificados*. Puesto que no trabajan directamente sobre el dominio del problema, sino sobre representaciones de sus elementos.
- *Múltiples*. Esto es, buscan simultáneamente entre un conjunto de candidatos.
- *Estocásticos*. Referido tanto a las fases de selección como a las de transformación. Ello proporciona control sobre el factor de penetración de la búsqueda.

Todas las características enumeradas se introducen deliberadamente para proporcionar más robustez a la búsqueda, esto es, para darle más eficiencia sin perder la generalidad y viceversa.

Como se menciona en párrafos anteriores los AGs, se diferencian, fundamentalmente, de las técnicas de búsqueda convencionales, porque, comienzan con un conjunto de soluciones llamado *población inicial*, entre otras cosas [63]. Cada *individuo* en la población es llamado *cromosoma*, y representa una solución a un problema dado. Un cromosoma es un conjunto de genes. Un gen puede tomar valores determinados pertenecientes a un conjunto de datos posibles. Dichos valores son llamados *alelos*. En otras palabras un cromosoma es un string de símbolos; que generalmente, pero no necesariamente, es un string de bits. Los cromosomas *evolucionan* a través de sucesivas iteraciones, llamadas *generaciones*. Durante cada generación, los cromosomas son *evaluados*, usando una medida de fitness (o aptitud) [56]. Para crear la próxima población nuevos cromosomas, llamados *hijos*, se forman por medio de:

1. La combinación de dos ó más cromosomas mediante un operador de crossover. Tales cromosomas padres son elegidos en general por su fitness, desde la población actual.
2. La modificación de un cromosoma usando el operador de *mutación*.

Una nueva población es obtenida al:

1. Seleccionar, en general de acuerdo al valor de fitness, algunos de los padres e hijos; y
2. Eliminar a otros para mantener constante el tamaño de la población.

Los cromosomas con mejor fitness tienen más probabilidad de ser seleccionados. Después de varias generaciones, los algoritmos convergen al mejor cromosoma, el cual puede representar la solución óptima o subóptima al problema. La estructura general de los AGs se describe en la figura 3.1. Durante la iteración *t* [107], un AG mantiene una población de soluciones potenciales (cromosomas, vectores), $P(t) = \{x^t_1, \ldots\ldots\ldots\ldots, x^t_n\}$. Cada solución x^t_i es evaluada para obtener una medida de su fitness. Entonces, una nueva población (iteración $t+1$) es producida al seleccionar los individuos de mejor fitness. Algunos miembros de esta nueva población sufren alteraciones por medio del crossover y la mutación, y

así forman nuevas soluciones. El crossover combina las características de dos cromosomas padres para formar dos vástagos similares al intercambiar los segmentos correspondientes de los padres. La idea que trae aparejada la aplicación del crossover es el intercambio de información entre distintas soluciones potenciales. Este es el principal operador genético en los AGs.

```
procedure Genetic Algorithm
begin
        t ← 0;
        inicializar P(t);
        evaluar estructuras en P(t);
        while la condición de terminación no se satisfaga do
        begin
          t ← t + 1;
          select_repro C(t) desde P(t-1);
          recombinar y mutar estructuras en C(t) formando C'(t);
          evaluar estructuras en C'(t);
          select_replace P(t) desde C'(t) y P(t-1);
        end
end
```

Figura 3.1. Estructura General de los Algoritmos Genéticos

La operación de mutación altera en forma arbitraria uno o más genes de un cromosoma seleccionado, al realizar un cambio aleatorio con una probabilidad establecida para la mutación. La mutación introduce una leve variabilidad en la población.

Un algoritmo genético para un problema en particular posee los siguientes cinco componentes:

1. Una representación genética para soluciones potenciales al problema.
2. Una manera de crear una población inicial de soluciones potenciales.
3. Una función de evaluación que cumple el papel del medio ambiente, ordenando las soluciones en términos de sus fitness,
4. Operadores genéticos que alteran la composición de los hijos.
5. Valores para los distintos parámetros que usa un algoritmo genético, como por ejemplo: tamaño de la población, probabilidades de aplicar los operadores genéticos, etc.

3.2.1. EL GENOTIPO

En genética, el *genotipo* es la colección abstracta de genes poseídos por un individuo [119, 136]. La estructura que contiene a los genes es el *cromosoma*. Un gen tiene un valor (*alelo*) y una posición en el genotipo (*locus*). En la naturaleza existe una compleja traducción entre el genotipo y las propiedades del organismo (el *fenotipo*). Se ha observado que un único gen puede afectar múltiples cualidades en el fenotipo, y esto se denomina *pleiotropía*. La otra traducción que se observa es la siguiente: múltiples genes controlan o afectan a una sola cualidad o característica del fenotipo (*poligenia*). Los sistemas evolutivos naturales hacen un uso extensivo de la poligenia y la pleiotropía [55]. Típicamente los AGs se abstraen mucho de la riqueza de la evolución natural y no explotan las codificaciones y mecanismos de crossover complejos que se encuentran en la naturaleza [55]. Muchos ejemplos pueden encontrarse en la literatura de los AGs, donde los genes tienen una traducción uno a uno a las propiedades fenotípicas. Este tipo de traducción representa una simplificación de la evolución natural.

Por ejemplo, una codificación binaria es una excelente opción para problemas en los cuales un punto del problema se traslada naturalmente a un string de ceros y unos. Un string binario se usa, a menudo, para representar tal codificación, ej. $E = [10001010]$, donde la codificación E tiene 8 genes. La posición (o locus) del i-*ésimo* gen es simplemente el bit i en el *bitstring*, y el valor o alelo está dado por el valor del bit i $E[i]$.

El problema de satisfacción booleana (SAT problem, en inglés) [32] se traduce muy bien a un string binario, ya que una solución a un problema SAT es una asignación de variables binarias. Sin embargo, para muchos problemas una codificación binaria no es apropiada porque:

1. Un alto grado de *epistasis* significa que buenos segmentos (bloques) no se pueden formar, entonces el problema es *deceptivo*. La epistasis mide la importancia en la cual la contribución de fitness de un gen depende de los valores de otros genes.

2. *La representación natural* del problema requiere un conjunto de símbolos de mayor orden que el binario. En estos casos un alfabeto de orden mayor puede ser más eficiente.
3. *Las soluciones ilegales* con codificación binaria pueden ser introducidas por los operadores genéticos. Ya que tal codificación puede no describir naturalmente un punto del problema.

3.2.2. EL FENOTIPO

En genética, el fenotipo es un rasgo distintivo o una característica medible que posee un organismo [119]. En AGs, el fenotipo indica las características definitorias o las cualidades del genotipo entero. En algunas situaciones, los rasgos fenotípicos surgen directamente desde el genotipo. Por ejemplo, si el genotipo codifica a un único parámetro binario de un problema, entonces el punto de dicho problema es descripto fenotípicamente al decodificar este parámetro binario en una cantidad numérica. Esta traducción directa entre el genotipo y el fenotipo es evidente en el problema del viajante de comercio (TSP en inglés) con una codificación de permutación [117, 118]. Sin embargo en otros tipos de problemas las características fenotípicas no surgen de forma tan sencilla; es el caso de una representación de *job shop scheduling* [23].

3.2.3. EXPLOTACIÓN VERSUS EXPLORACIÓN

La búsqueda es uno de los métodos más comunes para resolver problemas, en los cuales no se puede determinar a priori una secuencia de pasos para llegar a una solución. Dos puntos importantes son abordados por las estrategias de búsquedas, ciegas o heurísticas, ellos son: la *explotación* de la mejor solución y la *exploración* del espacio de búsqueda [15]. Michalewicz [107] hace una comparación entre las técnicas de búsqueda *hill-climbing, simulated annealing* y los algoritmos genéticos. Hill-climbing es una estrategia que explota la mejor solución para posibles mejoras pero ignora completamente la exploración del espacio de búsqueda. En tanto que simulated annealing [1] permite la exploración

del espacio de búsqueda a altas temperaturas, donde cualquier movimiento es permitido, y la explotación de la mejor solución cuando la temperatura es baja. Mientras que los algoritmos genéticos, son una clase de métodos de búsqueda de propósito general que combinan elementos de la búsqueda estocástica y dirigida; las cuales pueden remarcar el balance entre la exploración y la explotación del espacio de búsqueda.

En los AGs, los operadores de crossover, también llamados de cruce, se encargan de explotar las mejores características que disponga la población actual. Mientras que los operadores de mutación se encargan de explorar los nuevos dominios en busca de mejores soluciones [113]. Desde esta perspectiva, los operadores de cruce son los que principalmente se encargan de la búsqueda en profundidad (explotación) y los de mutación de la búsqueda en amplitud (exploración). Así dando mayor importancia a unos que a otros es posible ajustar el tipo de búsqueda a las necesidades del problema.

El crossover se encarga de realizar un intercambio estructurado de información, mientras que la mutación proporciona una garantía de accesibilidad para todos los puntos del espacio de búsqueda.

3.2.4. LIMITACIÓN EN LOS AGS

El aspecto más visible de la debilidad es la *ineficiencia*: los métodos débiles de resolución de problemas presentan las ventajas de ser muy poco específicos, poco exigentes y notablemente eficaces [113]. Sin embargo, todos ellos son ineficientes en mayor o menor grado, y especialmente si se los compara con otros métodos heurísticos. Los AGs sólo se pueden considerar eficientes en comparación con otros métodos estocásticos de búsqueda ciega, pero se puede garantizar que si se encuentra un método heurístico para resolver un problema específico este siempre lo hará mucho más eficientemente que un AG.

Pero eso no es todo, otro de los inconvenientes prácticos de la debilidad es la tendencia al *extravío* en la búsqueda. Este fenómeno es el siguiente: el AG simple realiza la búsqueda de los mejores puntos utilizando únicamente la aptitud (o fitness) de los individuos para luego recombinarlos. En ocasiones ocurre que la

información proporcionada (ej. el fitness) resulta insuficiente para orientar correctamente al algoritmo en su búsqueda del óptimo. A ésto se lo llama *desorientación* (deception en inglés). Esto se concreta porque ciertas combinaciones válidas de buenos segmentos originan individuos de baja aptitud. En el peor de los casos puede ocurrir que este fenómeno sea tan fuerte como para impedir que el AG converja al óptimo global, desviándolo finalmente hacia un óptimo local. Esto es, el extravío propiamente dicho. Afortunadamente, aunque puede que un AG se desoriente no necesariamente esto ocurre; para eso es necesario partir de una mala población inicial o plantear un problema especialmente diseñado para que se extravíe.

En resumen, para que funcione correctamente el mecanismo básico de los AGs es necesario proporcionarle una mínima cantidad (y calidad) de información. Si no se proporciona ese mínimo el mecanismo no funciona con propiedad, y el AG evolucionará incorrectamente.

Como es de suponer, el hecho que más contribuye a la existencia de desorientación es la forma de la función de fitness. Típicamente, la desorientación es frecuente en problemas donde los puntos óptimos están asilados y rodeados de puntos de muy baja aptitud. Sin embargo, conviene señalar que la presencia de desorientación suele estar fuertemente estimulada por una mala codificación que 'oculte' la ya de por sí escasa información disponible. De modo recíproco una buena codificación reduce la probabilidad de extravío.

Un caso especialmente desfavorable ocurre cuando hay una fuerte interacción entre dos o más atributos (ej. genes), de tal forma que la contribución al fitness de un individuo que realiza cierto gen depende, en gran medida, de los valores que tomen otros. A este fenómeno se lo denomina *epistasis* o acoplamiento y su presencia garantiza la desorientación dado que en esas condiciones resulta muy difícil constituir buenos segmentos.

3.2.5. EL PROBLEMA DE LA DIVERSIDAD EN LOS AGS

A efectos prácticos, es fundamental para el buen funcionamiento de un AG tener controlada en todo momento la diversidad de la población. Se entiende a la

diversidad en sentido general como 'diversidad de individuos' y en particular como 'diversidad de aptitudes'.

Con poca diversidad de individuos hay poca variedad de segmentos, a causa de ello el operador de crossover pierde casi por completo la capacidad de intercambio de información útil entre individuos, y en definitiva, la búsqueda se estanca. La necesidad de tener controlada la diversidad de aptitudes radica en la imposibilidad práctica de trabajar con una población infinita.

No es conveniente tener poca diversidad de aptitudes ya que en tal caso todos los individuos tendrían más o menos las mismas posibilidades de sobrevivir, la selección reproduciría la situación anterior y todo el peso de la búsqueda recaería en los operadores genéticos, lo que a la larga sería poco más que una búsqueda aleatoria. Como resultado final la búsqueda se estanca y, la situación puede empeorar si la población es finita. Resumiendo, para que la selección sea efectiva, la población debe contener en todo momento una cierta variedad de aptitudes.

Por otra parte cuando la población es finita –es decir, siempre- tampoco se puede tener gran disparidad de aptitudes, pues ello suele afectar muy negativamente a la diversidad de la población.

3.2.5.1. CONVERGENCIA PREMATURA POR PROBLEMAS CON LA DIVERSIDAD

Sea un AG básico con una población finita y admítase por simplicidad que la función de fitness coincide con la función de evaluación. En algún momento puede ocurrir que un individuo o un grupo de ellos obtengan un fitness notablemente superior a los demás. Esto es, especialmente probable en las fases tempranas de la evolución; donde suele surgir un buen candidato desde una población de individuos mediocres, por aplicación de los operadores genéticos. En tal circunstancia existe el riesgo de que se produzca una *evolución en avalancha*; debido al incremento de la presencia en la población de individuos más aptos, que por ser finita disminuye su diversidad. Ello hace que en la siguiente generación se favorezca aún más a los individuos más aptos hasta que éstos acaban dominando por completo la población. A esto se le conoce como el

fenómeno de los *superindividuos*. Habitualmente ocurre que tales superindividuos sólo son los más aptos en cierto momento, pero no los potencialmente más aptos -es necesario tener en cuenta que la falta de diversidad estanca la búsqueda-; provocando, en general, una *convergencia prematura* del AG, habitualmente hacia un subóptimo.

La única circunstancia en que este fenómeno es tolerable e incluso deseable, es en las fases tardías de la evolución; cuando el algoritmo genético ha 'localizado' correctamente la zona del espacio de búsqueda donde la solución óptima se encuentra, pero nunca antes.

La posibilidad de convergencia en avalancha es un fenómeno inherente a los AGs con población finita, debido a que la finitud activa el bucle "selección sesgada versus incremento de la diversidad". Tanto es así, que dicho fenómeno puede ocurrir incluso cuando no haya diferencias apreciables en las aptitudes de la población. Ya se ha comentado que en tal circunstancia la búsqueda se paraliza, dado que la selección no concede mayor preeminencia a uno u otro individuo. Sin embargo, cuando la población es finita y especialmente cuando no es muy grande, puede ocurrir el siguiente fenómeno conocido como *deriva genética*: en una población finita los procesos de muestreo son siempre imperfectos, pudiendo favorecer más de lo que les corresponde a los individuos ocasionalmente más aptos; a esto se lo denomina *presión selectiva*. Debido a la finitud, especialmente, en poblaciones de tamaño pequeño y como consecuencia de esos errores estocásticos en los muestreos, se puede producir la pérdida de material genético irrecuperable; haciendo que el AG converja prematuramente hacia un "individuo afortunado".

Es imprescindible tener el control sobre la diversidad de aptitudes de la población (finita) para evitar que se produzca una convergencia prematura (avalancha) ya sea por la presencia de superindividuos o incluso por deriva genética.

Mecanismos para prevenir la convergencia prematura son desarrollados en [40] y en [2].

3.3. REPRESENTACIÓN GENÉTICA (CROMOSOMAS)

Uno de los puntos clave de los AGs es como codificar la solución de un problema en un cromosoma. El trabajo de Holland ha sido llevado a cabo usando strings binarios (AG simple). Para muchas aplicaciones de AGs, especialmente para los problemas de la ingeniería industrial, el AG simple es difícil de aplicar porque el string binario no es una codificación natural. Han sido varias las codificaciones creadas con un formato diferente al del string, algunas de ellas son: codificación entera creada para problemas de optimización con restricciones, codificación en números reales para problemas de optimización de funciones reales. Elegir una representación apropiada de soluciones candidatas al problema en cuestión es fundamental para aplicar AGs. En cualquier aplicación es necesario realizar un análisis cuidadoso, para asegurar una apropiada representación de soluciones junto con operadores genéticos específicos del problema.

Una de las características básicas de los AGs, es que trabajan sobre un espacio codificado y un espacio de soluciones en forma alternativa. Esto es, las operaciones genéticas trabajan sobre un espacio codificado (cromosomas, genotipo), mientras que las operaciones de evaluación y selección lo hacen sobre el espacio de soluciones (fenotipo). Para las representaciones que no usan strings, surgen puntos críticos con la codificación y decodificación entre cromosomas y soluciones (o la traducción entre fenotipo y genotipo):

- *La factibilidad de un cromosoma*.
- *La legalidad de un cromosoma*.
- *La unicidad de la traducción.*

3.3.1. FACTIBILIDAD DE UN CROMOSOMA

La factibilidad de un cromosoma, se refiere al fenómeno de que una solución decodificada a partir de un cromosoma se encuentre en la región factible de un problema dado. La no factibilidad de un cromosoma proviene de la naturaleza de los problemas de optimización con restricciones. Todos los métodos, ya sean los convencionales o los evolutivos, deben manejar restricciones. En muchos

problemas de optimización, las regiones factibles pueden ser representadas como un sistema de ecuaciones o inecuaciones (lineales o no). Para tales casos, muchos métodos eficientes de penalidad se han propuesto para manejar cromosomas no factibles en los AGs [62, 106, 131]. En los problemas de optimización con restricciones, el óptimo típicamente ocurre en los límites entre las áreas factibles y las no factibles. La penalidad fuerza a la búsqueda genética para que se aproxime al óptimo desde ambas regiones.

3.3.2. LEGALIDAD DE UN CROMOSOMA

La legalidad de un cromosoma, se refiere al fenómeno de que un cromosoma represente una solución a un problema dado. La ilegalidad del cromosoma se origina en la naturaleza de la técnica de codificación. Para muchos problemas de optimización combinatorial se usan codificaciones específicas al problema, con las cuales se producen vástagos ilegales si se aplica la operación de crossover básica de un punto. Ya que, un cromosoma ilegal no puede decodificarse en una solución, en consecuencia el cromosoma no puede evaluarse; es decir que la penalidad no es aplicable en esta situación. Técnicas reparadoras se usan generalmente para convertir un cromosoma ilegal en uno legal. Por ejemplo el conocido operador PMX (el cual se introduce en el capítulo 5) es, esencialmente, un tipo de crossover de dos puntos para representaciones por permutación junto con un procedimiento para resolver la ilegalidad causada por un simple crossover de dos puntos. Orvosh y Davis [112] muestran que para muchos problemas de optimización combinatorial, es relativamente fácil reparar un cromosoma ilegal o no factible.

3.3.3. UNICIDAD DE UN CROMOSOMA

La función de traducción (mapping en inglés) desde el cromosoma a las soluciones (decodificación) puede pertenecer a uno de los siguientes casos:

1º. Traducción 1 a 1.

2º. Traducción *n* a 1.

3º. Traducción 1 a *n*.

En el primer caso es el mejor de los tres, mientras que el tercero es el menos deseado. Necesitamos considerar estos problemas cuidadosamente cuando se diseña una nueva codificación no binaria para construir un algoritmo genético efectivo.

3.4. SELECCIÓN

El principio que secunda a los algoritmos genéticos es, fundamentalmente, la selección natural Darwiniana. La selección impone la dirección en el proceso de búsqueda, donde la presión en la selección es crítica. En un extremo, la búsqueda termina prematuramente; mientras que en el otro, el progreso puede ser más lento de lo necesario. En general una baja presión selectiva se sugiere al inicio de la búsqueda genética, para favorecer la exploración del espacio de búsqueda; mientras que una alta presión selectiva se recomienda al final para explotar las regiones más prometedoras del espacio de búsqueda.

Una mayor presión selectiva restringe la cantidad de individuos a recombinar. Por ende se pierde diversidad poblacional (o diversidad de individuos). La pérdida de diversidad poblacional, el número limitado de generaciones y el tamaño de la población son causas de la convergencia prematura a un óptimo local antes de encontrar el óptimo global o un valor cercano al mismo.

Un parámetro asociado a la presión selectiva es el *takeover time*, que se define en el trabajo de Goldberg y Deb [71], como el número de generaciones necesarias para que el mejor individuo encontrado en la población inicial cope la población al aplicar repetidamente sólo un determinado mecanismo de selección. Si el takeover time es grande entonces la presión selectiva de un operador de selección es débil, si es pequeño ocurre lo contrario.

La *diversidad poblacional* fue introducida por Bäck y Hoffmeister [5], en términos de la medida del bias definida por Grefenstette:

$$b(P(t)) = \frac{1}{l \cdot \mu} \sum_{j=1}^{l} max \left(\sum_{\substack{i=1 \\ a_{i,j}^t = 0}}^{\mu} (1 - a_{i,j}^t), \sum_{\substack{i=1 \\ a_{i,j}^t = 1}}^{\mu} a_{i,j}^t \right)$$

donde l es el largo del cromosoma y $a^t_{i,j}$ denota el valor del alelo. El bias b $(0.5 \leq b \leq 1.0)$ indica el porcentaje promedio de mayor valor en cada posición del individuo. Valores pequeños de b indican una alta diversidad genotípica y viceversa. El bias b puede ser usado para formular un adecuado criterio de finalización.

La selección dirige la búsqueda genética hacia regiones prometedoras en el espacio de búsqueda. Los principales tópicos que involucra esta fase son:

- El espacio de muestreo.
- Los mecanismos de muestreo.
- La probabilidad de selección.

3.4.1. ESPACIO DE MUESTREO

El procedimiento de selección puede crear una nueva población para la próxima generación basada en cada uno de los padres y los hijos o partes de ellos. Un espacio de muestreo es caracterizado por dos factores: *tamaño* e *integrantes* (padres o hijos). Donde μ denota el tamaño de la población y λ denota la cantidad de hijos producidos en cada generación. Un espacio de muestreo regular tiene el tamaño μ y contiene a todos los hijos pero sólo a una parte de los padres [30, 77, 82, 107]. El espacio de muestreo extendido tiene el tamaño de $\mu + \lambda$ y contiene el total de padres y de hijos [5, 6, 53, 128].

3.4.1.1. ESPACIO DE MUESTREO REGULAR

En los algoritmos genéticos simples, los padres son reemplazados por sus hijos. Esto es denominado *reemplazo generacional*. Con esta estrategia, se pueden perder muy buenas soluciones en el proceso evolutivo. Ya que los vástagos pueden ser peores que sus progenitores. Para superar este problema, se han

propuesto distintas *estrategias de reemplazo*. Holland sugiere que cuando un hijo nace, reemplace a un cromosoma de la población actual; elegido aleatoriamente [82]. De Jong propone una *estrategia de crowding* [30]. En el modelo de crowding, cuando un hijo nace, un padre es selleccionado para morir. Estos padres, son aquellos que más se asemejan al nuevo hijo. Esta semejanza se mide calculando la distancia Hamming entre los cromosomas.

En el trabajo de Holland, se lleva a cabo la selección de los individuos (padres) que van a ser recombinados; se forma una nueva población al reemplazar a los padres por sus hijos. Esto se denomina *plan reproductivo*.

En el trabajo de Grefenstette y Baker [77], se usa la selección para formar la próxima generación, generalmente con un mecanismo probabilístico.

Michalewicz da una descripción detallada de algoritmos genéticos simples donde los vástagos en cada generación reemplazan a sus padres; la próxima generación se forma por medio de una selección *roulette wheel* (proporcional) [107]

La figura 3.2 ilustra la selección basada en un espacio de muestreo regular.

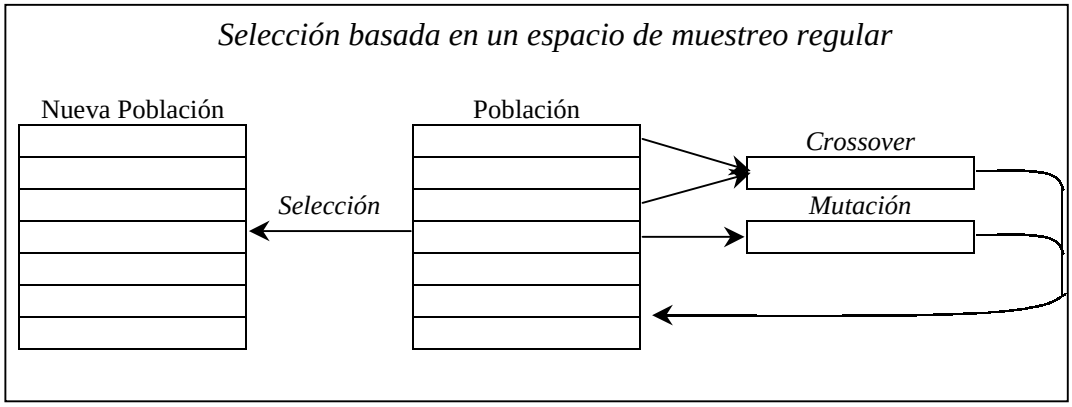

Figura 3.2. Selección realizada sobre un espacio de muestreo regular

3.4.1.2. ESPACIO DE MUESTREO EXTENDIDO

Cuando una selección se realiza sobre un espacio de muestreo extendido, tanto padres como hijos tiene la misma oportunidad por sobrevivir. Un caso típico es la selección (μ + λ) [53], usada originalmente en las estrategias evolutivas [128]. Bäck y Hoffmeister la introducen en los AGs [5, 6]. Con esta estrategia μ padres y λ hijos compiten por sobrevivir y los μ mejores son seleccionados como padres

en la próxima generación. Otro caso proveniente de las estrategias evolutivas, es la selección (μ, λ); en la cual se eligen los μ mejores hijos como padres de la próxima generación ($\mu < \lambda$). Ambos métodos son totalmente determinísticos y pueden convertírselos en probabilísticos.

Aunque la mayoría de los métodos se basan en un espacio de muestreo regular, es fácil implementar uno extendido. La figura 3.3 muestra la selección basada en un espacio de muestreo extendido.

Una ventaja evidente de esta opción es que se puede mejorar la performance del AG al incrementar las probabilidades de crossover y mutación. Tampoco es necesario preocuparse por la gran perturbación que pueden ocasionar estas altas probabilidades si la selección se realiza sobre un espacio de muestreo extendido. Ya que existe una probabilidad no nula de supervivencia de los padres.

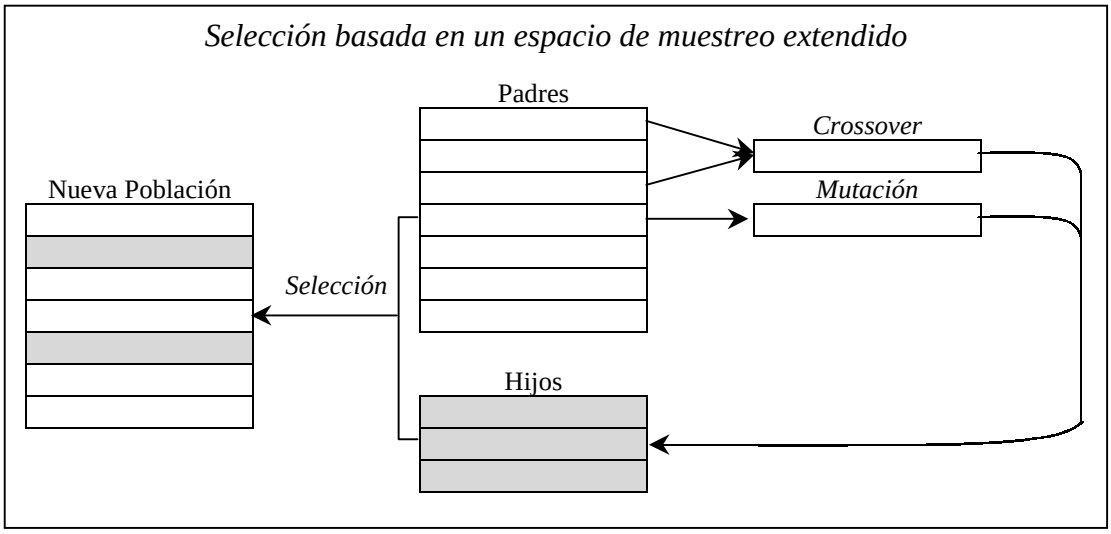

Figura 3.3. Selección realizada sobre un espacio de muestreo extendido

3.4.2. MECANISMOS DE SELECCIÓN

Los mecanismos de selección encierran el problema de cómo elegir cromosomas desde el espacio de muestreo. Las tres alternativas básicas usadas para el muestreo de individuos son:

- El muestreo estocástico.
- El muestreo determinístico.
- El muestreo mixto.

3.4.2.1. MUESTREO ESTOCÁSTICO

Una característica común en esta clase, dada por Baker, es que la fase de selección determina el número actual de copias que cada cromosoma va a recibir basado en la probabilidad de sobrevivencia (o de selección) [11]. Es decir que la fase de selección está compuesta por dos partes:

1. La determinación del valor esperado del cromosoma.
2. La conversión de los valores esperados al número de hijos.

El valor esperado de un cromosoma, es un número real que indica el número medio de vástagos que un cromosoma debería recibir. El procedimiento de muestreo es usado para convertir un valor esperado real al número de hijos.

El método mejor conocido de esta clase es la *selección proporcional* (de Holland) o *selección de la ruleta*. La idea básica es determinar la *probabilidad de selección* para cada cromosoma proporcionalmente a su valor de fitness. Para el cromosoma a_i con fitness $f(a_i)$, su probabilidad de selección p_{sel} es calculada como sigue:

$$P_{sel}\, f(a_i) = \frac{f(a_i)}{\sum_{j=1}^{\mu} f(a_j)}$$

Los individuos son ubicados en segmentos continuos de una recta, de forma tal que el segmento correspondiente a cada individuo es proporcional en tamaño a su fitness. Se genera un número aleatorio y se selecciona el individuo cuyo segmento alcanza dicho número. El proceso se repite hasta que se obtiene el número deseado de individuos (matting pool).

Baker [10, 11] propone el *muestreo estócastico universal* (conocido en inglés como Stochastic Universal Sampling – SUS). Aquí también, los individuos son ubicados en segmentos continuos sobre una línea, de manera tal que el segmento correspondiente a cada individuo es igual en tamaño a su fitness. Los punteros a cada segmento están separados en distancias iguales sobre la línea dependiendo de la cantidad de individuos a ser seleccionados. Si se considera N el número de individuos a seleccionarse, entonces la distancia entre los punteros es de $1/N$, la

posición del primer puntero se genera aleatoriamente dentro del rango [0, 1/*N*]. SUS asegura una selección de hijos más exacta que el método anterior. La figura 3.4 ilustra este método.

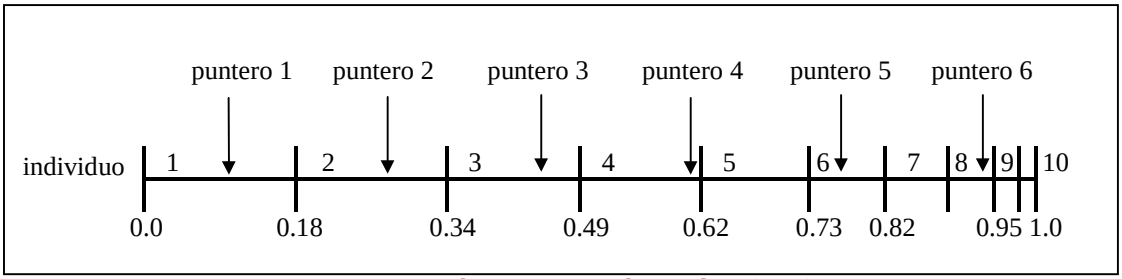

Figura 3.4. Stochastic Universal Sampling (SUS).

La consideración básica de SUS es mantener el número esperado de copias de cada cromosoma en la próxima generación. Existen dos razones para usar esta estrategia:

1. Previene la aparición de super cromosomas desde la población dominante al mantener demasiadas copias de estos en la misma. Esto es causa de convergencia prematura.
2. Mantiene la diversidad poblacional ya que el matting pool puede contener mucha más información para la búsqudea genética.

3.4.2.2. MUESTREO DETERMINÍSTICO

Selecciona los mejores μ cromosomas desde el espacio de muestreo. Los métodos de selección $(\mu+\lambda)$ y (μ,λ) hacen esto [6]. Ambas prohiben los cromosomas duplicados durante la selección.

La *selección por truncamiento* y la *selección por bloques* pertenecen a esta clase, y ordenan a todos los cromosomas de acuerdo a su fitness y selecciona a los mejores como padres [140]. En la selección por truncamiento un umbral *U* es definido tal que los *U*% mejores cromosomas son seleccionados y cada uno recibe $100/U$ copias. La selección por bloques es similar a la anterior, si para un tamaño de población dado μ, se da simplemente *s* copias a los mejores μ/s cromosomas. Ambas implementaciones son idénticas cuando $s = \mu/U$.

La *selección elitista* asegura que el mejor cromosoma se pase a la nueva generación si este no fuese elegido a través de otro proceso de selección.

El reemplazo generacional puede ser visto como otra versión de la opción determinística. Donde el conjunto total de padres se reemplaza por el de sus hijos [107, 137, 147].

3.4.2.3. MUESTREO MIXTO

Esta opción contiene en forma simultánea características aleatorias y determinísticas. Un ejemplo típico es la *selección por torneo* dada por Goldberg [70]. Este método aleatorio elige un conjunto de cromosomas (el tamaño de este conjunto se denomina *tamaño del torneo*) y toma el mejor de ellos para formar el conjunto de padres (matting pool) para la reproducción. Un tamaño de torneo común es 2, y se denomina *torneo binario*.

En la *selección por torneo estocástico* sugerida por Whetzel [145], las probabilidades de selección se calculan y pares sucesivos de cromosomas se eligen por medio de un método de selección proporcional o uniforme. Después de lo cual el cromosoma con fitness más alto se inserta en la nueva población. El proceso continúa hasta que la población está completa.

En el *muestreo estocástico remanente* propuesto por Brindle [16], cada cromosoma se muestra de acuerdo a la parte entera del número esperado. Compitiendo por las posiciones restantes en la población, de acuerdo a la parte fraccionaria del número esperado.

3.4.3. PROBABILIDAD DE SELECCIÓN

En este punto se trata cómo determinar la probabilidad de selección para cada cromosoma. La *probabilidad de selección* es un parámetro importante de un mecanismo de selección y normalmente determina el número de copias esperadas de un individuo después de la selección.

En un procedimiento de selección proporcional, la probabilidad de selección es proporcional a su fitness. Este esquema simple exhibe algunas propiedades a

considerar. Por ejemplo, en generaciones tempranas existe la tendencia de que unos pocos super cromosomas dominen el proceso de selección; en generaciones más tardías, cuando la población es convergente, la competencia entre cromosomas es más débil y la búsqueda se convierte en aleatoria.

Los mecanismos de selección por *escalamiento* y *ranking* son propuestos para mitigar estos problemas. El primero convierte a los valores de la función objetivo en algunos valores reales positivos, y la probabilidad de selección para cada cromosoma es determinada de acuerdo a estos valores. El método de selección por ranking ignora los valores de la función objetivo actual y usa un ordenamiento de cromosomas en lugar de determinar una probabilidad de sobrevevivencia.

El escalamiento de fitness tiene una doble intención:

1. Mantener una diferencia razonable entre los valores de fitness relativos.
2. Prevenir un rápido copamiento de la población por unos pocos super individuos, al limitar la competencia en las primeras generaciones y estimularla más tarde.

La mayoría de los métodos de escalamiento usan parámetros dependientes del problema. El ordenamiento (ranking) de fitness tiene un efecto similar al escalamiento de fitness pero evita la necesidad de escalamiento extra [115].

El escalamiento de valores de la función objetivo ha sido ampliamente aceptado en la práctica y una amplia variedad de mecanismos de escalamiento se han propuesto. Goldberg [69] y Michalewicz [107] han hecho una buena compilación de todos ellos.

3.5. MUTACIÓN

El operador de mutación para los algoritmos genéticos ha sido introducido por Holland como un operador secundario que ocasionalmente cambia un único bit del individuo al invertirlo [82]. La probabilidad de mutación p_m, por bit, pertenece al intervalo [0, 1] y es usualmente muy baja en los AGs. Algunas configuraciones comunes son $p_m = 0.001$ [30], $p_m = 0.01$ [74] y p_m en el intervalo [0.005, 0.01] en [121]

La forma del operador de mutación, $m'_{\{p_m\}} : S \to S$, usando la inversión del bit por cada posición que sufre mutación, produce un string de bits $\vec{a}' = (a'_1,....,a'_l) = m'_{\{p_m\}}(\vec{a})$ de acuerdo a:

$$a'_i = \begin{cases} a_i, & \text{si } x_i > p_m \\ 1 - a_i, & \text{si } x_i \le p_m, \end{cases} \qquad \forall i \in \{1,....,l\}$$

donde $x_i \in [0,1]$.

Bajas probabilidades de mutación, garantizan que un individuo producido por mutación no difiera genéticamente demasiado de sus ancestros.

3.6. RECOMBINACIÓN

Mientras que la mutación en los algoritmos genéticos sirve como un operador para introducir alelos perdidos en la población, el operador de crossover es el más importante en los GAs [9]. Mediante la recombinación, segmentos útiles de diferentes padres se combinan para obtener un nuevo individuo beneficiado con la combinación de bits de sus padres. De esta manera, se espera que substrings más grandes de alto fitness surjan, arribando finalmente a una buena solución.

El crossover en los algoritmos genéticos es siempre un operador sexual $r'_{\{p_c\}} : S^2 \to S$ que con probabilidad p_c selecciona dos padres $\vec{s}$ y $\vec{v}$, y las recombina para formar dos nuevos individuos. Las probabilidades de crossover propuesta son $p_c = 0.6$ [80], $p_c = 0.95$ [74] y $p_c \in [0.75,0.95]$ en [121]. El tradicional *crossover de un punto* introducido por Holland elige aleatoriamente una posición de corte $j \in \{1,...,l-1\}$ dentro del cromosoma e intercambia los genes a la derecha de esta posición entre ambos individuos [82], resultando en:

$$\vec{s}' = (s_1,....s_{j-1}, s_j, v_{j+1},...., v_l)$$

$$\vec{v}' = (v_1,....v_{j-1}, v_j, s_{j+1},...., s_l)$$

Este operador de crossover tiene una performance claramente inferior a la de otros operadores de crossover. El crossover de un punto sufre una fuerte dependencia de la probabilidad de intercambio sobre las posiciones de los bits.

Existe una asimetría entre la aplicación del crossover y la mutación: la unidad básica de la mutación es el gen, la del crossover es el individuo; en consecuencia, la probabilidad de que un individuo se cruce no depende de su longitud, mientras que la probabilidad de que contenga genes mutados es más alta cuanto más largo sea.

A continuación se describen los tipos de crossover más comunes:

- *Crossover Multipunto* [39]*:* consiste en combinar los individuos en torno a dos o más puntos de corte. Este cruce mejora la capacidad de procesado de segmetnos, pero a costa de perder velocidad de convergencia.
- *Crossover Segmentado* [39]: Es una versión del crossover multipunto que permite la introducción de variabilidad en el número de puntos de corte. Consiste en reemplazar el número fijo de los puntos de corte con una probabilidad de segmentación p_{seg} que da cuenta de la posibilidad de que se produzca un cruce al llegar a cierto punto del string. Así se trata de corregir la asimetría con respecto a la mutación.
- Por ejemplo, si la probabilidad de segmentación es $p_{seg} = 0.20$ entonces el promedio esperado de cruces será $0.2 \cdot l$ en cada par de progenitores, aunque no necesariamente tomará ese valor.
- *Crossover Uniforme* [137]: para cada bit del primer descendiente se decide, con probabilidad p_{unif}, de qué progenitor heredará su valor en esa posición. El segundo descendiente recibe el correspondiente gen del otro progenitor. El crossover uniforme se usa para combinar atributos específicos, independientemente de la posición en que han sido codificados. No obstante se recomienda utilizar el crossover uniforme sólo en casos concretos en los que haya motivos fundados para hacerlo.