

20
2ef



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
"ACATLAN"

SISTEMAS DE ADAPTACION ARTIFICIAL PARA
PROBLEMAS DE OPTIMIZACION.

T E S I S

QUE PARA OBTENER EL TITULO DE:

A C T U A R I O

P R E S E N T A :

JORGE SANCHEZ SANTIAGO

ASESOR: ACT. ISABEL RODRIGUEZ REBOLLEDO



268023
NOVIEMBRE 1998

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A mis padres

A mis hermanos, Adolfo, Rosa y Chela; a Sofía

A la familia Santiago

A Adrianita y familia

A Didier y familia

A David y familia

A Victor, Esteban, Juan Carlos, Asbel, Jorge y Bety

INDICE

Introducción		1
Capítulo 1 Presentación Preliminar		
1.1	Nociones básicas del análisis de algoritmos	4
1.2	Sistemas de adaptación	5
1.3	Un sistema de adaptación artificial	8
Capítulo 2 Marco Formal		
2.1	Componentes de un plan de adaptación	13
2.2	Comparaciones entre planes de adaptación	16
2.3	Planes de adaptación sobre el conjunto $(\mathbf{a}, \Omega, I, \tau)$	17
Capítulo 3 Esquemas		
3.1	Características comunes entre componentes	20
3.2	Desempeño asociado a los esquemas	23
3.3	Esquemas en el intervalo definido de una función	27
3.4	Ensayos de Poisson para cuantificar esquemas	28
Capítulo 4 Asignación Óptima de Ensayos		
4.1	Errores de muestreo	32
4.2	El problema del bandido	33
4.3	Aplicación a los esquemas	36
Capítulo 5 Planes Reproductivos y Operadores Genéticos		
5.1	Planes reproductivos generalizados	39
5.2	Operadores genéticos – Cruce	45
5.3	Operadores genéticos – Inversión	56
5.4	Operadores genéticos – Mutación	59
Capítulo 6 Eficiencia de los Planes de Adaptación		
6.1	Planes del tipo $\mathbf{R}_1 (P_C, P_M)$	62
6.2	Aplicación de la asignación óptima de ensayos a los planes reproductivos	69
6.3	Una implementación simple de un plan de adaptación	71
Conclusiones		77
Glosario		78
Bibliografía		79
Anexo		80

INTRODUCCION

Cuando me encontraba en el segundo semestre de la Maestría en Ciencias Computacionales escuché hablar de un tema que me llamó mucho la atención, los *Algoritmos Genéticos*, aunque por su nombre, los había relacionado con cuestiones de la computación aplicadas a la genética.

Más adelante me di a la tarea de investigar qué eran realmente estos algoritmos, cuál fue mi sorpresa al descubrir que son procedimientos que se utilizan, entre otras cosas, para aproximar o resolver problemas de optimización.

Los algoritmos genéticos son un caso particular de los sistemas de adaptación artificial propuestos por John H. Holland¹, éstos últimos basados en los mecanismos de la selección natural y los principios de la genética.

Los sistemas de adaptación artificial consisten en planes de adaptación que localizan soluciones a problemas en un espacio de búsqueda. Su fortaleza se encuentra basada en su mecanismo o estrategia para realizar una búsqueda "inteligente" en un dominio de acción, los planes de adaptación pueden ser utilizados para resolver o aproximar problemas de optimización al representar como "individuos" o "estructuras" a las soluciones de un problema y adaptarlas a un medio ambiente al reproducirlas y combinarlas entre sí, determinando qué tan eficiente es cada una en el medio a través de una función de aptitud, que consiste en la función objetivo del problema en cuestión.

Los planes de adaptación funcionan para cualquier función objetivo, los planes sólo necesitan saber el valor de la solución en el medio confrontado (utilizando la función objetivo) y un pequeño conjunto de estructuras (soluciones) para desarrollar una búsqueda eficiente.

Este nuevo tipo de herramienta de optimización, está diseñada para atacar problemas que si intentáramos resolver podríamos llevarnos años, y aún así, no encontrar la solución óptima. Este tipo de problemas es conocido como *NP* (Non Polinomial) y se presentan diariamente en diferentes actividades (ver sección 1.1).

Ya que la actividad profesional del actuario va mas allá de los seguros y pensiones, sistemas o finanzas, es necesario que conozca de las nuevas herramientas que se están desarrollando para resolver problemas de optimización que hasta la fecha no tienen un algoritmo eficiente de solución (ver sección 1.1).

Uno de mis intereses principales al desarrollar esta investigación es el dotar a un actuario de los elementos básicos que son necesarios para iniciar el estudio de estos algoritmos, y es necesario que sean divulgados entre la comunidad actuarial porque en verdad son útiles para atacar problemas como los *NP*.

¹ John H. Holland. *Adaptation in natural and artificial systems*. Ann Arbor:University of Michigan Press 1992.

Este trabajo está basado en la obra de John H. Holland[1]. El estudio desarrollado por Holland es hasta la fecha la base de la gran mayoría de las investigaciones que se han realizado.

A través del estudio de diversos textos y artículos relacionados con los algoritmos genéticos, decidí escribir este trabajo basado en [1] porque contiene los fundamentos y los conceptos principales necesarios para comprender la base de los planes de adaptación y que a pesar del tiempo siguen siendo los pilares de esta nueva corriente de investigación.

Debo aclarar que en [1] se presentan una gran cantidad de conceptos, en este trabajo yo abstraigo lo principal y aquello necesario para comenzar el estudio de los planes sobre una base formal sin necesidad de algún conocimiento no adquirido a lo largo de nuestra carrera.

Incorporaré una serie de ejemplos que harán más sencilla la primera incursión en este estudio y procuré en lo posible manifestar esos pequeños detalles que permiten hacer más claro el entendimiento de un concepto a lo largo de la exposición, además, puse énfasis en extender la explicación de términos y procedimientos que son utilizados por un plan para su mejor comprensión.

Este trabajo tiene como objetivo principal mostrar a los sistemas de adaptación artificial como una nueva herramienta para atacar problemas de optimización que no tienen un algoritmo eficiente de solución (ver sección 1.1).

Ya que la optimización es un tema íntimamente ligado con el actuario por su preparación en la academia, es necesario que conozca y entienda los fundamentos de los sistemas de adaptación artificial para poder utilizarlos y determinar cuándo puede ser necesario o útil aplicarlos.

Espero que con esta pequeña contribución al campo actuarial se desarrollen nuevos trabajos con fines más prácticos; sin embargo, antes de poder dar ese paso, es necesario contar con investigaciones como ésta para iniciar lo que podría ser un nuevo reto para los actuarios.

Este trabajo está diseñado para que el lector al finalizar su lectura conozca y entienda los principales conceptos de los planes de adaptación y sea capaz de formular uno para resolver algún problema.

En el primer capítulo se da una presentación preliminar de los componentes que están involucrados en la tarea de adaptación; el sistema, el plan de adaptación y la medida de desempeño de los individuos en el medio o sistema; además se incluye un sistema de adaptación artificial para ilustrar estos conceptos.

El segundo capítulo muestra de manera formal los componentes de un plan de adaptación definiendo a éste como un proceso que realiza una trayectoria sobre el espacio de búsqueda a través de la aplicación de operadores sobre las estructuras (soluciones) y la información de ensayos anteriores.

Las soluciones del espacio de búsqueda pueden ser representadas como estructuras formadas a partir de la concatenación de atributos; para realizar comparaciones entre ellas es necesario introducir el concepto de esquema, el tercer capítulo muestra cómo se consigue este objetivo.

El capítulo cuarto describe un método para encontrar un equilibrio entre la exploración y la explotación de información mientras se está llevando a cabo la aplicación del plan.

El capítulo cinco introduce a los operadores genéticos generalizados junto con su aplicación en los planes de adaptación reproductivos. El cruce y la mutación son los operadores que permiten explotar de manera efectiva las aptitudes de los elementos y a su vez crean una búsqueda en el espacio de soluciones garantizando la exploración de todas las zonas.

El capítulo seis muestra la fortaleza de los planes y establece la aplicación de la asignación de ensayos a las estructuras basada en el método desarrollado en el capítulo cuatro. Además se incluye un prototipo desarrollado en el lenguaje de programación C de un plan reproductivo para optimizar una función.

1. PRESENTACIÓN PRELIMINAR

1.1 NOCIONES BÁSICAS DEL ANÁLISIS DE ALGORITMOS

Un algoritmo es un procedimiento no necesariamente secuencial, que toma un conjunto de valores de entrada y los transforma para producir un conjunto de valores de salida².

La complejidad temporal de un algoritmo es el número de unidades de tiempo que le toma procesar una entrada de tamaño n .

La complejidad temporal de un algoritmo puede ser representada a través de una notación asintótica. Para el estudio a realizar bastará la notación $O(g(n))$ que está definida como un límite superior para cualquier función $f(n)$ de la siguiente manera:

$$O(g(n)) = \{f(n) \mid \exists c > 0, n_0, \forall n \geq n_0, 0 \leq f(n) \leq cg(n)\}^3$$

Por ejemplo, supongamos que un algoritmo para obtener el factorial de un número n utiliza n unidades de tiempo, decimos que para cualquier entrada n , este algoritmo tiene complejidad $O(n)$ pues se cumple la relación anterior, esto es

$$O(g(n) = n) = \{f(n) = n \mid \exists c > 0, n_0, \forall n \geq n_0, 0 \leq f(n) \leq cg(n)\}$$

Se dice que un algoritmo es eficiente cuando su complejidad temporal puede ser expresada como un polinomio en n (el tamaño de la entrada).

Existen cierto tipo de problemas que no pueden ser resueltos en un tiempo polinomial por ningún algoritmo, es decir, la complejidad temporal de dichos problemas es de orden exponencial, por ejemplo $O(2^n)$.

Estos problemas se conocen como NP^4 y han sido objetos de mucho estudio, sin embargo, aún no se ha encontrado un algoritmo eficiente para resolverlos.

Entre algunos problemas NP se encuentran:

² J. Sánchez. Introducción al Análisis de Algoritmos. Trillas, 1998, p. 15.

³ op. cit., p. 17.

⁴ op. cit., p. 118.

El agente viajero, en donde un agente debe visitar todas las ciudades de una provincia, visitando cada una sólo una vez y regresar al punto de partida. El objetivo del agente es minimizar la distancia que recorrerá.

La mochila o ladrón inteligente, éste consiste en la selección de artículos que debe de hacer el ladrón antes de robarlos para maximizar el monto de su hurto, el problema está en que su mochila o saco tiene ciertas dimensiones que no puede exceder, entonces debe seleccionar los objetos teniendo en cuenta la capacidad o espacio disponible.

SAT (Satisfabilidad), dada una expresión lógica, verificar si existe alguna asignación de valores a las variables que la conforman que la haga verdadera o que la satisfaga.

Estos problemas tienen la misma dificultad, su espacio de soluciones es de orden exponencial.

El agente viajero debe evaluar las $(n-1)!/2$ permutaciones cíclicas entre las ciudades para determinar la ruta óptima. Sin embargo, cuando n es relativamente pequeño, digamos 100, el número de permutaciones cíclicas posibles a evaluar es enorme, tanto, que cualquier algoritmo determinista (que asegure convergencia) podría tomar años en evaluar todas las rutas posibles.

Para el problema de la mochila, el ladrón deberá evaluar todas las posibles combinaciones de artículos que no excedan en volumen el espacio que él dispone y tomar aquella que le dé la mayor ganancia. El ladrón tendrá que evaluar $2^n - 1$ combinaciones, donde n es el número de artículos que tiene para elegir.

En el problema SAT, cada variable puede tomar dos valores, verdadero o falso, con un número de variables n , la cantidad de posibles asignaciones a la expresión es 2^n .

Como ya se dijo, no existen algoritmos eficientes que resuelvan estos problemas o cualquier *NP*, entonces se deben de desarrollar métodos alternativos para aproximarlos, uno de éstos son los sistemas de adaptación.

1.2 SISTEMAS DE ADAPTACIÓN

Resulta intrigante el proceso que deben realizar la evolución y la adaptación para producir organismos cada vez más aptos y permitirles a éstos sobrevivir en el medio ambiente donde nacen y se desarrollan.

Para que un organismo pueda sobrevivir es necesario que se adapte a las condiciones que establece un cierto medio ambiente, de otra forma se puede esperar que desaparezca o se extinga por ser débil.

La adaptación entonces debe ser un mecanismo en el cual se vean involucrados principalmente dos aspectos:

1. La historia de los elementos funcionales de los organismos, es decir, sus principales características.
2. El desempeño de los organismos en el medio conforme el proceso de adaptación se ha ido dando.

El mecanismo para resolver o aproximar un problema de optimización puede ser representado como un proceso de adaptación cuando las dimensiones del espacio de soluciones son de orden exponencial.

Una forma de atacar este tipo de problemas es utilizar un plan de adaptación que confronte al medio ambiente obteniendo individuos o estructuras que sobrevivan a él y hereden sus cualidades a generaciones posteriores en búsqueda de un incremento en su aptitud.

Para realizar la tarea de adaptación debemos identificar tres componentes principales:

1. E como el sistema que está bajo el proceso de adaptación⁵.
2. El plan de adaptación τ , por medio del cual los individuos son modificados para obtener incrementos en su aptitud⁶.
3. Una medida de aptitud o desempeño μ de los organismos en el sistema⁷.

El problema esencial para el plan de adaptación τ es que carece de información acerca de las estructuras con respecto a su aptitud en el sistema en el que se encuentran al inicio de su aplicación.

Podemos pensar, de manera intuitiva, que el plan debe hacer evaluaciones sobre las estructuras para determinar cuáles son mejores, al tomar como referencia su aptitud en el sistema.

El plan puede tener una tarea aún más complicada si existe más de un sistema E para confrontar, esto se debe a que el plan puede estar trabajando en la adaptación de las estructuras sobre un cierto sistema E y eventualmente este sistema puede cambiar a otro E' , entonces los resultados que había obtenido el plan de las estructuras sobre E pudieran dejar de ser útiles en E' .

⁵ John H. Holland. op. cit., p. 4.

⁶ ibídem.

⁷ ibídem.

Esta variedad de sistemas alternativos sobre los cuales el plan puede actuar lo obligan a reaccionar de manera distinta en cada una de ellos, los diferentes sistemas E serán contenidos en una clase \mathbf{E} .

El plan de adaptación realizará su aplicación sobre un conjunto de estructuras disponibles \mathbf{a} que representará el espacio de soluciones.

El hecho de que diferentes $E \in \mathbf{E}$ varíen el desempeño de una estructura dada $A \in \mathbf{a}$, significa que habrá una medida diferente de desempeño μ_E asociada a cada E ⁸.

Las modificaciones sucesivas a las estructuras dictadas por un plan τ se suman a una secuencia o trayectoria a través de \mathbf{a} (teniendo como resultado una búsqueda sobre este espacio de soluciones) que depende del sistema $E \in \mathbf{E}$ que se esté confrontando⁹.

En la operación detallada del plan se utilizan operadores que actúan sobre las estructuras, estos operadores serán determinados más adelante, por ahora es suficiente suponer que tenemos un conjunto Ω de ellos y el plan de adaptación los utilizará en el orden y presencia que sean necesarios para cada sistema $E \in \mathbf{E}$.

Entre los problemas específicos que debe enfrentar el plan de adaptación se encuentran dos que son primordiales:

- \mathbf{a} es de orden exponencial.
- La medida de desempeño μ_E es una función complicada, no lineal, con óptimos locales, discontinuidades, etc¹⁰.

Cuando se está tratando un problema de optimización se puede hablar de una incertidumbre con respecto a las posibles soluciones del problema, esto es, debemos conocer la función objetivo y suponemos que está no cambiará durante el proceso de evaluación de las soluciones, sin embargo, no sabemos qué tan buena es cualquiera hasta que la evaluemos.

Se puede entender a la función objetivo como la medida de desempeño de las soluciones.

Es posible atacar problemas de optimización definiendo una cierta clase de planes de adaptación, el presente trabajo mostrará la forma en que se construyen y probará su eficiencia, pero antes veamos una forma de ilustrar estas abstracciones con un ejemplo de un sistema de adaptación artificial.

⁸ op. cit., p. 5.

⁹ ibídem.

¹⁰ ibídem.

1.3 UN SISTEMA DE ADAPTACIÓN ARTIFICIAL

Supongamos que tenemos un dispositivo o "caja negra" que reacciona a impulsos luminosos cada vez que identifica un patrón, es decir, alguna característica común entre diferentes secuencias o configuraciones de impulsos.

El dispositivo tiene un conjunto de sensores o detectores (un arreglo bidimensional), con a unidades de alto y b unidades de ancho, por medio de los cuales se procesan los impulsos luminosos.

Cada sensor detecta luz y es activado cada vez que está recibiendo un impulso luminoso, por lo tanto cuando una escena le es presentada al arreglo de detectores en cierto momento del tiempo digamos t , cada uno de los sensores se prenderá o apagará dependiendo de la recepción de un impulso.

Sea $\delta_i(t)$ la actividad del i -ésimo sensor, $i = 1, 2, 3, \dots, ab$, en el instante del tiempo t , donde $\delta_i(t) = 1$ si el sensor se encuentra encendido y $\delta_i(t) = 0$ si está apagado.

Esto implica que una cierta secuencia de impulsos luminosos da origen a una combinación de ab ceros y unos. Es claro que el número total de posibles configuraciones es 2^{ab} .

El dispositivo indica, después de procesar los impulsos luminosos a través de los sensores, si la configuración presentada es una instancia del patrón que se encuentra recibiendo o no.

Formalmente sea

$$C = \{ \delta_1(t) \delta_2(t) \delta_3(t), \dots, \delta_{ab}(t) \mid \delta_i(t) \in \{0, 1\}, i = 1, 2, 3, \dots, ab \}$$

el conjunto que consiste de las 2^{ab} posibles configuraciones y sean

$$C_1 \subseteq C \qquad C_1 \cup C_0 = C \qquad C_1 \cap C_0 = \emptyset$$

con C_1 como el conjunto que corresponde a las configuraciones aceptadas por el dispositivo, es decir, el patrón en cuestión.

Inicialmente cualquier algoritmo desconoce las configuraciones que el dispositivo puede aceptar, podrían ser aquellas que contengan tantos unos como ceros, o bien las que tengan la mayor cantidad de unos, o ceros, o algún subconjunto del conjunto potencia de C .

Cada subconjunto del conjunto potencia de C representa un sistema $E \in \mathcal{E}$ diferente. Si el dispositivo no es dinámico entonces cualquier algoritmo determinista debe de hacer una búsqueda sobre las 2^{ab} posibles configuraciones para determinar cuales están siendo aceptadas. Es claro que tan sólo para $a = b = 10$ el espacio de búsqueda es enorme.

El problema es mayor si el dispositivo es dinámico, es decir, eventualmente puede cambiar de sistema $E \in \mathbf{E}$, esto provocaría que configuraciones que hayan sido determinadas como aceptadas después del cambio podrían dejar de serlo.

La tarea para cualquier algoritmo consiste, suponiendo que el dispositivo es estático, en determinar cual sistema $E \in \mathbf{E}$ está en vigor, esto es equivalente a encontrar el subconjunto $C_1 \subseteq C$ de configuraciones aceptadas por el dispositivo.

Para tratar de aproximar la solución de este problema por medio de un plan de adaptación se utilizará el siguiente método.

Inicialmente se generará de manera aleatoria un número, digamos M , de configuraciones de ab ceros y unos.

Los impulsos luminosos procesados por los sensores $\delta_i(t)$ asociados a cada configuración c_j ; $j = 1, \dots, M$, serán transformados a una representación decimal con la fórmula

$$\sum_{i=0}^{ab-1} (2^i) \delta_{i+1}(t) = K_j \quad j = 1, \dots, M$$

Cuando esta suma se encuentre en el intervalo inicial $[K_h, K_{h+1}]$; $h = 1, 2, 3, \dots$ se dirá que la configuración pertenece al conjunto C^+ y en otro caso a C^- , es decir

$$K_j \in [K_h, K_{h+1}] \Rightarrow c_j \in C^+$$

y

$$K_j \notin [K_h, K_{h+1}] \Rightarrow c_j \in C^-$$

Es claro que

$$C^+ \cup C^- = C \quad \text{y} \quad C^+ \cap C^- = \emptyset$$

es decir, C tiene una partición con dos clases, esto implica que C^+ se supondrá como una aproximación a C_1 .

El objetivo del plan de adaptación es entonces descubrir de la manera más rápida el intervalo $[K_h, K_{h+1}]$ que mejor aproxime C^+ a C_1 .

Es necesario hacer una aclaración, la forma de aproximar C^+ a C_1 para este ejemplo es muy pobre, sin embargo, estoy seguro que facilitará la comprensión de los principales componentes envueltos en la tarea de adaptación.

La función de desempeño μ_E , asociada al sistema $E \in \mathbf{E}$ en vigor (recordemos que suponemos estático al dispositivo), es la proporción de todas las configuraciones correctamente asignadas a $\{C_1, C_0\}$, esto es, μ_E asigna a cada intervalo $[K_h, K_{h+1}]$ una aptitud en función de las configuraciones que fueron bien asignadas.

$$\mu_E : [K_h, K_{h+1}] \rightarrow [0, 1]$$

En este ejemplo cada configuración junto con su evaluación en μ_E consiste en la entrada del plan de adaptación en cada generación.

Es claro que entre más sean las configuraciones, es decir, entre más grande sea el número M , el estimado de la aptitud de cada intervalo será más confiable

La determinación del intervalo inicial $[K_1, K_2]$ será hecha de manera aleatoria, cumpliendo las siguientes tres condiciones:

$$K_1 \geq 0$$

$$K_2 \leq \sum_{i=0}^{ab-1} (2^i)^{11}$$

$$K_1 \leq K_2$$

la segunda condición garantiza que bajo la transformación (descrita anteriormente) de las configuraciones a una representación decimal, el extremo derecho del intervalo no exceda el mayor valor que puede ser encontrado para configuraciones con ab unos.

El plan procederá de la siguiente forma:

1. Generar el intervalo inicial $[K_1, K_2]$.
2. Generar M configuraciones aleatoriamente
3. Determinar el desempeño promedio del intervalo en vigor.
4. Guardar el intervalo junto con su aptitud, si el intervalo ya existe se deberá recalcular el promedio con respecto a sus apariciones anteriores.
5. Si la aptitud del intervalo es superior o igual a 0.5 entonces el nuevo intervalo para la siguiente iteración del plan será el mismo de la iteración actual, empezar el ciclo desde el paso 2.

¹¹ En general, para la representación de números naturales en forma binaria, el primer bit de la expansión será el menos significativo. Esto tiene como objeto representar a las estructuras con una longitud fija (capítulo 3).

6. Si la aptitud es inferior a 0.5 entonces el nuevo intervalo para la siguiente iteración del plan será $[K_h-1, K_{h+1}+1]$, empezar el ciclo desde el paso 3.

Un punto que resalta de este problema es la magnitud o cardinalidad a (el conjunto de configuraciones), para este caso cualquier algoritmo se verá inicialmente en una incertidumbre total al inicio de su ejecución, no existe manera a priori de determinar qué configuraciones está aceptando el dispositivo sin intentar probar algunas.

Podemos definir una clase de planes o algoritmos enumerativos como aquellos que no toman ventaja de las evaluaciones anteriores de las entradas del plan en cada iteración.

Un ejemplo de plan enumerativo sería aquel que evalúa todas las posibles configuraciones y determina a través de esta búsqueda cual es el patrón en vigor.

Por lo tanto, teniendo tiempo suficiente (y estabilidad suficiente para que el sistema no cambie durante el proceso), un plan enumerativo garantiza encontrar el patrón para cualquier sistema que se esté confrontado¹².

Este tipo de planes con su garantía de encontrar el patrón en cuestión parecen ser ideales y útiles. Sin embargo, presentan un problema crítico que puede ser mostrado con el planteamiento de la pregunta

¿Cuánto es tiempo suficiente?

Una imagen clara de esto puede ser obtenida al analizar un sistema como el descrito por el ejemplo anterior donde a tiene 2^{100} estructuras, que para problemas reales sigue siendo un número "pequeño".

Un ejemplo sencillo es el agente viajero, el cual tiene una complejidad temporal de $O(n!)$ en una implantación normal, es decir, cuando el agente tiene que visitar 100 ciudades, el número posible de permutaciones cíclicas es $99!/2$, que excede por mucho a 2^{100} .

Dado lo anterior, un plan (o algoritmo) no puede ser calificado como "bueno" simplemente por asegurar convergencia.

Ya que diferentes estructuras pueden tener un desempeño diferente en diferentes sistemas, la tarea del plan es asignada por aspectos del sistema que son desconocidos inicialmente, es decir, el desempeño en el medio confrontado¹³.

El plan debe entonces crear una estrategia para generar estructuras con un desempeño alto en el sistema que se encuentre confrontando de manera eficiente.

¹² John H. Holland. op. cit., p. 16.

¹³ op. cit., p. 17.

En la discusión de planes de adaptación se deben enfatizar tres objetivos que deben cumplirse para lograr la eficiencia de los planes.

1. El plan adaptativo debe retener los avances ya hechos, junto con porciones de las interacciones que ha tenido con el sistema a través del tiempo¹⁴. Esto es hecho de manera automática conforme el plan es aplicado a través del tiempo y las estructuras van siendo modificadas y guardadas para futuras generaciones como se verá más adelante.
2. El plan debe utilizar la historia retenida para incrementar la proporción de estructuras aptas¹⁵. Esto se logra por medio de la recombinación de las principales características de dichas estructuras.
3. El plan debe expandir su búsqueda tanto como le sea posible para evitar que queden zonas sin explorar.

Comenzaremos el estudio formal para la descripción de los planes de adaptación describiendo sus componentes y estableciendo ciertas convenciones en el siguiente capítulo.

¹⁴ op. cit., p. 18.

¹⁵ ibídem.

2. MARCO FORMAL

2.1 COMPONENTES DE UN PLAN DE ADAPTACIÓN

Un plan de adaptación τ actuará de manera discreta con respecto al tiempo de su aplicación, sean estos momentos de tiempo $t = 1, 2, 3, \dots$, cada uno de estos momentos es una iteración del plan.

En adelante cada iteración del plan se denominará generación.

Como se vio en el capítulo anterior el plan procesa elementos de un conjunto \mathbf{a} , este será su dominio de acción.

Cada uno de estos elementos puede presentar características adicionales, esto nos da la idea de representaciones de elementos como composiciones de atributos, como lo son los cromosomas compuestos por genes donde cada uno de éstos tiene un conjunto de alelos asociado, es decir, un gen es alguna característica dentro del cromosoma, por ejemplo el color de ojos, los alelos representan el conjunto de colores que puede tomar el gen. Para esta exposición utilizaremos el término de atributo en vez de alelo.

Una estructura estará formada por la concatenación de atributos que la describan completamente. Por ejemplo, si deseamos optimizar una función de una variable que tiene como dominio un conjunto formado por números enteros, podemos representar a cada uno de estos números de forma binaria.

En general, para una estructura dada de longitud l , cada uno de sus l genes (atributos) tendrá un conjunto asociado con k_i atributos

$$A_i = \{a_{i1}, a_{i2}, a_{i3}, \dots, a_{ik_i}\}$$

Bajo esta suposición, el conjunto total de estructuras \mathbf{a} estará compuesto por todas las posibles combinaciones de los atributos de las estructuras¹⁶, es decir

$$\prod_{i=1}^l A_i$$

Es necesario aclarar que el conjunto de alelos asociado a cada gen en una estructura es el mismo para cualquier otra mientras no se indique lo contrario. La posición de cada gen en una estructura determina alguna característica común.

¹⁶ op. cit., p. 21.

La búsqueda sobre los elementos de \mathbf{a} se realizará de manera dinámica, es decir, el plan no necesita contar con el espacio total de soluciones, los elementos irán apareciendo conforme el plan aplique su estrategia.

Esta es una gran ventaja de los planes, realizan la búsqueda sin necesidad de generar y guardar inicialmente el espacio de búsqueda. Como se verá más adelante, los planes de adaptación generan una trayectoria sobre \mathbf{a} aplicando operadores de un conjunto Ω a las estructuras de la población vigente en una generación dada para obtener nuevos elementos, es de esta manera como el plan realiza su búsqueda.

Los diferentes sistemas a los que se puede enfrentar el plan generan una serie de impulsos I por medio de los cuales el plan determina la forma en que debe actuar.

Estos impulsos indican qué tan buena es una estructura en el sistema que se está confrontando, en el ejemplo presentado en el capítulo anterior la información que el dispositivo ofrecía con respecto a las configuraciones era aunque útil muy escasa. El dispositivo sólo indicaba si la configuración era aceptada o no.

Sin embargo, generalmente se puede obtener un poco más de información con respecto de las estructuras para determinar entre ellas cuál es mejor sin necesidad de que sean soluciones óptimas para el problema.

Esto implica que diferentes estructuras tendrán quizás diferentes impulsos, esto es, para $A \in \mathbf{a}$ y $A' \in \mathbf{a}$ con $A \neq A'$ se tiene que $I_A \neq I_{A'}$.¹⁷

Para la presentación formal del plan τ se utiliza $\mathbf{a}(t)$ como el estado del plan en la generación t , donde $\mathbf{a}(t)$ representa el conjunto de estructuras que está utilizando el plan en esa generación junto con la información que ha estado acumulando con respecto a otras estructuras que anteriormente trató.

La información de los impulsos de generaciones anteriores es guardada de manera implícita en las estructuras de la población vigente.

Para aclarar el punto anterior será necesario adelantar la operación detallada del plan que se verá en el capítulo 5.

El plan genera inicialmente una secuencia de estructuras por medio de algún procedimiento, después las modifica a través de la aplicación de ciertos operadores y las reemplaza por la población actual para formar la de la siguiente generación.

La historia de los impulsos está retenida (aunque no totalmente) en las nuevas estructuras para la siguiente generación debido a que para su creación sus ancestros fueron seleccionados de acuerdo a su aptitud en el sistema que se encuentran confrontando.

¹⁷ op. cit., p. 22.

Se puede perder información de los impulsos cuando se aplican los operadores a las estructuras, sin embargo el propósito de éstos últimos es proveer al plan de mejores individuos al combinar las características de sus ancestros.

Con esto se verifica la hipótesis acerca del uso que da el plan de adaptación a la historia de generaciones anteriores con respecto a las estructuras.

El plan puede ser representado entonces como una función de dos argumentos

$$\tau: I \times \mathbf{a} \rightarrow \mathbf{a} \quad 18$$

Aquí la estructura tratada en la generación $t + 1$, $\mathbf{a}(t+1)$, está dada por

$$\mathbf{a}(t+1) = \tau(I(t), \mathbf{a}(t)) \quad 19$$

El plan adaptativo τ puede ser tratado como un proceso que asigna una probabilidad a cada una de las estructuras pertenecientes a la generación en vigor en función de su aptitud con respecto al sistema en el que se encuentran.

Es decir, dado $I(t)$ y $\mathbf{a}(t)$ las varias estructuras $A_1, A_2, \dots, A_j, \dots$, de la generación actual serán tratadas por el plan con una probabilidad p_j ²⁰.

En la operación detallada del plan se ven involucrados una serie de operadores genéticos generalizados.

Una vez que el plan ha determinado las estructuras a modificar de la generación actual debe aplicarles los operadores que se hayan determinado previamente, es decir, el plan contemplará los operadores que utilizará antes de iniciar su ejecución.

El plan de adaptación podrá decidir de manera aleatoria si aplica los operadores o no a las estructuras, pero no agregará o quitará operadores que haya determinado al inicio de su búsqueda. Esto tiene como objeto hacer muy específica la descripción global del plan, pues se pretende mostrarlo como un algoritmo secuencial.

Los operadores $\omega \in \Omega$ pueden ser unitarios o binarios, y pueden ser representados de la siguiente forma

$$\omega_1: \mathbf{a} \rightarrow \mathbf{a}$$

para el caso unitario y

$$\omega_2: \mathbf{a} \times \mathbf{a} \rightarrow \mathbf{a}$$

para el caso binario.

¹⁸ op. cit., p. 23.

¹⁹ ibídem.

²⁰ ibídem.

2.2 COMPARACIONES ENTRE PLANES DE ADAPTACIÓN

Los planes de adaptación estarán contenidos en un conjunto Γ , y es necesario determinar cuáles son los mejores.

Un plan de adaptación puede variar en la forma en que aplica los operadores y su método de selección de las estructuras, la forma presentada en esta investigación se basa en la experiencia de una gran variedad de estudios realizados sobre este tema y mostrando aquélla que ha llevado a mejores resultados.

Cuando dos planes se comparan se tiene por supuesto que la aplicación de los operadores es útil en algún sentido y siempre buscando llevar a cabo las premisas presentadas en el primer capítulo.

Un problema de adaptación tiene una solución no trivial cuando desconoce las características del sistema $E \in \mathbf{E}$ que se encuentre confrontando. El plan estará dependiendo de la información que obtenga a partir de las señales o impulsos I .

Un caso de particular importancia es aquel en el cual un plan de adaptación recibe de manera directa información de las estructuras con respecto a su desempeño en el sistema, una parte de I será entonces una función de pago o medida de desempeño $\mu_E(\mathbf{a}(t))$ determinada por

$$\mu_E : \mathbf{a} \rightarrow \mathfrak{R} \quad ^{21}$$

Este trabajo se concentrará en aquellos planes que reciben información directa de las estructuras con respecto a su desempeño y no prestan atención otros factores, tendremos entonces que para cualquier generación t

$$I(t) = \mu_E(\mathbf{a}(t)) \quad ^{22}$$

En la presentación formal de los planes de adaptación se utiliza además un criterio denotado por χ para comparar la eficiencia de los planes $\tau \in \Gamma$ en los sistemas $E \in \mathbf{E}$.

La justificación de este criterio es sencilla, pese a que se tiene una medida de desempeño, las variantes sobre los sistemas pueden ocasionar que bajo algún plan una estructura tenga alta eficiencia pero sólo en algunos sistemas, mientras que otro plan puede inducir a las estructuras a un buen desempeño en global sobre \mathbf{E} .

Supongamos que se tiene un plan determinístico con la forma ya citada

$$\tau : I \times \mathbf{a} \rightarrow \mathbf{a}$$

²¹ op. cit., p. 25.

²² op. cit., p. 26.

se obtiene la función acumulativa de pago como

$$U_{\tau, E}(T) = \sum_{\tau} \mu_E(\mathbf{a}(\tau, t))^{23}$$

donde $\mathbf{a}(\tau, t)$ representa la estructura seleccionada por τ de acuerdo al sistema $E \in \mathbf{E}$ en la generación t , con la función de pago μ_E . El acumulado hasta T permite obtener la tasa promedio de pago τ con la función $U_{\tau, E}(T) / T$ ²⁴.

Cuando el plan asigna probabilidades a las estructuras y tiene la forma

$$\tau: I \times \mathbf{a} \rightarrow \mathbf{a}$$

es natural sustituir el pago esperado bajo la función de distribución de probabilidad $p(t)$ por

$$-\mu_E(\tau, t)^{25}$$

es decir

$$-\mu_E(\tau, t) = \sum_j p(A_j, t) \mu_E(A_j)^{26}$$

donde $p(A_j, t)$ es la probabilidad de seleccionar $A_j \in \mathbf{a}$ cuando la distribución sobre \mathbf{a} es $p(t)$, luego para los planes no deterministas se tiene

$$U_{\tau, E}(T) = \sum_{\tau} -\mu_E(\tau, t)^{27}$$

2.3 PLANES DE ADAPTACIÓN SOBRE EL CONJUNTO $(\mathbf{a}, \Omega, I, \tau)$

Una vez dados los elementos anteriores se dirá que un plan de adaptación basado en los elementos I , \mathbf{E} y χ puede describirse por el conjunto $(\mathbf{a}, \Omega, I, \tau)$ donde

²³ ibídem.

²⁴ ibídem.

²⁵ op. cit., p. 27.

²⁶ ibídem.

²⁷ ibídem.

$\mathbf{a} = \{ A_1, A_2, \dots \}$ es el conjunto de estructuras disponibles para el plan adaptación, es su dominio de acción²⁸.

$\Omega = \{ \omega_1, \omega_2, \dots \}$ es el conjunto de operadores para modificar las estructuras de \mathbf{a} ²⁹.

I es el conjunto de posibles entradas provenientes del sistema o medio ambiente³⁰

$\tau: I \times \mathbf{a} \rightarrow \mathbf{a}$ es el plan adaptativo que basándose en los impulsos o señales y las estructuras en cierta generación t determina la nueva población que será utilizada en el siguiente ciclo de su aplicación³¹.

Bajo esta interpretación tenemos

$$\tau(I(t), \mathbf{a}(t)) = \mathbf{a}(t+1)$$

Resumiendo, para determinar el conjunto $\mathbf{a}(t+1)$ el plan tomará una muestra aleatoria de $\mathbf{a}(t)$ de acuerdo a la función de probabilidad $p(t+1)$.

La secuencia de señales o impulsos

$$\{I(1), I(2), I(3), \dots, I(t-1)\}$$

está contenida en las estructuras de la población en la generación t , pues fueron formadas por atributos de sus ancestros por medio de la aplicación de los operadores.

Γ es el conjunto de planes factibles de la forma

$$\tau: I \times \mathbf{a} \rightarrow \mathbf{a}$$

Un conjunto de posibles sistemas \mathbf{E} que puede confrontar el plan.

²⁸ op. cit., p. 28.

²⁹ ibídem.

³⁰ ibídem.

³¹ ibídem.

Cuando una estructura $A \in \mathbf{a}(t)$ al momento t es tratada por el plan el sistema reacciona produciendo una señal $I(t)$. Esta señal contiene información importante para el plan, para el caso de planes con función de pago como única información se establece la función de pago o desempeño $\mu_E(\mathbf{a}(t)) = I(t)$ que mapea a los reales la eficiencia de las estructuras sobre E , siendo en general diferente en distintos sistemas.

Finalmente se establece un criterio χ para comparar la eficiencia de los diferentes planes en función de la acumulación de desempeño que tengan éstos sobre diferentes sistemas.

Dedo aclarar que en adelante supondremos que el medio al que se enfrenta el plan es estático, esto implica que la función de desempeño propuesta originalmente para algún problema no cambiará con respecto al tiempo, teniendo una estructura la misma aptitud en diferentes generaciones.

3. ESQUEMAS

3.1 CARACTERISTICAS COMUNES ENTRE COMPONENTES

Un plan de adaptación tiene una tarea muy difícil cuando el espacio de búsqueda, representado como \mathbf{a} , presenta un orden exponencial, además, la función de desempeño asociada al sistema en vigor μ_E puede ser complicada de evaluar o incluso de determinar, por esta razón es importante para el plan de adaptación contar con un mecanismo o estrategia que evite una búsqueda exhaustiva.

Es claro que la búsqueda en \mathbf{a} puede ser muy larga y aunque el plan de adaptación espera encontrar mejores elementos generación tras generación, no es posible determinar con exactitud cuándo va a finalizar su aplicación.

Sin embargo la estrategia que siguen los planes es eficiente, pues a través de la aplicación de los operadores y la forma de seleccionar a los elementos para formar una nueva población, provoca que la búsqueda no sea local, un hecho del cual se toma ventaja sobre los planes enumerativos descritos en el capítulo uno.

Para que el plan pueda aproximar de manera eficiente, la localización de las estructuras que llevan al valor óptimo en la función de desempeño, debe estar incorporando en cada generación aquellas que conlleven a un mejor desempeño promedio de la población vigente.

Es evidente que estas propiedades deben ser identificadas para ser explotadas, en este capítulo se mostrará la forma en la que el plan de adaptación consigue este objetivo.

Podemos definir de manera intuitiva que las características útiles de las estructuras se presentan como patrones en su composición.

Por ejemplo, supongamos que queremos optimizar la función

$$f(x) = x^3, \text{ para } x = 0, 1, 2, \dots, 15$$

es claro que $f(0) < f(1) < \dots < f(15)$.

Ahora, los valores de x pueden ser representados como cadenas binarias. Por ejemplo, los valores 0, 1, 2, 4, 8, 9, 11, 13, 14, 15 tienen las siguientes representaciones

$x = 0$	\Rightarrow	$x_b = 0000$
$x = 1$	\Rightarrow	$x_b = 1000$
$x = 2$	\Rightarrow	$x_b = 0100$
$x = 4$	\Rightarrow	$x_b = 0010$
$x = 8$	\Rightarrow	$x_b = 0001$
$x = 9$	\Rightarrow	$x_b = 1001$
$x = 11$	\Rightarrow	$x_b = 1101$
$x = 13$	\Rightarrow	$x_b = 1011$
$x = 14$	\Rightarrow	$x_b = 0111$
$x = 15$	\Rightarrow	$x_b = 1111$

Por medio de la vista, podemos identificar dos características importantes que tienen las cadenas con mejor desempeño al valorarlas en su forma decimal en la función (tomando en cuenta solamente los valores arriba citados); (i) mayor cantidad de unos y (ii) los mejores unos son aquellos que están más a la derecha en la cadena.

En este ejemplo pudimos determinar un patrón entre las cadenas a través del uso de la vista, lo que hicimos fue comparar las estructuras de las cadenas y asociarlas con su desempeño. Para un ser humano realizar este proceso es natural, pero el plan no tiene tal capacidad.

Entonces el plan debe comparar los elementos de \mathbf{a} determinando la forma en que contribuyen sus atributos en el desempeño de ellos.

Antes de continuar debo mencionar que los estudios actuales sobre planes de adaptación intentan descubrir la forma exacta en que los atributos contribuyen en las estructuras no sólo como composiciones, sino individualmente.

Sin embargo, hasta el momento los planes con su mecanismo de búsqueda han sido eficientes, como se verá más adelante la combinación de composiciones de atributos efectivamente se lleva a cabo, aunque no de una manera determinista.

Los elementos de \mathbf{a} tienen una representación bien definida, están compuestos por "concatenaciones" de atributos, es decir, cada elemento de \mathbf{a} tendrá la siguiente forma

$$(\delta_1(A), \delta_2(A), \delta_3(A), \dots, \delta_l(A))^{32}$$

donde cada δ_i es un atributo (gen)

$$\delta_i(A) \in V_i, i = 1, \dots, l^{33}$$

³² op. cit., p. 66.

³³ ibídem.

y cada V_i representa un conjunto valores que puede tomar el atributo i .

Cada estructura $A \in \mathbf{a}$ será entonces descrita por el orden de sus l atributos asignados por $\delta_i(A)$ ³⁴. Bajo este hecho dos estructuras sólo podrán ser distinguidas si son diferentes al menos en un atributo, supondremos en la exposición, sin pérdida de generalidad, que todos los elementos en \mathbf{a} son diferentes, ya que dos estructuras iguales tienen el mismo desempeño.

Para poder comparar las estructuras en términos de sus componentes, es necesario determinar subconjuntos de \mathbf{a} que tienen ciertos atributos en común, para esto se utilizará un símbolo comodín $\$$ ³⁵.

Este símbolo permite que esté presente cualquier atributo en una posición dada.

Debo aclarar de nuevo un aspecto importante, los atributos que puede tomar una estructura en una cierta posición son los mismos para cualquier otra en esa misma posición.

Supongamos que deseamos saber que característica presentan las representaciones binarias del ejemplo anterior con los valores mayores a 8.

Todos estos valores tienen en común el uno en la cuarta posición (el orden de las posiciones de los atributos se toma de izquierda a derecha), entonces tenemos que

$$\begin{aligned} (1,0,0,1) &\in [\$,\$, \$,1] \\ (0,1,0,1) &\in [\$,\$, \$,1] \\ (1,1,0,1) &\in [\$,\$, \$,1] \\ (0,0,1,1) &\in [\$,\$, \$,1] \\ (1,0,1,1) &\in [\$,\$, \$,1] \\ (0,1,1,1) &\in [\$,\$, \$,1] \\ (1,1,1,1) &\in [\$,\$, \$,1] \end{aligned}$$

y este esquema de elementos excluye a aquellos que no cumplen la condición, por ejemplo

$$\begin{aligned} (1,0,0,0) &\notin [\$,\$, \$,1] \\ (1,1,0,0) &\notin [\$,\$, \$,1] \\ (1,0,1,0) &\notin [\$,\$, \$,1] \end{aligned}$$

La condición para determinar el subconjunto de elementos que se desea puede ser tan específica como se quiera

$$(1,0,1,1) \in [1,\$,1,1]$$

³⁴ op. cit., p. 67.

³⁵ op. cit., p. 68.

$$(0,1,1,1) \in [0,1,\$,1]$$

$$(1,1,1,1) \in [1,\$,1,1]$$

El conjunto de todas las l -tuplas formadas por combinaciones del símbolo S y los conjuntos de atributos está dado por el producto

$$\Xi = \prod_{i=1}^l \{V_i \cup \{\$\}\}^{36}$$

Cada l -tupla $\xi = [\xi_1, \xi_2, \xi_3, \dots, \xi_l] \in \Xi$ se conoce como *esquema*³⁷ y al conjunto Ξ como *conjunto de esquemas*.

Cualquier l -tupla $\xi \in \Xi$ designa un subconjunto de \mathbf{a} de la siguiente manera:

Dado un elemento $A \in \mathbf{a}$, se dirá que A es una instancia de ξ , y por ende $A \in \xi$, si

1. cuando $\xi_i = \$$ cualquier atributo de V_i puede estar en la i -ésima posición de A .
2. cuando $\xi_i \in V_i$ el atributo de V_i presente en esa posición debe ser igual al de la i -ésima posición de A .

3.2 DESEMPEÑO ASOCIADO A LOS ESQUEMAS

El conjunto de esquemas permite asociar las características de las estructuras con el desempeño de sus instancias.

Si el objetivo del plan de adaptación es incrementar la calidad de los individuos que va encontrando (en realidad generando) generación tras generación, entonces debe buscar la manera de incrementar el promedio del desempeño de la población que esté tratando en cada ciclo de su aplicación con respecto a las anteriores.

El plan generalmente comienza su aplicación sobre un subconjunto de las posibles estructuras a tratar, este subconjunto es lo que hemos venido llamando población, son los elementos con los que puede tratar el plan en una generación dada.

Una forma de incrementar el desempeño promedio de la población al pasar a la siguiente generación, sería seleccionar aquellas estructuras que tiene los mejores

³⁶ *ibídem*.

³⁷ *ibídem*.

desempeños y reemplazarlas por otras de menor aptitud, sin embargo, con un procedimiento de este tipo, no se estaría realizando una verdadera búsqueda.

El plan de adaptación parte de una población inicial generalmente creada de manera aleatoria, los elementos de esta población son instancias de algunos esquemas.

Dados l atributos, una estructura $A \in \mathbf{a}$ es una instancia de 2^l distintos esquemas³⁸, es decir, A será instancia de un esquema si en cada posición de él se encuentra o bien un símbolo $\$$ o el mismo atributo de $A \in \mathbf{a}$ en esa posición.

Por ejemplo, la estructura 010 es instancia de los siguientes esquemas

[\$,\$,\$]	[\$,1,0]
[0,\$,\$]	[0,\$,0]
[\$,1,\$]	[0,1,\$]
[\$,\$,0]	[0,1,0]

Para este ejemplo suponemos $l = 3$ y que todas las posiciones tienen el mismo conjunto de atributos asociado, $V_i = \{0,1\}$ $i = 1, \dots, l$.

La selección que hace el plan de las estructuras de la población en vigor para modificar se ve afectada por el desempeño de éstas.

Esto es, la selección se hace de acuerdo a un experimento aleatorio sobre un espacio muestral que consiste de los elementos de la población vigente, en donde a cada elemento $A \in \mathbf{a}$ se le asigna una probabilidad en función de su desempeño. Este aspecto será tratado con más detalle en los capítulos 5 y 6, pero es necesario adelantar su exposición para resaltar algunas cualidades de los esquemas.

Entonces, los esquemas presentes a través de sus instancias en la población vigente, tienen un espacio muestral (quizás incompleto) que consiste de esas mismas instancias, además, cuando se selecciona un elemento de la población, se están seleccionando al mismo tiempo 2^l distintos esquemas.

Un esquema está definido dentro de un conjunto de h posiciones $\{i_1, \dots, i_h\}$ con $h \leq l$, para las cuales $\xi_i \neq \$$ ³⁹.

³⁸ op. cit., p. 69.

³⁹ op. cit., p. 72.

Si $V = \cup V_i$ y $|V| = k$, entonces todos los esquemas que se pueden definir sobre V están dados por

$$\Xi = \{V \cup \{S\}\}^l \quad 40$$

Podemos encontrar en total k^h esquemas distintos para cualquier conjunto de h posiciones tal que $h \leq l$ ⁴¹.

Cabe mencionar que, sobre este conjunto de h posiciones, cada miembro o estructura de $A \in V^l$ es una instancia de alguno de los k^h esquemas ⁴².

Para tener más claro el concepto anterior consideremos el conjunto $V = \{0,1\}$ y $l = 8$, es decir, definiremos esquemas para estructuras $A \in V^l$ de longitud 8 y con 2 atributos para cada posición, esto implica que \mathbf{a} tiene $2^8 = 256$ elementos.

Cualquiera de estas estructuras debe tener entonces un 1 o bien un 0 en la primera posición, lo que lleva a una partición de \mathbf{a} en dos clases, sean éstas los elementos que comienzan con 1 y los que empiezan con 0.

Supongamos que tenemos la estructura $\{10010000\}$, que es la forma abreviada de $\{1,0,0,1,0,0,0,0\}$, esta estructura es una instancia el esquema $\{1\$\$\$\$\$\$\}$ pero no de $\{0\$\$\$\$\$\$\}$, es decir tenemos un conjunto con dos esquemas

$$\{\{0\$\$\$\$\$\$\}, \{1\$\$\$\$\$\$\}\}$$

De igual manera \mathbf{a} se puede particionar en otras dos clases diferentes utilizando el conjunto con la posición $\{5\}$ como referencia

$$\{\{\$\$\$\$0\$\$\$\}, \{\$\$\$\$1\$\$\$\}\}$$

para las posiciones $\{1,4\}$

$$\{\{0\$\$0\$\$\$\}, \{0\$\$1\$\$\$\}, \{1\$\$0\$\$\$\}, \{1\$\$1\$\$\$\}\}$$

En general, para un número h de posiciones tal que $\{1 \leq i_1, i_2, i_3, \dots, i_h \leq l\}$ sobre la l -tupla, \mathbf{a} se puede ser particionar en las diferentes formas en que se pueden escoger el conjunto de posiciones anterior, el número total de particiones diferentes que se pueden generar sobre \mathbf{a} esta dado por

$$\sum_{i=1}^l \binom{l}{i} = 2^l - 1 \quad 43$$

⁴⁰ ibídem.

⁴¹ ibídem.

⁴² ibídem.

$n=1$

Decimos que un esquema puede tener su espacio muestral incompleto (desde el punto de vista del plan) porque en la población vigente pueden no estar presentes todas sus instancias.

Cuando se toma un elemento de la población se puede determinar con exactitud cual es su aptitud, sin embargo, no se puede calcular con la misma certeza cual es el desempeño de los esquemas de los cuales es instancia.

Supongo que en este punto el lector se preguntará cuál es la funcionalidad de los esquemas si aparentemente el plan utiliza a los impulsos y a las estructuras como su entrada.

Esta pregunta puede ser contestada parcialmente en este momento. El plan utiliza a las estructuras como base para guiar su búsqueda, pero lo que es realmente importante es la forma en que se pueden combinar las estructuras por medio de la aplicación de los operadores. El plan crea su búsqueda agregando y quitando atributos de las estructuras de acuerdo al desempeño que aportan éstos a ellas, entonces es necesario determinar cuales subestructuras mejoran el desempeño de los individuos. Esto se consigue con los esquemas.

Supongamos que el promedio de aptitud de la población en una cierta generación t está dada por $\mu(t)$.

Ahora supongamos que dentro de la población se encuentra presente un esquema ξ y que el desempeño promedio las instancias presentes en la población de este esquema está dado por $^*\mu_{\xi}(t)$.

Es claro que si

$$^*\mu_{\xi}(t) > \mu(t)$$

las instancias presentes de ξ tienen un mejor desempeño en promedio que aquel de la población en vigor.

Basándose en esto Holland propone el siguiente procedimiento⁴⁴:

- (i) Tratar instancias de varios esquemas hasta que al menos uno de estos, sea éste ξ , sea localizado y exhiba un promedio en una muestra $^*\mu_{\xi}$, tal que $^*\mu_{\xi} > \mu(t)$ y

⁴³ ibídem.

⁴⁴ op. cit., p. 69.

- (ii) Generar nuevas instancias de este esquema regresando al paso (i) hasta que $\mu(t)$ se acerque lo suficiente a $^*\mu_i$ para localizar un nuevo esquema, sea éste ξ' , que cumpla de nuevo $^*\mu_i > \mu(t)$.

Con este procedimiento el crédito será atribuido a una combinación de atributos de acuerdo con el desempeño observado de sus instancias.

3.3 ESQUEMAS EN EL INTERVALO DEFINIDO DE UNA FUNCIÓN

Para entender mejor el funcionamiento de los esquemas tomemos un caso concreto utilizando la función que se presentó en la sección 3.1

$$f(x) = x^3, \text{ para } x = 0, 1, 2, \dots, 15$$

El esquema [1\$\$\$] representa los valores 1,2,3,...15; [\$\$\$1] contiene a 8,9,...,15; [\$\$\$\$] contiene a todos los valores.

Teniendo cuatro valores, digamos 4, 5, 7 y 10, podemos estimar el promedio de diferentes esquemas

$$x = 4 \quad \Rightarrow \quad x_b = 0010$$

$$x = 5 \quad \Rightarrow \quad x_b = 1010$$

$$x = 7 \quad \Rightarrow \quad x_b = 1110$$

$$x = 10 \quad \Rightarrow \quad x_b = 0101$$

$$\xi_1 = [1$$$] \Rightarrow \quad ^*f_{1sss} = (f(5) + f(7)) / 2 = 234$$

$$\xi_2 = \{ \$0\$ \$ \} \Rightarrow \quad ^*f_{soss} = (f(4) + f(5)) / 2 = 94.5$$

$$\xi_3 = \{ \$1\$ \$ \} \Rightarrow \quad ^*f_{siss} = (f(7) + f(10)) / 2 = 671.5$$

El esquema $\xi_1 = [1$$$]$ con los dos valores que tiene como instancia posee un promedio de 234, para obtener el promedio real de él debemos calcular el valor de todas sus instancias

$$x = 1 \quad \Rightarrow \quad x_b = 1000$$

$$x = 3 \quad \Rightarrow \quad x_b = 1100$$

$$x = 5 \quad \Rightarrow \quad x_b = 1010$$

$$x = 7 \quad \Rightarrow \quad x_b = 1110$$

$$x = 9 \quad \Rightarrow \quad x_b = 1001$$

$$x = 11 \quad \Rightarrow \quad x_b = 1101$$

$$x = 13 \quad \Rightarrow \quad x_b = 1011$$

$$x = 15 \quad \Rightarrow \quad x_b = 1111$$

$$\xi_1 = [1$$$] \Rightarrow f_{1sss} = (f(1) + f(3) + f(5) + f(7) + f(9) + f(11) + f(13) + f(15)) / 8$$

$$= 1016$$

La diferencia entre el valor inicial obtenido y el real es demasiada, esto se debe a que sólo tratamos con dos de las ocho instancias que tiene el esquema, sin embargo, su promedio real está aún lejos del valor óptimo de la función.

Esto hecho va confirmando nuestra hipótesis sobre la forma en la que el plan trabaja. Recordando la forma en que calificamos a las estructuras en la sección 3.1, se infirió que las buenas estructuras eran aquellas que tenían un mayor número de unos o bien que tenían algunos unos a la derecha, este no es el caso del esquema ξ_1 .

Aunque el esquema ξ_1 contiene a la estructura con valor óptimo en la función existen otros esquemas que pueden tener un mayor promedio y contener a esta estructura también, además, el promedio de los cuatro valores es superior al observado del esquema, entonces, el plan debería tratar con otro esquema que rebase el promedio de la población.

Un esquema se puede considerar como "pobre" cuando tiene un número pequeño de posiciones definidas, esto provoca que una gran cantidad de estructuras sean instancias de él y entre estas se pueden encontrar algunas que tengan una aptitud muy baja que opaca a otras que son mucho mejores.

El plan debe incorporar mejores estructuras, y por ende mejores esquemas, al combinar características útiles de otras. Bajo el procedimiento mostrado en la sección anterior el plan espera producir estructuras que sean instancias de esquemas como

$$\begin{aligned}\xi_4 = [\$111] &\Rightarrow f_{\$111} = (f(14) + f(15)) / 2 \\ &= 3059.5\end{aligned}$$

3.4 ENSAYOS DE POISSON PARA CUANTIFICAR ESQUEMAS

Es útil determinar el número de esquemas que reciben al menos un número $n < N$ de ensayos cuando se seleccionan aleatoriamente N elementos de \mathbf{a} para mostrar una característica esencial de los planes de adaptación, el *paralelismo intrínseco*⁴⁵.

El paralelismo intrínseco es la cualidad que tiene un plan de adaptación de obtener información de diferentes esquemas al evaluar una estructura.

⁴⁵ op. cit., p. 71.

Nos referimos a ensayos como el número de instancias presentes de un esquema dado en la población en vigor.

Antes de mostrar la forma en que se hace el cálculo de este número creo conveniente mostrar, con un ejemplo sumamente sencillo, a que se refiere el paralelismo intrínseco.

Supongamos que $\mathbf{a}(1) \in \mathbf{a}$ consiste de los elementos $\{01, 11\}$ y que $V = \{0, 1\}$ con $l = 2$ y $|V| = k = 2$, el conjunto de esquemas Ξ estará formado por:

definidos en la posición $\{1\}$

$$\xi_1 = \{0\$ \} \quad \xi_2 = \{1\$ \}$$

definidos en la posición $\{2\}$

$$\xi_3 = \{\$0 \} \quad \xi_4 = \{\$1 \}$$

definidos en las posiciones $\{1, 2\}$

$$\xi_5 = \{00\} \quad \xi_6 = \{01\} \quad \xi_7 = \{10\} \quad \xi_8 = \{11\}$$

Los esquemas ξ_3, ξ_5, ξ_7 no tuvieron ensayos, mientras que $\xi_1, \xi_2, \xi_6, \xi_8$ tuvieron uno y ξ_4 tuvo dos, de aquí se puede ver que entre más posiciones definidas tiene un esquema menor es el número de sus instancias.

El número de instancias de un esquema puede ser determinado fácilmente, si el esquema en una posición dada tiene un símbolo S entonces puede tener $|V| = k$ instancias (considerando solo esa posición), si tiene un atributo en vez del símbolo comodín entonces sólo puede tener (de nuevo considerando solo esa posición) a una instancia, entonces el número total de instancias de un esquema está dado por k^{l-h} , donde h es el número de posiciones definidas del esquema y l es el número total de posiciones.

En promedio se puede esperar que un esquema definido en $h \leq l$ posiciones reciba (N/k^h) ensayos cuando se seleccionan N elementos de \mathbf{a} , en este caso k^h denota el número de esquemas definidos en un conjunto h posiciones⁴⁶.

Este promedio lo podemos utilizar como el parámetro λ de la distribución de Poisson, pues estamos interesados en determinar el número de esquemas que reciben al menos $n < N$ ensayos, la distribución de Poisson es una buena aproximación para nuestra tarea, pues estima el número de ocurrencias de algún evento por unidad de tiempo.

⁴⁶ op. cit., p. 73.

Para obtener la probabilidad de que un esquema definido en h posiciones reciba al menos $n < N$ ensayos se utiliza la función de distribución acumulativa

$$r(n, N) = \sum_{n'=n}^{\infty} ((N/k^h)^{n'} / n'!) \exp(-N/k^h) \quad ^{47}$$

Hasta este punto $r(n, N)$ solo representa la proporción de que un esquema reciba al menos n ensayos, tenemos un total de k^h esquemas en h posiciones, luego la proporción para estos es

$$r(n, N) = k^h \sum_{n'=n}^{\infty} ((N/k^h)^{n'} / n'!) \exp(-N/k^h)$$

que se pueden escoger de $l! / ((l-h)!h!)$ formas distintas, entonces

$$r(n, N) = (l! / ((l-h)!h!)) k^h \sum_{n'=n}^{\infty} ((N/k^h)^{n'} / n'!) \exp(-N/k^h)$$

finalmente tomamos el conjunto total de posiciones y obtenemos

$$r(n, N) = \sum_{h=1}^l (l! / ((l-h)!h!)) k^h \sum_{n'=n}^{\infty} ((N/k^h)^{n'} / n'!) \exp(-N/k^h) \quad ^{48}$$

como el número total de esquemas que reciben al menos $n < N$ ensayos.

La siguiente tabla muestra este comportamiento para valores fijos de $l = 32$ y $k = 2$

n	N	r(n, N)
8	16	186.3
8	32	3878.5
8	64	60844.2
16	32	50.7
16	64	1451.3

⁴⁷ op. cit., p. 72.

⁴⁸ ibidem.

Hasta ahora se ha supuesto que las mejores estructuras son instancias de esquemas que tienen un desempeño promedio observado por arriba del promedio de la población.

Si el plan pudiera de alguna manera retener la capacidad de los esquemas a través de las generaciones, entonces su búsqueda sería eficiente y conforme avance su aplicación se espera que sean encontrados (generados) mejores elementos.

El plan asigna de manera automática un rango a cada esquema. este rango consiste en el número de instancias presentes en la población que tiene dicho esquema, entonces, si el plan está desarrollando una estrategia eficiente, esperamos que

$$*\mu_{\xi}(t) > *\mu_{\xi'}(t)$$

donde el rango de ξ es mayor al de ξ' .

La estrategia que sigue el plan será detallada en el capítulo cinco. El siguiente capítulo aborda el tema de errores de muestreo, pues como ya se dijo, los esquemas pueden tener un espacio muestral incompleto, lo que ocasiona que el promedio observado de sus instancias esté muy lejos del real.

4. ASIGNACIÓN ÓPTIMA DE ENSAYOS

4.1 ERRORES DE MUESTREO

Un esquema es considerado útil cuando $*\mu_i$, el desempeño promedio observado de sus instancias, es superior al promedio de la población vigente, sin embargo, $*\mu_i$ es determinado a partir de una muestra, esto implica que el valor obtenido de $*\mu_i$ puede presentar una desviación considerable del promedio real μ_i ¹⁹.

El promedio observado de un esquema ξ está en función de sus instancias, entre mayor sea la diversidad de éstas mejor será la aproximación de $*\mu_i$ a μ_i , esto se debe a que en la población vigente podríamos encontrar como instancias de un esquema a aquellas que tienen la mejor aptitud entre todas las demás (siendo en general un esquema con desempeño bajo), una situación inversa se podría dar para otro esquema ξ' , el cual tiene como instancias a aquellas que tienen la peor aptitud, esto implica que siempre exista una probabilidad mayor a cero de que

$$*\mu_i > *\mu_{i'}$$

aún cuando

$$\mu_i > \mu_{i'}$$

Esto involucra un problema de decisión, elegir entre explorar para obtener más información o explotar la información que ya se tiene.

Por esta razón es necesario desarrollar un criterio por medio del cual el plan tenga un margen de seguridad aceptable en los rangos que asigna a los esquemas.

El plan puede asignar un ensayo a un esquema para explotarlo (bajo el supuesto de que el esquema tiene un buen desempeño) o puede asignarlo para explorarlo e incrementar la seguridad de su búsqueda. El plan debe elegir entre exploración y explotación, no puede hacer ambas cosas a la vez.

Si el plan utiliza en su estrategia la asignación de rangos a los esquemas en función del número de instancias que tienen, puede estar incurriendo en un error al asignar ensayos a un esquema que tiene desempeño real bajo pero que en ese momento tenía a sus mejores instancias.

Antes de continuar con el análisis del problema de decisión en el que está envuelta la tarea del plan es conveniente presentar un problema clásico de teoría de la decisión, el bandido.

¹⁹ op. cit., p. 75.

4.2 EL PROBLEMA DEL BANDIDO

Ésta es una de las maneras de hacer sencilla la formulación del problema del bandido, para nuestros fines es bastante clara y comprensible.

Considérense dos experimentos aleatorios

1. Lanzar la moneda m_1 y observar el resultado
2. Lanzar la moneda m_2 y observar el resultado

Los espacios muestrales asociados a cada experimento son

1. $E_1 = \{\text{cara, cruz}\}$
2. $E_2 = \{\text{cara, cruz}\}$

La probabilidad de obtener cara con la moneda m_1 es p_1 , de igual forma, la probabilidad de obtener cara con la moneda m_2 es p_2 .

Definamos dos variables aleatorias X_1 y X_2 asociadas a los espacios muestrales E_1 y E_2 respectivamente, esto es

$$X_1(\text{cara}) = 1, X_1(\text{cruz}) = 0, X_2(\text{cara}) = 1, X_2(\text{cruz}) = 0$$

Supongamos que las monedas son idénticas a la vista, sin embargo, una de ellas tiene mayor probabilidad de mostrar cara después de lanzarla.

Un jugador tiene la oportunidad de lanzar las monedas N veces, es decir, puede lanzar $N - n$ veces la moneda m_1 y las restantes n veces la moneda m_2 con $0 \leq n \leq N$, él sabe que una moneda es "mejor" que otra, obviamente no las puede distinguir porque son idénticas.

Si el objetivo del jugador es maximizar sus ganancias (obtener la mayor cantidad de caras) ¿qué estrategia debe seguir?

El jugador podría establecer como estrategia utilizar todas sus oportunidades en la moneda m_1 o bien en la moneda m_2 .

Otra estrategia (probablemente mejor que la anterior) sería determinar (con una probabilidad de error mayor a cero) cuál es la mejor moneda a través de un número pequeño de ensayos (lanzamientos).

Por ejemplo, si tiene derecho a 100 ensayos podría utilizar inicialmente 20, asignar 10 a cada moneda, y después comparar los promedios observados de los lanzamientos y elegir a la moneda con mejor "aptitud" observada para asignarle los restantes 80.

Desgraciadamente no se puede establecer con certeza, por medio de un número finito de ensayos, cuál de las dos variables aleatorias X_1 , X_2 posee realmente la media más alta cuando las distribuciones de probabilidad de X_1 y X_2 se traslapan.

Por ejemplo, X_2 podría presentar cara en los 10 ensayos y X_1 cruz en al menos uno, entonces

$${}^* \mu_{X_2} > {}^* \mu_{X_1}$$

aún cuando el valor promedio real de X_1 sea superior al de X_2 , es decir

$$\mu_{X_1} > \mu_{X_2}$$

Si seguimos la segunda estrategia propuesta anteriormente, y denotamos a $X_{(1)}$ como la variable aleatoria con mayor promedio observado y $X_{(2)}$ a la segunda mejor observada, el procedimiento sugerido tiene dos fuentes de pérdida claramente distinguibles, es decir, para cualquier número de ensayos n , $0 \leq n \leq N$, asignados a $X_{(2)}$, existe una probabilidad positiva, $q(N-n, n)$, de que $X_{(2)}$ sea realmente la variable aleatoria con la media más alta, las fuentes de pérdida asociadas al procedimiento son⁵⁰:

1. La mejor opción observada $X_{(1)}$, es realmente la segunda mejor, dado esto los $N-n$ ensayos asignados a $X_{(1)}$ incurren en una pérdida (esperada) acumulativa $(N-n)|\mu_{X_1} - \mu_{X_2}|$, esto ocurre con probabilidad $q(N-n, n)$.
2. La mejor opción observada $X_{(1)}$, es realmente la mejor, de acuerdo a esto los n ensayos asignados a $X_{(2)}$ incurren en una pérdida (esperada) acumulativa de $(n)|\mu_{X_1} - \mu_{X_2}|$, esto ocurre con probabilidad $(1 - q(N-n, n))$.

Entonces, la pérdida total esperada para cualquier asignación n de ensayos a $X_{(2)}$ y $N-n$ ensayos a $X_{(1)}$, puede ser estimada por la ecuación

$$L(N-n, n) = [q(N-n, n)(N-n) + (1 - q(N-n, n))(n)] |\mu_{X_1} - \mu_{X_2}| \quad {}^{51}$$

⁵⁰ op. cit., p. 77.

⁵¹ ibídem.

Estas pérdidas son ocasionadas por lo que Holland denomina exploración y explotación de la información en un problema donde se debe decidir entre diferentes alternativas y estimar cual es la mejor de acuerdo a cierto criterio de comparación⁵².

Holland realizó una aproximación de este problema determinando un número de ensayos n' que deben ser asignados a la segunda mejor opción observada con el fin de minimizar la pérdida total esperada, es decir, para todo $n \leq N$ se tiene

$$L(N - n', n') \leq L(N - n, n) \quad 53$$

Debo aclarar que esta aproximación será útil para reforzar la forma en que proceden los planes de adaptación haciendo las asignaciones de ensayos entre las estructuras, sin embargo, los cálculos realizados por Holland en [1] suponen el conocimiento de los promedios reales de las variables aleatorias X_1 y X_2 , un hecho que no conocemos a priori en el problema del bandido.

Esto no quiere decir que la aproximación de Holland sea "incorrecta", al contrario, como se verá más adelante, los planes siguen una estrategia de asignación muy similar a la que se realiza en el problema del bandido, y esta estrategia es otra de las bases importantes de los planes de adaptación, y que hasta la fecha, se sigue utilizando con buenos resultados en la práctica.

El valor de n' encontrado por Holland es aproximadamente igual a

$$b^2 \ln [N^2 / (8\pi b^4 \ln N^2)] \quad 54$$

donde

$$b = \sigma_{X_1} / (\mu_{X_1} - \mu_{X_2})$$

y X_1 se supone como la variable aleatoria con mayor promedio esperado. El valor de n' permite determinar el número de ensayos que se deben asignar a la mejor opción observada, Holland obtuvo que

$$N - n' \cong (8\pi b^4 \ln N^2)^{1/2} \exp(n' / 2b^2) \quad 55$$

⁵² op. cit., p. 75.

⁵³ op. cit., p. 77.

⁵⁴ ibídem.

⁵⁵ op. cit., p. 83.

Esto implica que la tasa de pérdida esperada será reducida de manera óptima, si el número de ensayos asignados a $X_{(1)}$ tiene un incremento un poco mayor que una función exponencial del número de ensayos asignados a $X_{(2)}$.

Para el caso del problema del bandido cuando se tienen r opciones la filosofía de asignación es similar, es decir, dadas r variables aleatorias

$$X_1, X_2, X_3, \dots, X_r$$

tal que sus medias se encuentran en orden decreciente

$$\mu_{X_1} > \mu_{X_2} > \mu_{X_3} > \dots > \mu_{X_r}$$

la mínima pérdida después de N ensayos debe exceder

$$(\mu_{X_1} - \mu_{X_r}) (r - 1) b^2 [2 + \ln [N^2 / (8\pi(r - 1)^2 b^4 \ln N^2)]]^{56}$$

donde

$$b = \sigma_{X_1} / (\mu_{X_1} - \mu_{X_r})$$

Debo aclarar, que el propósito de esta sección, sólo es mostrar un criterio de asignación que intenta seguir un plan de adaptación, por esta razón no incluyo los teoremas desarrollados por Holland, el lector interesado en un detalle más amplio de los cálculos mostrados en los párrafos anteriores podrá referirse a la obra de Holland [1].

4.3 APLICACIÓN A LOS ESQUEMAS

Se puede utilizar la estrategia de asignación de ensayos de la sección anterior para aproximar la forma en que los esquemas van ganando rango conforme el plan de adaptación va siendo aplicado generación tras generación.

El rango de un esquema es simplemente el número de instancias que tiene en una generación dada.

El problema de asignar rangos a los esquemas, de acuerdo a las instancias que tienen en la población actual, puede ser visto como un problema del bandido. El incremento del rango de un esquema está determinado en gran parte por el desempeño que tienen sus instancias.

⁵⁶ op. cit., p. 85.

El plan favorecerá entonces aquellos esquemas que presentan una aptitud alta, sin embargo, esto no se realiza de manera determinista (como se verá en los siguientes dos capítulos), lo que provoca la primera desviación de la solución aproximada propuesta por Holland.

En el problema del bandido la asignación de ensayos a la mejor opción observada debe ser aproximadamente igual a una función exponencial del número de ensayos asignados a las otra opciones.

Un plan de adaptación no puede seguir exactamente esta asignación pues la selección de estructuras es aleatoria (aunque los mejores elementos son favorecidos), sin embargo, la filosofía de dar mayor oportunidad a las mejores opciones observadas es llevada a cabo, el paralelismo intrínseco corrige los factores de pérdida adicionales que son ocasionados al dar un poco más de ensayos de lo sugerido a otras estructuras distintas de las "óptimas" presentes en la población actual.

El rango de los esquemas se encuentra en una pequeña muestra de estructuras que hemos venido manejando como población, el objetivo de asignación de ensayos del plan puede analizarse observando las cualidades de los esquemas presentes en la población en vigor.

Conforme avanza la aplicación del plan esperamos que los mejores esquemas ganen rango y aquellos con desempeño bajo sean eliminados, es decir, el plan provocará un incremento, en aquellas estructuras que se hayan manifestado como las mejores, en generaciones avanzadas.

Si un esquema es en realidad suficientemente apto para subsistir en generaciones avanzadas, el plan debe darle un mayor rango introduciendo más instancias de él en las poblaciones futuras.

Esto implica que, dada una secuencia de "observaciones" representada por las poblaciones futuras

$$\mathbf{a}(0), \mathbf{a}(1), \mathbf{a}(2), \mathbf{a}(3), \dots, \mathbf{a}(t)$$

el número total de instancias del esquema ξ en la población $\mathbf{a}(t)$ se espera que sea mayor al de la generación $(t - 1)$ si el esquema es útil para el plan.

El número de instancias de ξ , a partir de la aplicación inicial del plan, y hasta una cierta generación T , está dado por

$$\bigcup_{t=0}^T \mathbf{a}_\xi(t)$$

Si un esquema ξ debe persistir como el mejor observado, entonces la porción ξ en $\mathbf{a}(t)$ debe incrementarse de manera significativa con respecto al resto.

De forma intuitiva podemos pensar, que si la aptitud de los esquemas es factor clave para su reproducción, entonces su crecimiento esperado puede ser aproximado utilizando la diferencia entre el promedio observado del esquema y el promedio de la población actual, es decir

$$M_{\xi}(t+1) = (*\mu_{\xi}(t) - \mu(t)) M_{\xi}(t) \quad ^{57}$$

donde $M_{\xi}(t)$ representa el número de instancias de ξ en la generación t .

Todos estos puntos serán tratados con más cuidado y detalle en los siguientes dos capítulos, sin embargo, es necesario remarcar que el objetivo de haber presentado el problema del bandido es confirmar que la asignación de ensayos realizada por el plan, es útil y se aproxima a la minimización de pérdidas originalmente propuestas por Holland.

⁵⁷ op. cit., p. 88.

5. PLANES REPRODUCTIVOS Y OPERADORES GENÉTICOS

5.1 PLANES REPRODUCTIVOS GENERALIZADOS

Con los elementos presentados en los capítulos anteriores se puede comenzar la formulación detallada de los planes de adaptación. Como se ha estado mencionando los planes deben estar incorporando estructuras con mejor desempeño para hacer la búsqueda de manera eficiente.

La idea de incorporar elementos da la noción de planes de adaptación que reproducen parcial o totalmente a las estructuras.

La reproducción parcial se logra aplicando los operadores genéticos a las estructuras combinándolas con características de otras para obtener nuevos individuos.

Ya hemos hablado que los planes de adaptación utilizan un conjunto de estructuras $\mathbf{a}(1)$ para ser probadas y a partir de las cuales se crea o genera la búsqueda de otras. Este conjunto representa la población a ser utilizada en la primera generación. En general $\mathbf{a}(t)$ representará la población a utilizar por el plan en la generación t .

$\mathbf{a}(t)$ consta de estructuras formadas por "concatenaciones" de atributos como se vio en el capítulo 3. Cada una de éstas tiene asociada una aptitud o desempeño determinada por la función μ_E .

En general, las características esenciales de un plan de adaptación reproductivo están contenidas en los siguientes hechos:

- La probabilidad de que un elemento sea seleccionado por el plan es directamente proporcional a su aptitud.
- El plan, una vez seleccionada la estructura a modificar, hace una copia de ella y le aplica uno o más operadores genéticos. La aplicación de los operadores es aleatoria e independiente entre ellos. La copia de la estructura seleccionada puede tener una probabilidad positiva de no ser alterada por uno o más de los operadores.
- La nueva estructura que se obtuvo (pudiendo ser la misma que la seleccionada) reemplaza a otra de la población. Para la selección de esta estructura se asigna la misma probabilidad a cada una de ellas, esto implica que la población para la generación siguiente puede ser la misma. El desempeño de la nueva estructura debe ser guardado.

El sentido de reproducción de los elementos se encuentra en el reemplazo que se hace por la nueva estructura.

La ventaja que se tiene de hacer aleatoria la aplicación de los operadores está basada en la oportunidad de "convergencia" que pueda tener el plan al incorporar más de una misma estructura en la población y combinar sus atributos de manera distinta con otra. Este punto quedará más claro cuando se vea el operador de cruce.

Una clase de planes de adaptación reproductivos conocida como \mathbf{R}_1 en [1] está dada por el siguiente algoritmo

Algoritmo 5.1.1 ⁵⁸

1. Iniciar $t = 0$, seleccionar aleatoriamente M estructuras de \mathbf{a} para formar el conjunto $\mathbf{a}(t) = \{A_h, h = 1, 2, \dots, M\}$
2. Observar y guardar los desempeños $\{\mu_E(A_h), h = 1, 2, \dots, M\}$
3. Ir al paso 5
4. Observar el desempeño de $A'(t)$ y reemplazar $\mu_E(A_i)$ por $\mu_E(A')$
5. Incrementar t en 1
6. Seleccionar una estructura A_i de $\mathbf{a}(t)$
7. Aplicar los operadores $\omega_1, \omega_2, \dots, \omega_n$ a la copia de A_i para crear A'
8. Seleccionar A_i y reemplazarla por A' , ir al paso 4.

Ninguno de los algoritmos que se presentarán en adelante tiene una condición de fin, esto se debe a que son algoritmos de adaptación, por lo que se pueden realizar tantas iteraciones como se desee.

En el primer paso se generan aleatoriamente las estructuras, esto se puede hacer asignando a cada posición de una estructura un atributo seleccionado aleatoriamente del conjunto de atributos asociado.

El tamaño de la población depende generalmente de la capacidad de cómputo que se tenga, sin embargo, muchas de las aplicaciones actuales toman poblaciones que van desde 100 hasta 500 individuos.

⁵⁸ op. cit., p. 92.

El tamaño de la población se determina antes de la aplicación del plan y no varía durante su ejecución.

Aunque en el paso dos se guardan los desempeños de todas las estructuras, lo común ahora es sólo guardar (en un sentido histórico) aquella estructura que tuvo el mejor desempeño en una generación. Claro, para la población vigente, éstos se necesitan para determinar las probabilidades de selección.

La observación de los desempeños tiene como objeto verificar que la búsqueda se esté haciendo de manera correcta, recordemos que se espera que las estructuras estén mejorando en aptitud con respecto a la de sus ancestros.

En el paso seis el plan realiza un experimento aleatorio al hacer la selección, el espacio muestral consiste en las estructuras de la población vigente, donde cada una tiene una probabilidad directamente proporcional a su desempeño, esto es

$$p(A_h) = \mu_E(A_h) / \sum \mu_E(A_M)$$

El paso siete puede involucrar una segunda estructura en el caso de que el operador a aplicar sea binario, ésta se determina por selección como se hizo en el paso seis. Debemos recordar que los operadores ya fueron determinados antes del inicio de la aplicación del plan junto con su orden de aplicación.

La estructura que será reemplazada, como se ve en el octavo paso del algoritmo, puede ser cualquiera de la población, todas tienen la misma probabilidad

$$p_r(A_h) = 1 / M$$

Al inicio de cualquier generación, el plan cuenta con la población $\mathbf{a}(t)$, que incluye la información actualizada a través de la historia de las estructuras que ya han sido probadas, y la serie de impulsos o información del desempeño de las estructuras vigentes a través de la función μ_E .

Entonces los planes \mathbf{R}_1 encajan en marco formal presentado en el capítulo dos,

$$\tau: I \times \mathbf{a} \rightarrow \mathbf{a}$$

$$\tau([\mu_E(A_1(t)), \dots, \mu_E(A_M(t))], [A_1(t), \dots, A_M(t)]) = (A_1(t+1), \dots, A_M(t+1)) \quad ^{59}$$

Los planes reproductivos provocan, a través de su método de selección, que una estructura tenga una descendencia proporcional a su desempeño, esto se establece con el siguiente

⁵⁹ op. cit., p. 93.

Lema 5.1: Si en cualquier generación t , la probabilidad de que una estructura A produzca una descendencia es p_1 y p_2 denota la probabilidad de que A muera al final de esa generación, entonces la descendencia esperada de A esta dada por p_1/p_2 ⁶⁰.

Prueba: Inicialmente necesitamos determinar el número de generaciones en las que se espera que esté presente A , es decir, la vida esperada de A puede interpretarse como el número de generaciones consecutivas que sobrevive además de aquélla en la que apareció.

Para que una estructura esté presente en la generación t es necesario que haya sobrevivido en las $t - 1$ generaciones anteriores, esto tiene probabilidad

$$p(t) = (1 - p_2)^{t-1}$$

Es claro que en la generación $t = 1$, la estructura siempre estará presente pues apareció en ella, ahora, si su probabilidad de descendencia es p_1 , entonces su descendencia esperada en la generación t es

$$p_1(1 - p_2)^{t-1}$$

Empezando en la generación $t = 1$, la vida esperada de A está dada por

$$\begin{aligned} \sum_{t=1}^{\infty} (1 - p_2)^{t-1} &= \sum_{t=0}^{\infty} (1 - p_2)^t \\ &= 1 / [1 - (1 - p_2)] \\ &= 1 / p_2 \end{aligned}$$

El producto de la vida esperada de la estructura y su probabilidad de generar una descendencia entonces está dada por

$$p_1/p_2.$$

que representa la descendencia esperada de la estructura. ■

Es importante remarcar que los operadores pueden hacer que una estructura vuelva a aparecer después de que haya muerto, esto se confirmara más adelante, sin embargo, en el lema anterior no se considera tal caso.

⁶⁰ ibídem.

Debo aclarar que la siguiente igualdad puede no ser cierta

$$p_1 + p_2 = 1$$

p_1 y p_2 representan probabilidades de eventos no complementarios, por lo cual no debe tomarse a una como complemento de la otra.

Bajo los planes \mathbf{R}_1 , p_1 representa la probabilidad de que A sea seleccionada para ser modificada por los operadores, la descendencia A se ve manifestada en los atributos que le hereda a la nueva estructura A' (siendo posible que $A = A'$). La probabilidad de que una estructura desaparezca de la población será $1/M$.

La probabilidad de descendencia p_1 en los planes \mathbf{R}_1 puede no ser constante, al contrario de lo que se supuso en el lema, sin embargo, podemos utilizar a éste para tener una buena aproximación del desempeño de los planes con respecto a sus descendientes.

Si

$$p_1 = \mu_E(A_h) / \sum \mu_E(A_M)$$

y

$$p_2 = 1/M$$

entonces la descendencia esperada de una estructura A_h está dada por

$$\begin{aligned} (\mu_h / \sum \mu_M) / (1/M) &= \mu_h / (\sum \mu_M / M) \\ &= \mu_h / -\mu. \quad 61 \end{aligned}$$

De ahora en adelante $\mu = \mu_E$ pues suponemos que el medio al que se enfrenta el plan de estático.

$\mu_h / -\mu$ debe ser interpretado como la eficiencia de la estructura A_h con respecto al promedio de la población en alguna generación t .

Dado $\mu_h / -\mu$ se espera que una estructura pueda generar un descendiente cuando su promedio de aptitud sea superior al de la población.

En los planes \mathbf{R}_1 una estructura puede dar origen, en cada generación, solamente a un descendiente, los planes \mathbf{R}_2 en cambio, pueden generar $\mu_h / -\mu$ descendientes (utilizando algún redondeo cuando sea necesario).

⁶¹ op. cit., p. 94.

Algoritmo 5.1.2 ⁶²

1. Iniciar $t = 0$ y seleccionar M estructuras aleatoriamente para formar $\mathbf{a}(t)$
2. Observar y guardar los desempeños $\{\mu(A_i), i = 1, 2, \dots, M\}$
3. Incrementar t en 1, iniciar $j = 0$ y $b = 0$
4. Si $b < M$, ir al paso al paso 6
5. Sustituir $\mathbf{a}(t)'$ por $\mathbf{a}(t)$, ir al paso 2
6. Seleccionar A_h de $\mathbf{a}(t)$ para ser modificada, $j = \mu_h / -\mu$ redondeado a un entero
7. Si $j = 0$ ó si $b = M$, ir al paso 4
8. Aplicar los operadores $\omega_1, \omega_2, \dots, \omega_n$ a la copia de A_h para crear A_h'
9. Incrementar b en 1, guardar A_h' en la posición b en $\mathbf{a}(t)'$, disminuir en 1 a j , ir al paso 7

En este algoritmo j es un contador decreciente del número de descendientes producidos por el individuo h y b es un contador del número total de descendientes. Los planes \mathbf{R}_1 y \mathbf{R}_2 se pueden englobar en la clase \mathbf{R} por satisfacer un comportamiento de reproducción de acuerdo al desempeño.

Algoritmo 5.1.3 ⁶³

1. Iniciar $t = 0$ y seleccionar M estructuras aleatoriamente para formar $\mathbf{a}(t)$
2. Observar y guardar los desempeños $\{\mu(A_h), h = 1, 2, \dots, M\}$
3. Incrementar t en 1 e iniciar los parámetros para una nueva generación
4. Si el proceso de la generación ha terminado sustituir $\mathbf{a}(t)'$ por $\mathbf{a}(t)$, ir al paso 2
5. Modificar los parámetros para la producción de una nueva estructura
6. Seleccionar $A \in \mathbf{a}(t)$ para ser modificada
7. Aplicar los operadores $\omega_1, \omega_2, \dots, \omega_n$ a la copia de A para crear A'
8. Guardar A' en $\mathbf{a}(t)'$, ir al paso 4.

⁶² op. cit., p. 95.

⁶³ op. cit., p. 96.

Antes de estudiar a los operadores, es necesario mostrar algunas ventajas de manejar una población $\mathbf{a}(t)$.

$\mathbf{a}(t)$ funciona como una especie de depósito de esquemas presentes a través de sus instancias, en $\mathbf{a}(t)$ puede haber un número entre (2^l) y $(M2^l - M + 1)$ de esquemas dependiendo de la diversidad de la población.

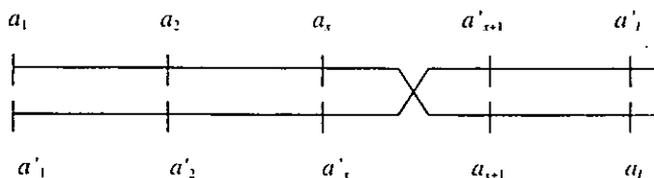
Conforme ha avanzado la aplicación del plan, se han agregado nuevos esquemas y otros han desaparecido de acuerdo al desempeño de sus instancias, esto puede interpretarse como la evolución que ha sufrido el conjunto $\mathbf{a}(t)$.

5.2 OPERADORES GENÉTICOS – CRUCE

El cruce es el operador básico de los planes de adaptación, a través de él los elementos de la población combinan sus atributos para generar nuevas estructuras. El cruce permite al plan desarrollar una búsqueda aleatoria dirigida en función del desempeño de las estructuras.

El operador puede describirse completamente por medio de los siguientes pasos⁶⁴

1. Se toman dos estructuras $A = a_1, a_2, \dots, a_l$ y $A' = a'_1, a'_2, \dots, a'_l$ seleccionadas aleatoriamente (las estructuras tienen una probabilidad de ser seleccionadas de proporcional a su aptitud) de la población actual $\mathbf{a}(t)$, donde a_i y a'_i son elementos del conjunto de atributos V .
2. Un número x es seleccionado aleatoriamente de $\{1, 2, \dots, l-1\}$, todos tienen la misma probabilidad.
3. Dos nuevas estructuras son formadas a partir de A y A' al intercambiar el conjunto de atributos a la derecha de x para formar $a_1, a_2, \dots, a_x, a'_{x+1}, \dots, a'_l$ y $a'_1, a'_2, \dots, a'_x, a_{x+1}, \dots, a_l$



⁶⁴ op. cit., p. 98.

Al incorporar el cruce a los planes **R** uno de los dos elementos nuevos deberá ser eliminado, ya que el objetivo del plan es incorporar mejores estructuras. deberá ser descartado aquel que tenga el menor desempeño, en caso de empate no importa cual sea desechado.

Ya que el cruce afecta a las estructuras, los esquemas que tienen como instancias a éstas se ven afectados en su rango de manera automática, si $\mathbf{a}(t)$ funciona como un depósito de esquemas es necesario analizar en que forma le está afectando el operador.

El cruce genera más instancias de esquemas que ya se encuentran en $\mathbf{a}(t)$.

Por ejemplo, sean

$$A = a_1 a_2, \dots, a_{i-1}, w, a_{i+1}, \dots, a_i$$

$$A' = a'_1 a'_2, \dots, a'_{i-1}, w, a'_{i+1}, \dots, a'_i$$

instancias del esquema

$$\xi = [\$ \$, \dots, \$, w, \$, \dots, \$ \$]$$

Después del cruce entre A y A' en cualquier posición, el esquema ξ tendrá una instancia más, en este caso suponemos que la nueva estructura instancia de ξ puede ya existir en $\mathbf{a}(t)$, podría ser la copia de A o A' , con lo que ξ tiene un incremento cuantitativo y no cualitativo en sus instancias.

El cruce genera "nuevas" instancias de esquemas que ya se encuentran en $\mathbf{a}(t)$ ⁶⁵.

Cuando se realiza el cruce entre A y A' en la posición $x = i$, el esquema ξ podrá tener una nueva instancia si se cumplen las siguientes dos condiciones

para $i > x$, $a_i \neq a'_i$; al menos en una posición

para $i \leq x$, $a_i \neq a'_i$; al menos en una posición

Por ejemplo, dadas las estructuras

$$A = 1000$$

$$A' = 1101$$

⁶⁵ ibídem.

un cruce en la posición $x = 2$ tendría como resultantes

$$\begin{aligned} A_1'' &= 1001 \\ A_2'' &= 1100 \end{aligned}$$

Entonces, asumiendo que ninguna de estas estructuras ya existía en la población, se generan nuevas instancias del esquema

$$\xi = [1SS1]$$

con la estructura A_1'' .

El cruce también genera "nuevos" esquemas⁶⁶. La misma estructura A_1'' da origen al primer ensayo del esquema

$$\xi' = [S001]$$

de nuevo asumiendo que A_1'' no existía en la población.

El cruce, al incrementar el rango de los esquemas, hace más confiable el desempeño promedio observado de éstos.

Lema 5.2.1: Sean $A = a_1 a_2, \dots, a_l$ y $A' = a'_1 a'_2, \dots, a'_l$ estructuras que difieren en los valores de sus atributos en x' posiciones a la izquierda de $x + 1$ y x'' a la derecha de x . Entonces cualquiera de las estructuras "nuevas" después del cruce de A con A' en la posición x será una instancia de $2^{l-x'} - 2^{l-x''} - 2^{l-(x'+x'')}$ esquemas "nuevos" y será una instancia más para $2^{l-x'} + 2^{l-x''} - 2^{l-(x'+x'')}$ esquemas que ya existen en $\mathbf{a}(t)$. Se asume que $x' \neq 0$ y $x'' \neq 0$ ⁶⁷.

Debo aclarar, que en este lema, se asume que las estructuras "nuevas" resultantes del cruce no se encontraban presentes en la población.

Prueba : Para que el cruce cree una nueva estructura, es necesario que los ancestros de ella difieran por lo menos en un atributo a la izquierda de $x + 1$ y a la derecha de x .

Esto implica que ningún esquema definido en una o más de las x' posiciones a la izquierda de $x + 1$ y una o más de las x'' posiciones a la derecha de x tendrá como instancia a A o A' .

El número de combinaciones que se pueden hacer con $j = 1, \dots, x''$ atributos y $(x'' - j)$ símbolos S está dado por

⁶⁶ ibídem.

⁶⁷ op. cit., p. 99.

$$\sum_{h=1}^l (t_h) = 2^l - 1 = 2^{x''} - 1$$

el mismo cálculo se realiza para las x' posiciones a la izquierda de $x + 1$

$$2^{x'} - 1$$

Ahora, para las restantes $(l - (x' + x''))$ posiciones no se tiene restricción, por lo tanto cualquiera de estas posiciones puede tomar un atributo o un símbolo S

$$2^{l-(x'+x'')}$$

El producto de estas tres cantidades es el número de esquemas "nuevos"

$$(2^{x'} - 1)(2^{x''} - 1)(2^{l-(x'+x'')}) = 2^l - 2^{l-x'} - 2^{l-x''} + 2^{l-(x'+x'')}$$

Como se supone que cualquiera de los resultantes del cruce no estaba presente y difiere de sus ancestros por lo menos en un atributo, entonces la diferencia entre los 2^l esquemas de los cuales es instancia y los "nuevos" que está aportando, son esquemas que ya estaban presentes y que ahora reciben una nueva instancia con esta estructura

$$2^l - (2^l - 2^{l-x'} - 2^{l-x''} + 2^{l-(x'+x'')}) = 2^{l-x'} + 2^{l-x''} - 2^{l-(x'+x'')}$$

Por ejemplo, dadas las estructuras

$$\begin{aligned} A &= 1010 \\ A' &= 0101 \end{aligned}$$

y el punto de cruce $x = 2$, se obtienen los siguientes resultantes del cruce

$$\begin{aligned} A_1'' &= 1001 \\ A_2'' &= 0110 \end{aligned}$$

la estructura A_1'' genera los siguientes esquemas "nuevos"

$$\begin{aligned} \xi_1 &= [1001] \\ \xi_2 &= [S001] \\ \xi_3 &= [1S01] \\ \xi_4 &= [10S1] \\ \xi_5 &= [100S] \end{aligned}$$

$$\xi_6 = [S00S]$$

$$\xi_7 = [1SS1]$$

$$\xi_8 = [S0S1]$$

$$\xi_9 = [1S0S]$$

El operador de cruce genera, como se vio en el lema anterior, esquemas nuevos y también instancias nuevas para ellos.

Dado esto, sería útil determinar aproximadamente el tiempo que le lleva al plan incorporar un esquema cuando éste no se encuentra presente en la población.

Limitaremos la funcionalidad del plan momentáneamente para establecer las probabilidades de ocurrencia de los esquemas.

El plan seguirá partiendo de una población inicial y aplicará el operador de cruce a los esquemas (representados a través de sus instancias) seleccionándolos de manera uniforme, es claro que entre más instancias tenga un esquema mayor será su probabilidad de ser seleccionado.

Una vez que el plan ha seleccionado a dos estructuras para el cruce, lo lleva a cabo y los individuos resultantes reemplazan a sus ancestros, después de esto, el plan comienza de nueva su aplicación en la siguiente generación.

Entonces, bajo la repetición continua del cruce sobre los elementos la población inicial, un esquema tendrá una probabilidad de ocurrencia $\lambda(\xi)$ bien definida, es decir, $\lambda(\xi)$ será la misma aún después de aplicado el operador a la población.

Con lo anterior podemos tomar a $1/\lambda(\xi)$ como una medida razonable del tiempo esperado de una ocurrencia de ξ .

Lema 5.2.2: La acción repetida del cruce, con selección uniforme aleatoria de estructuras en una población $\mathbf{a}(t)$ y en la ausencia de algún otro operador, lleva a un "estado fijo", en el cual un esquema ξ ocurre con probabilidad

$$\lambda(\xi) = \prod_j P_j(\xi_j)$$

donde $P_j(\xi_j)$ es la proporción total en $\mathbf{a}(t)$ del atributo que ocurre en la j -ésima posición de ξ , (si en dicha posición se encuentra un símbolo S entonces se debe tomar $P_j(\xi_j) = 1$)⁶⁸.

Prueba⁶⁹: Sean ${}^x\xi_1$ y ${}^y\xi_2$ los resultados del cruce entre ξ_1 y ξ_2 en el punto x .

⁶⁸ op. cit., p. 100.

⁶⁹ ibidem.

Por la definición del operador, si cruzamos a ${}^x\xi_1$ y ${}^x\xi_2$ en el punto x tendremos de vuelta a ξ_1 y ξ_2 .

Sea $P(\xi)$ la proporción de instancias del esquema ξ en $\mathbf{a}(t)$.

Entonces la probabilidad de que ξ_1 y ξ_2 sean seleccionados para el cruce es

$$P(\xi_1)P(\xi_2)$$

La probabilidad de obtener ${}^x\xi_1$ y ${}^x\xi_2$ del cruce de ξ_1 y ξ_2 en el punto x está dada por

$$P(\xi_1)P(\xi_2)P_x$$

donde P_x es la probabilidad de que el cruce se realice en el punto x . De manera similar, la probabilidad de obtener a ξ_1 y ξ_2 del cruce entre ${}^x\xi_1$ y ${}^x\xi_2$ en el punto x es

$$P({}^x\xi_1)P({}^x\xi_2)P_x.$$

Si sólo se consideran los efectos del cruce sobre los pares ${}^x\xi_1, {}^x\xi_2$ y ξ_1, ξ_2 en el punto x no habrá cambios en sus probabilidades de ocurrencia si

$$P(\xi_1)P(\xi_2)P_x = P({}^x\xi_1)P({}^x\xi_2)P_x.$$

Si y sólo si esta ecuación es válida para todo x y toda cuádruple $(\xi_1, \xi_2, {}^x\xi_1, {}^x\xi_2)$ no habrá cambio en la probabilidad de ocurrencia de cualquier esquema.

Para balancear todas estas ecuaciones simultáneamente notamos en inicio que el conjunto de atributos $\{, \xi_1, , \xi_2\}$ es idéntico a $\{, {}^x\xi_1, , {}^x\xi_2\}$ ya que después del cruce los mismos atributos están presentes, entonces

$$P(, \xi_1)P(, \xi_2) = P(, {}^x\xi_1)P(, {}^x\xi_2)$$

Por lo tanto si $P(\xi) = \prod_j P(, \xi_j)$ como enuncia el lema se tiene que para cualquier x y cualquier cuádruple $(\xi_1, \xi_2, {}^x\xi_1, {}^x\xi_2)$

$$\begin{aligned} P(\xi_1)P(\xi_2) &= (\prod_j P(, \xi_{1j}))(\prod_j P(, \xi_{2j})) \\ &= \prod_j P(, \xi_{1j})P(, \xi_{2j}) \\ &= \prod_j P(, {}^x\xi_{1j})P(, {}^x\xi_{2j}) \\ &= (\prod_j P(, {}^x\xi_{1j}))(\prod_j P(, {}^x\xi_{2j})) \\ &= P({}^x\xi_1)P({}^x\xi_2). \end{aligned}$$

Esto implica que las ecuaciones serán balanceadas si el esquema ocurre con probabilidad

$$\lambda(\xi) = \prod_i P_i(\xi)$$

Este lema es pieza clave del estudio de los planes de adaptación, como se vio, la probabilidad de aparición de un esquema está en función de la proporción que tienen sus atributos en la población.

Esto implica que conforme el plan avanza, la asignación de más ensayos a mejores esquemas se incrementa, además, el cruce guía al plan por el desempeño de los individuos, entonces se puede esperar que los mejores atributos ganen también un rango mayor.

Debemos aclarar que ningún plan reproductivo **R** se aproximará a un equilibrio en las probabilidades de aparición de los esquemas, al menos en un inicio de su aplicación, el plan guía su búsqueda por medio del desempeño y la reproducción de mejores individuos, esto ocasiona que las estructuras con bajo desempeño paulatinamente desaparezcan y los esquemas asociados a ellas pierdan instancias.

Una estructura tiene probabilidad $2/M$ de ser involucrada en una operación de cruce cuando $\mathbf{a}(t)$ contiene M elementos, por lo tanto en N generaciones se espera que un individuo participe en el cruce $2N/M$ veces⁷⁰.

Cuando N se encuentra en la vecindad de $lM/2$, donde l representa la longitud de los elementos (el número de atributos que poseen), se espera que la estructura experimente un cruce en casi todas sus posiciones, sin embargo, los esquemas pueden ser afectados en diferentes magnitudes dependiendo de su longitud⁷¹.

La longitud de un esquema es la "distancia" que existe entre los extremos de sus posiciones definidas, esto es

$$l(\xi) = (i_h - i_1) \quad 72$$

con $i_1 < i_2 < \dots < i_h$ representando el índice de las posiciones definidas del esquema.

Entonces, la probabilidad de que el punto de cruce caiga entre los extremos de las posiciones definidas del esquema una vez que alguna de sus instancias fue seleccionada es

$$l(\xi)/(l-1) \quad 73$$

⁷⁰ op. cit., p. 101.

⁷¹ ibídem.

⁷² op. cit., p. 102.

⁷³ ibídem.

Por ejemplo supongamos que la estructura $a_1 a_2 a_3 a_4 \dots a_l$, que es una instancia de $\xi = \$a_2 \$a_4 \$ \dots \$$ fue seleccionada. Dado el punto x para el cruce, la probabilidad de que éste caiga entre las posiciones definidas de ξ es $2 / (l - 1)$.

De aquí que los esquemas con longitud pequeña sean menos propensos a ser afectados por el cruce.

Los atributos que definen a un esquema que tiene longitud pequeña y que además exhiben un desempeño por arriba del promedio serán tratados con mayor frecuencia, más modificaciones serán hechas y recibirán más ensayos junto con combinaciones de otros esquemas similares favorecidos en otras posiciones definidas, esto es establecido de manera precisa por el siguiente

Teorema 5.2.3: Considérese un plan reproductivo del tipo **R** usando solamente el operador de cruce, utilizando dos precursores y el punto de cruce seleccionado aleatoriamente de manera uniforme. Entonces la proporción esperada en la siguiente generación de cada esquema presente en $\mathbf{a}(t)$ cambia de $P(\xi, t)$ a

$$P(\xi, t+1) \geq [1 - P_c(l(\xi)/(l-1))(1 - P(\xi, t))](\mu^*(\xi(t)/\mu(t))P(\xi, t))$$

donde P_c es la proporción de los individuos que puede ser afectados en una generación por el cruce y $\mu(t)$ es el promedio observado del desempeño de $\mathbf{a}(t)$ ⁷⁴.

Prueba⁷⁵: Durante cada generación cada una de las estructuras $A \in \mathbf{a}(t)$ espera producir $\mu(A)/\mu(t)$ descendencias bajo un plan reproductivo del tipo **R**.

Entonces cada instancia $\mathbf{a}_\xi(t)$ de ξ en $\mathbf{a}(t)$ espera producir descendientes de acuerdo a su desempeño, la suma de estas descendencias esperadas $M'_\xi(t)$ es el número de descendientes que espera generar el esquema ξ . Si $M_\xi(t)$ representa el número de instancias de ξ en esa generación entonces

$$M'_\xi(t) = (\sum_{A \in \mathbf{a}_\xi(t)} \mu(A)) / \mu(t)$$

$$M_\xi(t) \cdot \sum_{A \in \mathbf{a}_\xi(t)} \mu(A) / M_\xi(t) = \mu^*(\xi(t)) \cdot M_\xi(t)$$

$$M'_\xi(t) = \mu^*(\xi(t)) M_\xi(t) / \mu(t)$$

⁷⁴ ibídem.

⁷⁵ ibídem.

Hasta este punto el plan sólo ha seleccionado las estructuras para ser modificadas por el operador.

Debemos recordar que la aplicación de los operadores es aleatoria, es decir, una estructura cuando es seleccionada puede no verse afectada por el operador, para este caso P_c denota la probabilidad de que cualquier estructura una vez seleccionada sea afectada por el cruce.

Entonces, dado P_c y la longitud $l(\xi)$ del esquema ξ , se espera que la proporción

$$P_c l(\xi) / (l - 1)$$

de $M'_\xi(t)$ tendrá un cruce sobre las posiciones definidas de ξ . Este es un factor de pérdida, cuando un esquema sufre un cruce en el intervalo de sus posiciones definidas se "destruye".

Para asegurar que los resultantes del cruce sean instancias del esquema ξ , es necesario que los ancestros también sean instancias de ξ .

La probabilidad de que una instancia de ξ sea cruzada con otra también de ξ es

$$P(\xi, t)$$

esto se debe a que la selección de las estructuras funciona como un muestreo con reemplazo. Entonces se espera que una proporción no mayor a

$$(1 - P(\xi, t)) P_c l(\xi) / (l - 1)$$

de la descendencia de ξ puede ser instancia de otro esquema que distinto de ξ , el complemento

$$[1 - (1 - P(\xi, t)) P_c l(\xi) / (l - 1)]$$

será entonces la proporción de las instancias de ξ que fueron seleccionadas para el cruce y que no será alterada, es decir, la proporción de las instancias de ξ en $t + 1$ estará acotada inferiormente por

$$\begin{aligned} P(\xi, t + 1) &= M_\xi(t + 1) / M \\ &\geq [1 - (1 - P(\xi, t)) P_c l(\xi) / (l - 1)] M'_\xi(t) / M \\ &= [1 - P_c (l(\xi) / (l - 1)) (1 - P(\xi, t))] [(*\mu_\xi(t) M_\xi(t) / \mu(t)) / M] \\ &= [1 - P_c (l(\xi) / (l - 1)) (1 - P(\xi, t))] [* \mu_\xi(t) / \mu(t)] [M_\xi(t) / M] \\ &= [1 - P_c (l(\xi) / (l - 1)) (1 - P(\xi, t))] [* \mu_\xi(t) / \mu(t)] P(\xi, t) \end{aligned}$$

Podemos notar que el cruce aplicado a los precursores que no son instancias de ξ pueden conducir a una nueva instancia de ξ , por lo tanto $M'_\xi(t+1)$ puede tener aún otro incremento⁷⁶.

Del resultado del teorema anterior se deduce que la proporción de instancias de un esquema ξ se incrementará mientras

$$[1 - P_C(l(\xi) / (l - 1))(1 - P(\xi, t))] [* \mu_\xi(t) / \mu(t)] \geq 1$$

Cuando el operador de cruce deja de aplicarse de manera aleatoria se tiene que

$$[1 - (l(\xi) / (l - 1))(1 - P(\xi, t))] [* \mu_\xi(t) / \mu(t)] \geq 1$$

Si además $P(\xi, t)$ es muy pequeña, un esquema tiene una gran probabilidad de incrementar su rango si

$$* \mu_\xi(t) \geq \mu(t) / [1 - (l(\xi) / (l - 1))]$$

En el caso de los esquemas con longitud pequeña, el factor de pérdida

$$1 - (l(\xi) / (l - 1))$$

será menor, por lo tanto, cualquier esquema de longitud pequeña necesita mostrar un desempeño sólo un poco mayor al de la población para incrementar su rango, sin embargo, otro con longitud mayor debe ser muy apto para tener oportunidad de sobrevivir.

El cruce permite que la asignación de rangos a los esquemas se realice de manera automática y eficiente. El teorema anterior muestra claramente el paralelismo intrínseco de los planes reproductivos, no es necesario realizar alguna operación adicional para controlar el gran flujo de información que representan los esquemas.

Los planes de adaptación reproductivos exploran una gran variedad de regiones al incorporar características de otras estructuras de acuerdo al desempeño de éstas, el operador de cruce permite que el plan pueda escapar a óptimos locales al realizar la búsqueda en varios puntos al mismo tiempo.

Como los planes del tipo **R** miden el desempeño de los esquemas en relación con el promedio actual de la población conforme las generaciones avanzan, los esquemas presentes están compuestos de los atributos de otros que han tenido un rango alto, como resultado se obtiene que los esquemas asociados con óptimos locales van perdiendo rango conforme resultados superiores son encontrados, logrando una búsqueda eficiente sobre las regiones de **a**.

⁷⁶ op. cit., p. 103.

Será conveniente, antes de continuar con la exposición de otros operadores, reafirmar el funcionamiento de los planes reproductivos al incorporar el operador de cruce con un ejemplo.

Asumamos que tenemos $M = 5$ elementos (representaciones binarias de números enteros como lo hemos venido manejando)

$$A_1 = 000000$$

$$A_2 = 110000$$

$$A_3 = 011000$$

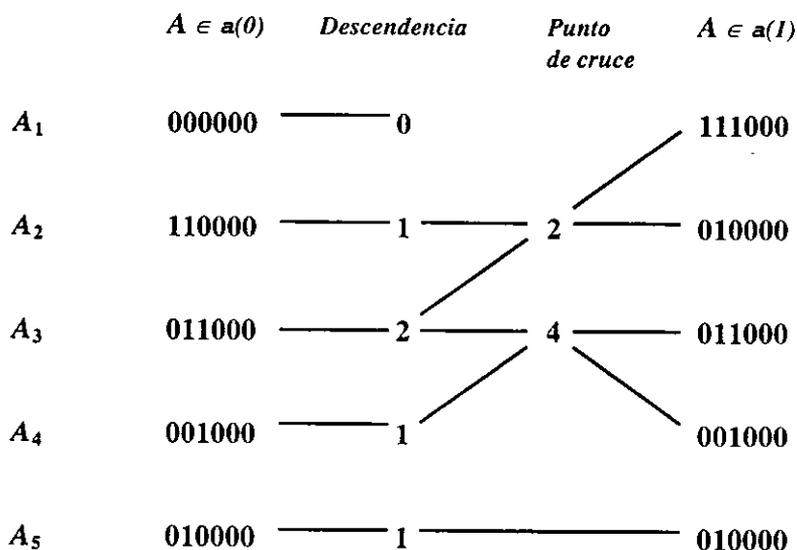
$$A_4 = 001000$$

$$A_5 = 010000$$

que han sido seleccionados aleatoriamente para formar $\mathbf{a}(0)$.

Tenemos que $\mu_1 = 0$, $\mu_2 = 3$, $\mu_3 = 6$, $\mu_4 = 4$ y $\mu_5 = 2$.

De acuerdo a esto las descendencias esperadas serían, tomando en cuenta que el promedio poblacional es 3, para $A_1 \cong 0$, $A_2 \cong 3/3$, $A_3 \cong 6/3$, $A_4 \cong 4/3$, $A_5 \cong 2/3$.



El cruce que se realiza sobre A_2 con uno de los descendientes de A_3 en el punto de intersección 2, genera una nueva instancia de 111000 y genera la primera instancia del esquema 111000, además la nueva estructura 111000 tiene la mejor aptitud, el plan espera, a través de su estrategia, mejorar continuamente el promedio del desempeño poblacional.

5.3 OPERADORES GENÉTICOS – INVERSIÓN

Es posible que los esquemas que tienen como instancias a las mejores estructuras tengan una longitud grande, esto implica que el operador de cruce continuamente los esté afectando, para resolver este problema se incluye el operador de inversión.

Aunque en la teoría este operador corrige pérdidas ocasionadas por el cruce, implementarlo no es sencillo (computacionalmente hablando), por esta razón su uso en las aplicaciones no es común, sin embargo, es útil incluirlo en la exposición para destacar otros aspectos importantes de los planes.

En el desarrollo de los planes hemos considerado al conjunto \mathbf{a} como cualquier dominio que pueda ser expresado como concatenaciones de atributos y también hemos establecido la funcionalidad de un atributo por medio de la posición que ocupa la estructura.

Esto implica que un buen atributo está ligado con su posición, entonces, al colocarlo en otra puede perder su funcionalidad, para modificar la liga entre los atributos se utiliza el operador de inversión.

El objetivo de modificar la liga de los atributos eficientes es acercarlos cuando se encuentran en posiciones lejanas entre sí, siendo entonces menos propensos a ser afectados por el cruce.

Como ya se mencionó, el sentido funcional de los atributos no debe perderse al colocarlo en otra posición, la forma más sencilla de introducir este cambio en la representación actual consiste en asignar a cada atributo un índice indicando la posición a la que está asociado, cada atributo será representado por un par (i, a) indicando que $a = \delta_i(A)$.

Bajo esta representación toda estructura A puede ser representada como cualquier permutación de

$$((1, \delta_1(A)), (2, \delta_2(A)), (3, \delta_3(A)), \dots, (l, \delta_l(A))) \quad ^{77}$$

la extensión a los esquemas es similar.

Supongamos que tenemos, en la representación inicial, la estructura

$$(011)$$

bajo la nueva representación, la estructura puede ser cualquier permutación de sus atributos

⁷⁷ op. cit., p. 106.

$((1,0),(2,1),(3,1))$
 $((1,0),(3,1),(2,1))$
 $((2,1),(1,0),(3,1))$
 $((2,1),(3,1),(1,0))$
 $((3,1),(1,0),(2,1))$
 $((3,1),(2,1),(1,0))$

algunos esquemas que tienen como instancia a esta estructura son

$\{S(2,1)(3,1)\}$
 $\{(2,1)S(3,1)\}$

donde el segundo esquema tiene mayor longitud que el primero.

Para definir este nuevo conjunto de representaciones V_i será definido nuevamente como $V'_i = (i, v)$ para toda $v \in V_i$ ⁷⁸.

Sea A^\dagger el conjunto de todas las permutaciones de la l -tupla A , entonces

$$\mathbf{a}^\dagger = \left(\prod_{i=1}^l V'_i \right)^\dagger$$
⁷⁹

y el conjunto de esquemas asociados será

$$\Xi = \left(\prod_{i=1}^l \{ V'_i \cup \{ S \} \} \right)^\dagger$$
⁸⁰

El objetivo del operador de inversión debe ser entonces producir esquemas de menor longitud, es decir, se debe reemplazar a $\xi \in \Xi^\dagger$ por alguna de sus permutaciones $\xi' \in \Xi^\dagger$ tal que $l(\xi') < l(\xi)$.

El operador de inversión se define por medio del siguiente procedimiento⁸¹

1. Una estructura $A = a_1, a_2, \dots, a_l$ es seleccionada aleatoriamente de la población actual $\mathbf{a}(t)$, donde cada a_j ahora representa un par $(j, v) \in V'_j$.

⁷⁸ op. cit., p. 107.

⁷⁹ ibídem.

⁸⁰ ibídem.

⁸¹ ibídem.

2. Dos números x'_1 y x'_2 son seleccionados aleatoriamente de $\{0, 1, 2, \dots, l + 1\}$ y se utilizan para definir $x_1 = \min \{x'_1, x'_2\}$ y $x_2 = \max \{x'_1, x'_2\}$.
3. Una nueva estructura es formada a partir de A invirtiendo el segmento a la derecha de la posición x_1 y a la izquierda de x_2 , después de esta operación se tiene $a_1, \dots, a_{x_1}, a_{x_2-1}, a_{x_2}, \dots, a_{x_1+1}, a_{x_2}, \dots, a_l$

Con este procedimiento la inversión puede ligar alelos que se encontraban separados entre sí.

El efecto de la inversión sobre las instancias de un esquema ξ es la producción de permutaciones ξ' con diferentes longitudes. Aunque el operador altera la liga entre los atributos de los esquemas, no altera los subconjuntos de a que designan.

Recordemos que el teorema 5.2.3 hace referencia al factor de pérdida

$$1 - (l(\xi) / (l - 1))$$

si la inversión produce una permutación de menor longitud, entonces este factor también será menor y el cambio en la proporción de un esquema de una generación a otra será mayor.

El operador de inversión muestra también la cualidad del paralelismo intrínseco al afectar la longitud de diferentes esquemas al mismo tiempo cuando opera sobre una estructura.

Aun cuando el operador de inversión es, en un sentido "idealista" eficiente, esta exposición no pretende incluir aspectos demasiado complicados en una primera etapa del aprendizaje de los planes como herramienta para resolver problemas de optimización, además, existen dos puntos importantes que debemos aclarar para poder omitir al operador de inversión

1. Conforme la búsqueda del plan avanza, se espera que los esquemas con mayor rango también tengan un mayor número de posiciones definidas, entonces, en generaciones avanzadas la inversión deja de jugar un papel importante.
2. Implementar la inversión en una aplicación puede llevar a una baja en el desempeño de la complejidad temporal del plan.

Dado lo anterior la inversión no será incluida en la formulación final de los planes de adaptación.

5.4 OPERADORES GENÉTICOS – MUTACIÓN

En genética la mutación es un proceso en el que un alelo es reemplazado por otro para formar una nueva estructura, generalmente existe una probabilidad pequeña de mutación en cada gen dentro de una estructura.

El operador de mutación procede de la siguiente manera⁸²

1. Las posiciones x_1, x_2, \dots, x_h , que se verán afectadas por la mutación son determinadas de manera aleatoria, donde cada posición tiene una probabilidad pequeña de verse afectada y esta probabilidad es independiente de lo que les suceda a otra posiciones.
2. Una nueva estructura $A' = a_1, a_2, \dots, a_{x_1-1}, a'_{x_1}, a_{x_1+1}, \dots, a_{x_2-1}, a'_{x_2}, a_{x_2+1}, \dots, a'_h$ es formada, donde a'_{x_i} es tomado aleatoriamente del conjunto de atributos V_i asociado a la posición x_i , teniendo cada elemento en V_i la misma probabilidad de ser seleccionado, el proceso es el mismo para cualquier otra posición.

El uso de la mutación en los planes es esencial. Dada la forma en que los planes funcionan, las estructuras que tienen un desempeño bajo probablemente serán eliminadas en alguna generación, si estas estructuras contenían atributos que en otras provocarían mejores desempeños, al ser eliminadas (y si dichos atributos no se encontraban en otros individuos) el plan no podrá incorporar de nuevo esas características.

El plan, al eliminar un atributo, por medio del borrado de una estructura, no puede volver a incorporarlo porque utiliza un conjunto inicial de individuos y crea su búsqueda a partir de ellos, si un atributo desaparece de la población en alguna generación no volverá a aparecer después.

Visto como un problema de optimización, el plan puede “quedarse” en un óptimo local si el óptimo global tiene, en su representación estructural, al atributo que fue eliminado.

Este problema se resuelve con la incorporación de la mutación en los planes, la mutación es una especie de “póliza de garantía” que asegura que ningún atributo desaparecerá permanentemente de la población.

⁸² op. cit., p. 109.

Si la probabilidad de mutación en cada posición es igual a 1P_M entonces un esquema definido en $l(\xi)$ posiciones puede ser afectado al menos por una mutación con probabilidad

$$1 - (1 - {}^1P_M)^{l(\xi)}$$

Al incorporar la mutación en el teorema 5.2.3 se obtiene

Corolario 5.4.1: Considérese un plan reproductivo del tipo **R** usando el operador de cruce y mutación, la proporción esperada de cada esquema representado en $\mathbf{a}(t)$ cambia a la generación siguiente de $P(\xi, t)$ a

$$P(\xi, t+1) \geq [1 - P_c(l(\xi)/(l-1))(1 - P(\xi, t))] (1 - {}^1P_M)^{l(\xi)} (*\mu_\xi(t)/\mu(t)) P(\xi, t) \quad 83$$

Prueba: La mutación se interpreta como una fuente de pérdida para "un" esquema, es decir, éste puede perder rango si alguna de sus posiciones definidas se muta, aunque otro esquema ganará rango al ganar otra instancia.

Entonces, al factor de pérdida

$$[1 - (1 - P(\xi, t)) P_c(l(\xi)/(l-1))]$$

ahora se añade

$$(1 - {}^1P_M)^{l(\xi)}$$

como la probabilidad de que el esquema no sea afectado en sus posiciones definidas, esto lleva a

$$P(\xi, t+1) \geq [1 - P_c(l(\xi)/(l-1))(1 - P(\xi, t))] [(1 - {}^1P_M)^{l(\xi)}] [*\mu_\xi(t)/\mu(t)] P(\xi, t)$$

⁸³ op. cit., p. 110.

El cruce y la mutación son hoy en día los operadores más utilizados, los estudios actuales se desarrollan utilizando a éstos como base pues a través de la experimentación han demostrado ser los mejores.

Existen otras versiones de estos operadores, aunque en general no superan el desempeño de los originales.

Holland propone en [1] otro operador conocido como "operador dominante", consiste en que los de atributos asociados a cada posición tengan uno "recesivo" y otro "dominante", es decir, una estructura sería ahora "bidimensional".

Esto tiene como objetivo evitar que atributos con desempeño bajo sean continuamente evaluados y permanezcan por un tiempo prolongado en la población.

Sin embargo, incluirlo requiere también de un "mapeo" que determine el atributo que estará presente en la estructura al momento de ser evaluada y modificada.

La mutación corrige este problema al incorporar de nuevo esos atributos perdidos aunque quizás con menor frecuencia, de cualquier forma, incluir al operador dominante implica mayor complejidad temporal y más recursos de cómputo, lo que provoca que no sea utilizado en las aplicaciones actuales.

Conforme ha avanzado el estudio de los algoritmos genéticos se han desarrollado nuevas formas de selección de estructuras, Goldberg en [2] da una breve explicación de las más importantes. Estos diferentes procedimientos han dado origen a lo que se conoce como *operador de selección*, el método que se expuso es conocido como *selección proporcional a la aptitud* (paso 6 del algoritmo 5.1.1), y simplemente consiste en la asignación de una probabilidad a cada estructura proporcional a su aptitud, dada esta asignación el plan realiza un experimento aleatorio al seleccionar una estructura, el espacio muestral del experimento es el conjunto de estructuras, es claro que estructuras con mejor desempeño tienen mayor probabilidad de ser tomadas por el plan.

Holland en [1] no describe a la selección de las estructuras como un operador porque la selección no las altera, es una guía para que el plan logre una búsqueda eficiente.

6. EFICIENCIA DE LOS PLANES DE ADAPTACIÓN

6.1 PLANES DEL TIPO R_1 ($P_C, {}^1P_M$)

En este capítulo se hará la formulación final de un plan de adaptación del tipo R_1 ⁸⁴, como se verá, los planes simulan un proceso de adaptación al incorporar mejores individuos y combinar las características de éstos utilizando los operadores genéticos.

La eficiencia de los planes se encuentra basada en la forma en que guían su búsqueda utilizando los impulsos que obtienen del medio al que están confrontando al evaluar a las estructuras en él.

Los planes de adaptación tienen la ventaja de no necesitar información adicional al desempeño de las estructuras para realizar su búsqueda.

Los parámetros básicos que utiliza el plan de adaptación son:

P_C , la probabilidad constante de la aplicación del operador de cruce a la estructura seleccionada por el plan

1P_M , la probabilidad de mutación de algún atributo de una estructura seleccionada

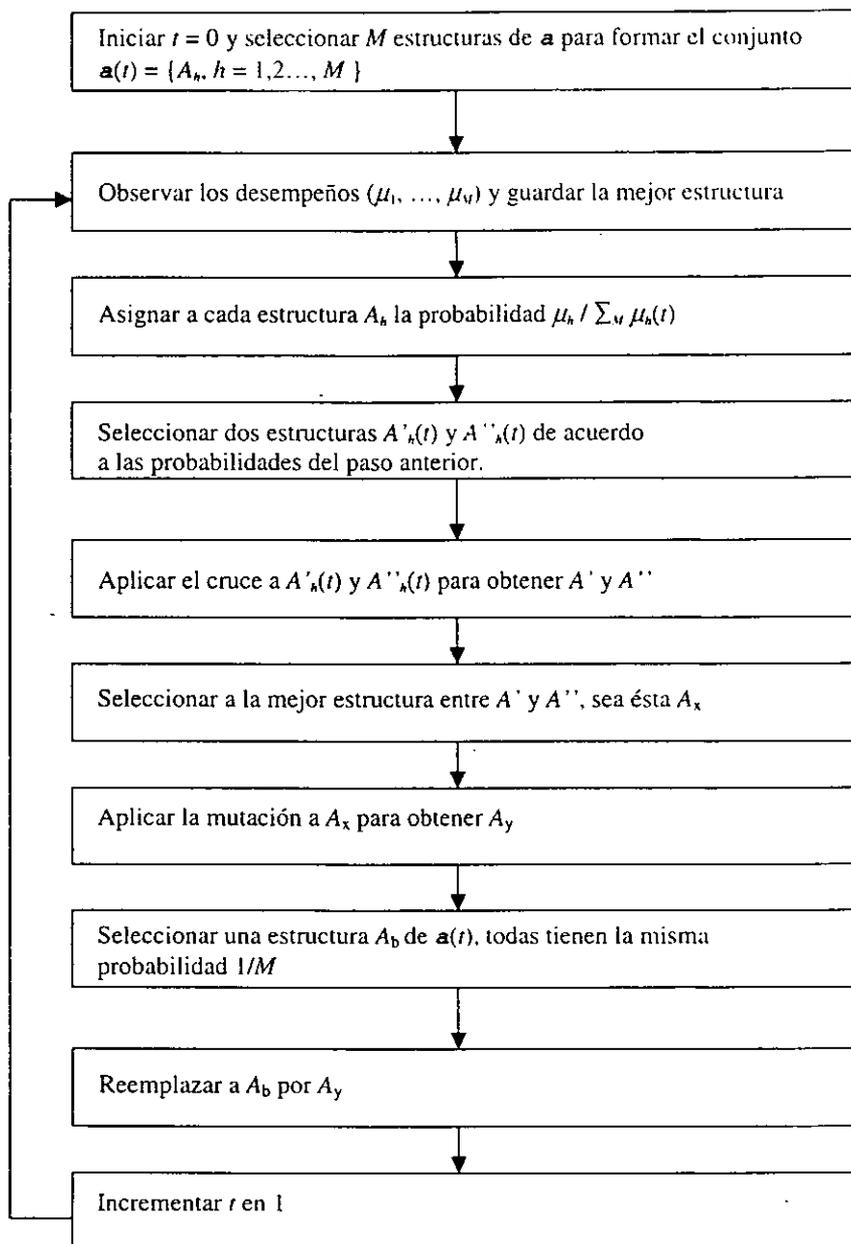
En general la probabilidad de mutación es constante, algunas aplicaciones varían la tasa conforme el plan avanza, es decir, si se desea que el plan muestre un comportamiento totalmente "convergente" es necesario que conforme las generaciones avancen la probabilidad de mutar un atributo sea menor con el fin de evitar pérdidas en el rango de los esquemas.

Como los planes son utilizados en problemas en los que para obtener la solución óptima está involucrada una gran cantidad de tiempo, los algoritmos que los describen no tienen algún paso o criterio de término.

Generalmente se habla de convergencia en la población, cuando una misma estructura ocupa una proporción predeterminada en cierta generación.

El siguiente algoritmo describe totalmente la forma en que el plan debe actuar para desarrollar su búsqueda.

⁸⁴ Esta es una versión modificada de los planes R_1 originalmente propuestos por Holland en [1].

Algoritmo 6.1.1

Estos algoritmos serán designados como

$$R_i (P_G, P_M)$$

Analicemos con detalle la funcionalidad del algoritmo.

Al asignar una probabilidad de selección a cada estructura directamente proporcional a su aptitud, el plan favorece a aquellos individuos que han mostrado un desempeño alto.

Sin embargo, no descarta a estructuras poco aptas, pues otra de las características de los planes de adaptación reproductivos, con respecto a su eficiencia, es mantener una población "diversa" al menos al inicio de su aplicación.

Si en las primeras generaciones el plan posee poblaciones con variadas formas estructurales (diferentes individuos), su oportunidad de alcanzar una mejor aproximación al óptimo global es mayor, pues explora varias secciones del espacio de búsqueda en un número pequeño de aplicaciones.

La selección de las estructuras se hace con reemplazo, es decir, la misma estructura puede ser seleccionada dos veces pues al tomar la primera muestra, ésta se devuelve a la población.

La recombinación de atributos producto de la operación del cruce, provoca que el plan tenga diferentes oportunidades de exploración y mejoras en los desempeños de los individuos.

Al seleccionar la mejor estructura producto del cruce, el plan mantiene su hipótesis de incorporar individuos más aptos.

La mutación ejerce su virtud de "póliza de garantía", al asegurar que no habrá atributo que desaparezca permanentemente de la población.

Para mantener la diversidad, el plan elimina una estructura seleccionada aleatoriamente al reemplazarla por la nueva creada, si el plan no realizara esta selección de manera uniforme sobre los individuos, podría incurrir en una especie de "convergencia prematura", ya que los mejores individuos ocuparían rápidamente fracciones importantes de la población ocasionando que queden zonas del espacio de búsqueda sin explorar.

Las probabilidades de aplicación del cruce y de la mutación pueden variar en diferentes corridas del plan para observar los efectos de estos cambios.

Generalmente la probabilidad de cruce se encuentra entre 0.5 y 0.7, la tasa de mutación es pequeña, muchos estudios ocupan 0.01 (ver [8]).

Cuando las estructuras han sido seleccionadas, la forma de determinar si se realiza la operación del cruce consiste, en simular el lanzamiento de una moneda con probabilidad de cara igual a la probabilidad de cruce, si la observación posterior al experimento es cara el cruce se aplica.

La aplicación de la mutación es análoga a la del cruce, sin embargo, ésta se realiza por cada una de las nuevas estructuras.

Realizadas las dos operaciones anteriores se selecciona a la mejor estructura.

Como se vio en el capítulo uno, la convergencia a una estructura óptima no es una guía útil para estimar la eficiencia de algún procedimiento cuando una gran cantidad de tiempo está involucrada.

Es necesario remarcar que los planes no pueden asegurar convergencia pues son procedimientos aleatorios que sufren de errores producto de estimaciones estadísticas, problemas como el que se presenta en el bandido en el capítulo cuatro.

Claro, si suficientes estructuras son probadas, la tasa de error en la que incurre un plan puede ser cada vez más pequeña.

El capítulo cuatro muestra que una guía útil para encontrar un equilibrio entre la exploración y la explotación de la información, consiste en asignar más ensayos a la opción que por medio de las observaciones se ha estimado como la mejor.

Debemos estar conscientes de que los planes son procedimientos aleatorios y por lo tanto no se puede asegurar que convergen, sin embargo en la práctica, los planes conforme avanzan las generaciones, empiezan a incrementar la proporción de un mismo individuo en la población, esto tiene una explicación sencilla.

La selección proporcional a la aptitud y la pérdida de diversidad provocan que los mejores individuos reemplacen a otros con bajo desempeño, además, el cruce tiene un efecto menor en generaciones avanzadas, pues al cruzar dos estructuras iguales en cualquier punto se obtiene como resultante la misma estructura.

Los planes R_1 pretenden seguir este criterio, al incrementar o disminuir el número de instancias de esquemas a una tasa relacionada con la observación de su desempeño $*\mu_i(t)$ en cada generación.

Lema 6.1.1: Bajo un plan del tipo $R_1 (P_C, {}^1P_M)$, dadas $M_\xi(t_0)$ instancias de ξ en la población $a(t_0)$, el número esperado de instancias de ξ al momento t , $M_\xi(t)$, es acotado inferiormente por

$$M_\xi(t) \prod_{t'=t_0}^{t-1} \varepsilon_\xi^* \mu_\xi(t') / \mu(t')$$

donde

$$\varepsilon_\xi = [1 - P_C l(\xi)/(l-1)](1 - {}^1P_M)^{l(\xi)}$$

es una constante que depende solamente de los parámetros del plan, la longitud $l(\xi)$ de ξ , y el número $l(\xi)$ de posiciones definidas de ξ ⁸⁵.

Prueba : Utilizando el corolario 5.4.1 tenemos que el cambio en la proporción de un esquema de una generación a otra está dado por:

$$P(\xi, t_0 + 1) \geq [1 - P_C (l(\xi) / (l-1))(1 - P(\xi, t_0))] (1 - {}^1P_M)^{l(\xi)} (*\mu_\xi(t_0)/\mu(t_0)) P(\xi, t_0)$$

Sean

$$C = P_C (l(\xi) / (l-1))$$

$$P = (*\mu_\xi(t_0)/\mu(t_0))P(\xi, t_0)$$

entonces

$$P(\xi, t_0 + 1) \geq [1 - P_C (l(\xi) / (l-1))(1 - P(\xi, t_0))] (1 - {}^1P_M)^{l(\xi)} (*\mu_\xi(t_0)/\mu(t_0))P(\xi, t_0)$$

$$= [1 - C (1 - P(\xi, t_0))] (1 - {}^1P_M)^{l(\xi)} P$$

$$= [1 - C + CP(\xi, t_0)](1 - {}^1P_M)^{l(\xi)} P$$

$$\geq (1 - C)(1 - {}^1P_M)^{l(\xi)} P$$

$$\geq [1 - P_C (l(\xi) / (l-1))] (1 - {}^1P_M)^{l(\xi)} (*\mu_\xi(t_0)/\mu(t_0))P(\xi, t_0)$$

⁸⁵ Variación del lema propuesto por Holland, op. cit., p. 125.

De la desigualdad anterior notamos que se puede establecer una razón de crecimiento (esperada) asociada al desempeño de los esquemas en función de los parámetros del plan

$$\epsilon_{\xi} = [1 - P_C(l(\xi) / (l - 1))] (1 - {}^1P_M)^{l(\xi)}$$

este valor es constante, entonces, para determinar como afecta sobre un intervalo de tiempo, es decir, en un número dado de generaciones consecutivas, sólo es necesario modificar a las instancias iniciales con la razón de crecimiento tantas generaciones como se desee avanzar, esto se logra con la iteración de

$$P(\xi, t_0 + h) \geq P(\xi, t_0) \prod_{t'=t_0}^{t_0+h-1} \epsilon_{\xi} (*\mu_{\xi}(t') / \mu(t'))$$

multiplicando ambos miembros de la desigualdad por el número total de estructuras en la población obtenemos

$$M P(\xi, t_0 + h) \geq M P(\xi, t_0) \prod_{t'=t_0}^{t_0+h-1} [\epsilon_{\xi} (*\mu_{\xi}(t') / \mu(t'))]$$

$$M_{\xi}(t_0 + h) \geq M_{\xi}(t_0) \prod_{t'=t_0}^{t_0+h-1} [\epsilon_{\xi} (*\mu_{\xi}(t') / \mu(t'))]$$

El lema 6.1.1 marca un punto muy importante, aún cuando los planes del tipo $\mathbf{R}_1(P_C, {}^1P_M)$ tratan sólo una estructura de $\mathbf{a}(t)$ a la vez, los esquemas son realmente quienes están siendo probados y a los que se les asigna rango de acuerdo al desempeño de sus instancias.

Como se mencionó antes, en una población de tamaño M , existe un número de esquemas entre (2^l) y $(M2^l - M + 1)$ dependiendo de la variedad de estructuras que se encuentren en ese momento en ella, es decir, la diversidad de la población. Se espera que conforme la aplicación del plan avanza, este número se acerque a 2^l , pues en generaciones avanzadas los esquemas estarán más definidos y las estructuras más aptas empezarán a predominar.

Cada uno de los esquemas en $a(t)$ cambia su proporción, de una generación a otra, a una tasa en gran parte determinada por su desempeño observado $*\mu_{\xi}(t)$, este cambio puede verse afectado por lo que le ocurra a otros esquemas, sin embargo, esta afectación no determina de manera principal el nuevo rango, es un factor de pérdida o ganancia que es provocado al evolucionar las estructuras.

Es importante destacar que, conforme las generaciones avanzan, se espera que los esquemas con instancias aptas tengan una proporción cada vez mayor, entonces si

$$P(\xi, t) \rightarrow 1$$

el factor de pérdida por el cruce

$$P_c(l(\xi) / (l-1))(1 - P(\xi, t)) \rightarrow 0$$

y

$$[1 - P_c(l(\xi) / (l-1))(1 - P(\xi, t_0))] \rightarrow 1$$

lo que implica que la tasa global de cambio estará cercana a

$$(1 - P_M)^{l(\xi)} (*\mu_{\xi}(t_0) / \mu(t_0)) - 1$$

además, si $M_{\xi}(t)$ es bastante grande, $*\mu_{\xi}(t)$ se aproximará rápidamente al verdadero desempeño promedio de las instancias del esquema $\mu_{\xi}(t)$.

Si la tasa de mutación se reduce conforme avanza la aplicación del plan, en consecuencia si

$$P_M \rightarrow 0$$

entonces

$$[(1 - P_M)^{l(\xi)}] \rightarrow 1$$

y la tasa global de cambio podría ser aproximada por

$$[\mu_{\xi}(t) / \mu(t)] - 1$$

De aquí podemos establecer que el desempeño que tiene una estructura, es la guía que el plan de adaptación utiliza, y es ésta la que permite realizar una búsqueda "inteligente" en problemas difíciles de aproximar, problemas como los *NP*.

6.2 APLICACIÓN DE LA ASIGNACIÓN ÓPTIMA DE ENSAYOS A LOS PLANES REPRODUCTIVOS

En esta sección (basada totalmente en [1] pp. 128-129) se determinará el número de ensayos asignados al mejor esquema observado como una función del número de ensayos asignados a estructuras que no son instancias de ξ , confirmaremos que la estrategia del problema del bandido es similar a la del plan de adaptación reproductivo.

$n_{\xi, n}(t_0)$ designará el número de estructuras en $\mathbf{a}(t_0)$ que no son instancias del esquema ξ .

$N_{\xi, n}$ y $n_{\xi, n}$ designarán el número de ensayos asignados de t_0 a t a estructuras que son instancias de ξ y que no lo son respectivamente.

Esto es

$$N_{\xi, n} + n_{\xi, n} = (t - t_0 + 1) M$$

para $t \geq t_0$.

Lema 6.2.1: Si cada instancia de ξ tiene un aumento en promedio, por lo menos de una nueva instancia de ξ en cada generación sobre el intervalo (t_0, t) , en consecuencia si

$$N_{\xi, n} \geq (t - t_0 + 1) M_{\xi}(t_0 - 1)$$

entonces el número de ensayos partir de t_0 a t satisface

$$N_{\xi, n} \geq M_{\xi}(t_0 - 1) \exp \left[\sum_{t_0}^t (n_{\xi, n} / n_{\xi, n}(t_0)) \right]$$

donde

$$\sum_{t_0}^t = (1/(t - t_0 + 1)) \sum_{t'=t_0}^t \ln [\epsilon_{\xi}(*\mu_{\xi}(t' - 1) / \mu(t' - 1))]$$

es una cota inferior para el promedio del logaritmo del desempeño sobre (t_0, t) .

Prueba :

$$N_{\xi, n} = \sum_{t'=t_0}^t M_{\xi}(t')$$

$$> M_{\xi}(t)$$

por el lema 6.1.1 obtenemos

$$\begin{aligned} M_{\xi}(t) &\geq M_{\xi}(t_0 - 1) \prod_{t'=t_0}^t [\varepsilon_{\xi}(*\mu_{\xi}(t' - 1) / \mu(t' - 1))] \\ &= M_{\xi}(t_0 - 1) \exp \left[\ln \prod_{t'=t_0}^t [\varepsilon_{\xi}(*\mu_{\xi}(t' - 1) / \mu(t' - 1))] \right] \\ &= M_{\xi}(t_0 - 1) \exp \left[\sum_{t'=t_0}^t \ln [\varepsilon_{\xi}(*\mu_{\xi}(t' - 1) / \mu(t' - 1))] \right] \\ &= M_{\xi}(t_0 - 1) \exp \left[\sum_{t_0}^t (t - t_0 + 1) \right] \end{aligned}$$

sin embargo, por definición

$$n_{\xi, n} / n_{\xi, n}(t_0) = (M(t - t_0 + 1) - N_{\xi, n}) / (M - M_{\xi}(t_0))$$

por la premisa del lema

$$n_{\xi, n} / n_{\xi, n}(t_0) \leq (M(t - t_0 + 1) - (t - t_0 + 1) M_{\xi}(t_0)) / (M - M_{\xi}(t_0))$$

$$n_{\xi, n} / n_{\xi, n}(t_0) \leq (M - M_{\xi}(t_0))(t - t_0 + 1) / (M - M_{\xi}(t_0))$$

se obtiene finalmente

$$n_{\xi, n} / n_{\xi, n}(t_0) \leq t - t_0 + 1$$

ahora sustituyendo $t - t_0 + 1$ obtenemos

$$N_{\xi, n} \geq M_{\xi}(t_0 - 1) \exp \left[\sum_{t_0}^t (n_{\xi, n} / n_{\xi, n}(t_0)) \right]$$

El objetivo de este lema consiste en mostrar como las instancias de esquemas que poseen buena aptitud, se ven favorecidas en su asignación de ensayos de manera similar a aquella propuesta por el problema del bandido.

Efectivamente, los planes de adaptación reproductivos dan mayor oportunidad de evolución a estructuras que poseen aptitud alta.

Es con este lema como termina la exposición de los planes de adaptación, sin embargo, creo necesario incluir en una última sección una pequeña implementación de un plan de adaptación para que se tenga una mejor visión de lo importantes y eficientes que son estas herramientas para fines de optimización.

6.3 UNA IMPLEMENTACIÓN SIMPLE DE UN PLAN DE ADAPTACIÓN

Esta última sección pretende mostrar como se realiza la implementación de un plan de adaptación reproductivo $R_1(P_C, P_M)$ para la optimización de una función.

Antes de comenzar debo aclarar que la función a optimizar no presenta las dificultades que suponemos para la utilización de esta herramienta, pretendo que esta pequeña demostración sea de carácter ilustrativo y pedagógico.

Sin embargo, aún cuando la función que utilizaremos es sencilla, se debe destacar el mérito del plan al no utilizar algún elemento adicional a las propias estructuras y su información en el medio ambiente o sistema para llevar a cabo la adaptación.

El programa `planR1.c` está desarrollado en el lenguaje de programación C, el código se encuentra en el apéndice y puede ser modificado para diferentes tamaños de población, longitud de estructuras, probabilidades de aplicación de operadores, diferentes funciones de desempeño u objetivo y el número de generaciones que se desee.

El plan `planR1.c` tiene como objetivo maximizar la función

$$f(x) = (x)^{1/2}$$

para los valores de $x = 0, 1, 2, 3, \dots, 1023$.

Hemos venido hablando acerca de las estructuras como composiciones de atributos, para este caso es fácil notar que cualquiera de los números $0, 1, 2, \dots, 1023$ puede ser expresado en forma binaria por alguna cadena o estructura de longitud 10.

Para este caso el conjunto de atributos en cada posición es $V_i = \{0, 1\}$, $i = 1, 2, \dots, 10$, entonces el conjunto total de estructuras está dado por

$$\mathbf{a} = V_1 \times V_2 \times V_3 \times V_4 \times V_5 \times V_6 \times V_7 \times V_8 \times V_9 \times V_{10}$$

Las estructuras 0000000000 y 1111111111 son los extremos de los valores de x , es decir, 0 y 1023. La fórmula de equivalencia entre representaciones binarias y decimales puede ser encontrada en el capítulo uno sección tres.

La función de desempeño será

$$\mu = f(x) = (x)^{1/2}$$

representando a cada valor en su forma binaria correspondiente.

Es claro que el óptimo de la función, para los valores de x propuestos, se encuentra en el punto $x = 1023$, sin embargo, el plan no conoce este hecho, más aun, el plan sólo tiene a las estructuras junto con su desempeño para crear otras mejores y así encontrar el óptimo.

Como primer paso el programa establece los parámetros básicos del plan: el tamaño de la población, la longitud de las estructuras (número de atributos), la probabilidad de aplicación de los operadores de cruce y mutación, genera un arreglo de números aleatorios que utilizará durante su ejecución y el número de generaciones o iteraciones del plan.

Dados estos parámetros el programa genera aleatoriamente los elementos que estarán presentes en la primera generación utilizando la función *poblacion_inicial*().

El programa asigna la aptitud correspondiente a cada estructura al evaluarla en la función de desempeño, registrando cual es la mejor a lo largo de la historia de la aplicación del plan. Este procedimiento es realizado en la función *funcion_objetivo*().

Después del punto anterior el programa está listo para comenzar el ciclo, es decir, la operación detallada del plan durante el número de generaciones que se haya determinado.

El programa muestra los elementos de la población al inicio de la generación por medio de la función *reporte*().

La selección de las estructuras se realiza de manera aleatoria, recordemos que cada estructura tiene una probabilidad proporcional a su desempeño, esta selección es realizada en la función *seleccion*().

Con las estructuras seleccionadas, el siguiente paso consiste en realizar la operación de cruce sobre éstas para obtener dos nuevos individuos. En la función *cruce*() se realiza este procedimiento.

A las estructuras resultantes del cruce se les aplica la mutación con la función *mutacion()*.

La mejor estructura resultante del proceso comprendido por los dos pasos anteriores reemplaza a alguna de la población actual, con este reemplazo queda conformada la nueva población para la siguiente generación, esto se realiza en *reemplazo()*.

La aptitud de las estructuras es de nuevo calculada y se despliega la información de éstas al final de la generación. El ciclo comienza de nuevo.

Para determinar la aplicación de los operadores se utiliza la función *flip(prob)*, que simula el lanzamiento de una moneda con probabilidad de mostrar cara *prob*. Si después del lanzamiento se obtiene cara el operador en cuestión es aplicado.

Para determinar la población inicial del plan, se utiliza la función *flip(prob)*, en este caso se asigna la misma probabilidad a cara que a cruz, si el resultado es cara el valor del atributo en la posición que se esté definiendo será 1 de lo contrario será 0.

Cuando los elementos son seleccionados, el programa despliega cuáles fueron éstos junto con las modificaciones producto de la aplicación de los operadores, también se despliega el mejor elemento creado y se indica la estructura a la que reemplazará.

La corrida que se realizó de este programa, consistió de 100 generaciones, los resultados que obtuvo el plan en algunas generaciones son los siguientes

Elementos al inicio de la generacion 0

Estructura	Valor	Aptitud
0101110111	954	30.886890
1011111000	125	11.180340
1101001101	715	26.739485
1001001111	969	31.128765
1101101000	91	9.539392
1110101010	343	18.520260
1000010110	417	20.420578
1000111101	753	27.440845
0000000000	0	0.000000
0010100010	276	16.613247

Los elementos seleccionados fueron 3 y 10, punto de cruce 6, nuevas estructuras

1101000010 estructura 1
0010101101 estructura 2

Punto de mutacion estructura 1, 10, nueva estructura

1101000011

Elemento 5 reemplazado por la estructura 2

Elementos al final de la generacion 0

Estructura	Valor	Aptitud
0101110111	954	30.886890
1011111000	125	11.180340
1101001101	715	26.739485
1001001111	969	31.128765
0010101101	724	26.907248
1110101010	343	18.520260
1000010110	417	20.420578
1000111101	753	27.440845
0000000000	0	0.000000
0010100010	276	16.613247

Mejor encontrado en generacion 0

1001001111	969	31.128765
------------	-----	-----------

Elementos al inicio de la generacion 20

Estructura	Valor	Aptitud
1110101001	599	24.474476
1000010111	929	30.479502
1110110111	951	30.838287
1110110111	951	30.838287
0010101101	724	26.907248
1110110111	951	30.838287
1000010110	417	20.420578
1000111101	753	27.440845
1110101101	727	26.962938
1110111111	1015	31.859064

Los elementos seleccionados fueron 9 y 5, punto de cruce 2, nuevas estructuras

1110101101 estructura 1

0010101101 estructura 2

Elemento 1 reemplazado por la estructura 1

Elementos al final de la generacion 20

Estructura	Valor	Aptitud
1110101101	727	26.962938
1000010111	929	30.479502
1110110111	951	30.838287
1110110111	951	30.838287
0010101101	724	26.907248
1110110111	951	30.838287
1000010110	417	20.420578
1000111101	753	27.440845
1110101101	727	26.962938
1110111111	1015	31.859064

Mejor encontrado en generacion 18

1110111111	1015	31.859064
------------	------	-----------

Elementos al inicio de la generacion 42

Estructura	Valor	Aptitud
0011110111	956	30.919250
1100101111	979	31.288975
1100101111	979	31.288975
1101101111	987	31.416555
1101101111	987	31.416555
1000101101	721	26.851442
1100110111	947	30.773365
1110110111	951	30.838287
1100101111	979	31.288975
1110111111	1015	31.859064

Los elementos seleccionados fueron 10 y 2,punto de cruce 9, nuevas estructuras

1110111111 estructura 1
1100101111 estructura 2

Punto de mutacion estructura 1, 4,nueva estructura
1111111111

Elementos al final de la generacion 42

Estructura	Valor	Aptitud
0011110111	956	30.919250
1111111111	1023	31.984371
1100101111	979	31.288975
1101101111	987	31.416555
1101101111	987	31.416555
1000101101	721	26.851442
1100110111	947	30.773365
1110110111	951	30.838287
1100101111	979	31.288975
1110111111	1015	31.859064

Mejor encontrado en generacion 42
1111111111 1023 31.984371

Elementos al inicio de la generacion 100

Estructura	Valor	Aptitud
1110111111	1015	31.859064
1110111111	1015	31.859064
1110111111	1015	31.859064
1110111111	1015	31.859064
1110011111	999	31.606962
1110111111	1015	31.859064
1110111111	1015	31.859064
1111111111	1023	31.984371
1110111111	1015	31.859064
1111111111	1023	31.984371

Los elementos seleccionados fueron 4 y 8,punto de cruce 1, nuevas estructuras
1111111111 estructura 1
1110111111 estructura 2

Elemento 5 reemplazado por la estructura 1

Elementos al final de la generacion 100

Estructura	Valor	Aptitud
1110111111	1015	31.859064
1110111111	1015	31.859064
1110111111	1015	31.859064
1110111111	1015	31.859064
1111111111	1023	31.984371
1110111111	1015	31.859064
1110111111	1015	31.859064
1111111111	1023	31.984371
1110111111	1015	31.859064
1111111111	1023	31.984371

Mejor encontrado en generacion 42

1111111111	1023	31.984371
------------	------	-----------

CONCLUSIONES

Los sistemas de adaptación artificial se han mostrado como una herramienta para atacar problemas de optimización con espacio de soluciones extenso de manera eficiente utilizando a los algoritmos descritos con asignación de más ensayos a aquellas estructuras con desempeño por arriba del promedio generación tras generación.

Sin embargo, la formulación de los planes que aquí se presentaron, y que llevaron después al desarrollo de los algoritmos genéticos, presentan aun grandes retos para hacerlos más eficientes y confiables.

El principal, en mi opinión, consiste en determinar exactamente cuál es la aportación de cada uno de los atributos que posee una estructura en su desempeño, esto se debe a que los planes de adaptación (algoritmos genéticos) procesan los esquemas de manera "implícita", es decir, en realidad el plan no conoce que tan bueno es un esquema.

Otro punto de investigación muy importante es la forma en que los operadores presentados incrementan o disminuyen el rango de un esquema, claro, asumiendo que el punto anterior ya fue resuelto.

A través de la experiencia se ha mostrado que los algoritmos genéticos convergen a una población que consta de una gran cantidad de replicas de una misma estructura, pero esto no se ha probado formalmente, además esta estructura que ocupa gran parte de la población puede no ser la mejor.

La codificación de las estructuras se realiza de acuerdo al problema a atacar de manera intuitiva, no existe una metodología para representar cualquier posible solución a un problema dado.

A pesar de esto los algoritmos genéticos empiezan a ser cada vez más utilizados por sus buenos resultados en aproximaciones de problemas difíciles, como los *NP*.

Esta breve exposición fue útil para el entendimiento de la base de los algoritmos genéticos y sus principales cualidades.

GLOSARIO

τ	el plan de adaptación
\mathbf{a}	el conjunto total de estructuras A disponibles para el plan
$\mathbf{a}(t)$	el conjunto de estructuras A disponibles para el plan en la generación t
I	las señales o impulsos que obtiene el plan al confrontar un medio o sistema
Ω	el conjunto de operadores que utiliza el plan
Γ	conjunto de planes factibles
\mathbf{E}	conjunto de posibles medios o sistemas
μ_E	medida de desempeño de las estructuras en el medio $E \in \mathbf{E}$
χ	criterio para comparar la eficiencia de los planes
V_i	conjunto de valores para la i -ésima posición de una estructura
ξ	esquema
Ξ	conjunto de esquemas
$\$$	símbolo comodín
$^*\mu_k$	promedio observado del desempeño de las instancias del esquema ξ

BIBLIOGRAFIA

- [1] HOLLAND, J. H. *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press, 1992.
- [2] GOLDBERG, D. E. *Genetic algorithms in search, optimization, and machine learning*. New York, Addison-Wesley, 1989.
- [3] GOLDBERG, D. E. *The race, the hurdle, and the sweet spot*. ILLiGAL Report No. 98007, 1998.
- [4] GOLDBERG, D. E.; DEB, K.; THIERENS, D. *Toward a better understanding of mixing of genetic algorithms*. ILLiGAL Report No. 92009, 1992.
- [5] BEASLEY, D.; BULL, D.; MARTIN, R. *An overview of genetic algorithm: Part 1, Fundamentals*. University Computing, 1993, Vol. 15, N-2, p. 58-69.
- [6] MICHALEWICZ, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1992.
- [7] MILLER, B. L.; GOLDBERG, D. E. *Genetic algorithms, selection schemes, and the varying effects of noise*. ILLiGAL Report No. 95009, 1995.
- [8] MITCHELL, M. *An introduction to genetic algorithms*. MIT Press, 1997.
- [9] MEYER, P. *Probabilidad y Aplicaciones Estadísticas*. Addison-Wesley, 1986.
- [10] PARZEN, E. *Teoría Moderna de Probabilidades*. Limusa, 1992.
- [11] PAPADIMITRIOU, C. *Combinatorial Optimization and Complexity*. MIT, 1992.

**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

ANEXO

```

/*Programa planR1.c                               */
/*Autor Jorge Sánchez Santiago                    */
/*Esta es una implementación sencilla del plan R1. Los parámetros */
/*pueden ser modificados según se desee. En esta versión se     */
/*utiliza una población de tamaño 10 y longitud de estructura 10. */
/*La función a optimizar es  $f(x) = \text{sqr}(x)$  */

#include <stdio.h>
#include <conio.h>
#include <conio.h>
#include "random.h"

struct estructura
{
    int    atributo[10];
    int    valor;
    float  aptitud;
};

struct mejor
{
    int    atributo[10];
    int    valor;
    float  aptitud;
    int    generacion;
};

struct mejor mejor_encontrado;
struct estructura poblacion[10];
struct estructura hijo1;
struct estructura hijo2;
int ind1, ind2;
int generacion;
int imprime;
float suma_aptitud;
int tam_poblacion;
int long_estructura;
float p_cruce;
float p_mutacion;

poblacion_inicial()
{
    int i, j;
    for(i = 0; i < tam_poblacion; i++)
    {
        for(j = 0; j < long_estructura; j++)
        {
            if (flip(0.5))
                poblacion[i].atributo[j] = 1;
            else
                poblacion[i].atributo[j] = 0;
        }
    }
}

```

```

funcion_objetivo()
{
int i, j, k;
k = 0;
suma_apitud = 0;
for(i = 0; i < tam_poblacion; i++)
{
poblacion[i].valor = 0;
poblacion[i].aptitud = 0.0;
for(j = 0; j < long_estructura; j++)
{
poblacion[i].valor += (int)
(poblacion[i].atributo[j] * pow(2.0, (double) j));
}
poblacion[i].aptitud = pow((double)poblacion[i].valor, 0.5);
if (poblacion[i].aptitud > mejor_encontrado.aptitud)
{
k = i+1;
mejor_encontrado.aptitud = poblacion[i].aptitud;
}
suma_apitud += poblacion[i].aptitud;
}
if (k > 0)
{
k--;
for(i = 0; i < long_estructura; i++)
{
mejor_encontrado.atributo[i] = poblacion[k].atributo[i];
}
mejor_encontrado.aptitud = poblacion[k].aptitud;
mejor_encontrado.valor = poblacion[k].valor;
mejor_encontrado.generacion = generacion;
}
}

seleccion()
{
float prob1, prob2, suma;
prob1 = randomperc();
prob2 = randomperc();
suma = 0.0;
for(ind1 = 0; (suma < prob1) && (ind1 < tam_poblacion); ind1++)
suma += poblacion[ind1].aptitud/suma_apitud;
suma = 0.0;
for(ind2 = 0; (suma < prob2) && (ind2 < tam_poblacion); ind2++)
suma += poblacion[ind2].aptitud/suma_apitud;
ind1--;
ind2--;
printf("\n Los elementos seleccionados fueron %d y %d", (ind1+1), (ind2+1));
}

```

```

cruce()
{
    int punto_cruce, i, j;
    punto_cruce = -1;
    if(!flip(p_cruce))
    {
        punto_cruce = rnd(0,(long_estructura - 2));
        for(i = 0; i < long_estructura; i++)
        {
            if (i <= punto_cruce)
            {
                hijo1.atributo[i] = poblacion[ind1].atributo[i];
                hijo2.atributo[i] = poblacion[ind2].atributo[i];
            }
            else
            {
                hijo1.atributo[i] = poblacion[ind2].atributo[i];
                hijo2.atributo[i] = poblacion[ind1].atributo[i];
            }
        }
    }
    else
    {
        for(i = 0; i < long_estructura; i++)
        {
            hijo1.atributo[i] = poblacion[ind1].atributo[i];
            hijo2.atributo[i] = poblacion[ind2].atributo[i];
        }
    }
    printf("punto de cruce %d, nuevas estructuras \n",(punto_cruce+1));
    for(j = 0; j < long_estructura; j++)
        printf("%d",hijo1.atributo[j]);
    printf(" estructura 1 \n");
    for(j = 0; j < long_estructura; j++)
        printf("%d",hijo2.atributo[j]);
    printf(" estructura 2 \n");
    hijo1.valor = 0.0;
    hijo2.valor = 0.0;
    hijo1.apititud = 0.0;
    hijo2.apititud = 0.0;
    for(j = 0; j < long_estructura; j++)
    {
        hijo1.valor +=
        (hijo1.atributo[j] * pow(2.0, (double) j ));
        hijo2.valor +=
        (hijo2.atributo[j] * pow(2.0, (double) j ));
    }
    hijo1.apititud = pow((double)hijo1.valor, 0.5);
    hijo2.apititud = pow((double)hijo2.valor, 0.5);
}

```

```

mutacion()
{
int punto_mutacion, i, j;
if (generacion >= 50)
    p_mutacion = 0.2;
if (generacion >= 75)
    p_mutacion = 0.1;
if(flip(p_mutacion))
{
    punto_mutacion = rnd(0 ,(long_estructura - 1));
if (hijo1.atributo[punto_mutacion] == 0)
    hijo1.atributo[punto_mutacion] = 1;
else
    hijo1.atributo[punto_mutacion] = 0;
printf("\n Punto de mutacion estructura 1, %d.", (punto_mutacion+1));
printf("nueva estructura \n");
for(j = 0; j < long_estructura; j++)
    printf("%d",hijo1.atributo[j]);
printf("\n");
}

if(flip(p_mutacion))
{
    punto_mutacion = rnd(0 ,(long_estructura - 1));
if (hijo2.atributo[punto_mutacion] == 0)
    hijo2.atributo[punto_mutacion] = 1;
else
    hijo2.atributo[punto_mutacion] = 0;
printf("\n Punto de mutacion estructura 2, %d.", (punto_mutacion+1));
printf("nueva estructura \n");
for(j = 0; j < long_estructura; j++)
    printf("%d",hijo2.atributo[j]);
printf("\n");
}
}

reemplazo()
{
int i, elemento;
elemento = rnd(0 , tam_poblacion-1);
printf("\n Elemento %d reemplazado ", (elemento+1));
if(hijo1.aptitud > hijo2.aptitud)
{
    for(i = 0; i < long_estructura; i++)
        poblacion[elemento].atributo[i] = hijo1.atributo[i];
    printf("por la estructura 1");
}
else
{
    for(i = 0; i < long_estructura; i++)
        poblacion[elemento].atributo[i] = hijo2.atributo[i];
    printf("por la estructura 2");
}
}
}

```

```

reporte()
{
    int i, j;
    printf("\n");
    for(i = 0; i < tam_poblacion; i++)
    {
        for(j = 0; j < long_estructura; j++)
            printf("%d", poblacion[i].atributo[j]);
        printf("  %d  %f\n", poblacion[i].valor, poblacion[i].aptitud);
    }
    if (imprime == 1)
    {
        printf("\n Mejor encontrado en generacion %d",
            mejor_encontrado.generacion);
        printf("\n");
        for(j = 0; j < long_estructura; j++)
            printf(" %d", mejor_encontrado.atributo[j]);
        printf("  %d  %f\n", mejor_encontrado.valor,
            mejor_encontrado.aptitud);
        imprime = 0;
    }
    else
        imprime = 1;
}

main(void)
{
    p_cruce = 0.8;
    p_mutacion = 0.3;
    tam_poblacion = 10;
    long_estructura = 10;
    randomize();
    poblacion_inicial();
    funcion_objetivo();
    for (generacion = 0; generacion <= 100; generacion++)
    {
        clrscr();
        printf("\n Elementos al inicio de la generacion %d\n", generacion);
        printf("\n Estructura  Valor  Aptitud");
        reporte();
        getch();
        seleccion();
        cruce();
        mutacion();
        reemplazo();
        funcion_objetivo();
        printf("\n Elementos al final de la generacion %d\n", generacion);
        printf("\n Estructura  Valor  Aptitud");
        reporte();
        getch();
    }
    return 0;
}

```

```

/*-----*/
/* random.h - Es una libreria para la generacion de numeros aleatorios */
/* desarrollada por R.E. Smith */
/* Una version completa de otro algoritmo genetico junto con esta libreria */
/* puede ser encontrada en http://www.aic.nrl.navy.mil/galist/src/#C */
/*-----*/

#include <math.h>

static double oldrand[55];          /* Array of 55 random numbers */
static int jrand;                  /* current random number */
static double rndx2;               /* used with random normal deviate */
static int rndcalcfalg;           /* used with random normal deviate */

advance_random()
/* Create next batch of 55 random numbers */
{
    int j1;
    double new_random;

    for(j1 = 0; j1 < 24; j1++)
    {
        new_random = oldrand[j1] - oldrand[j1+31];
        if(new_random < 0.0) new_random = new_random + 1.0;
        oldrand[j1] = new_random;
    }
    for(j1 = 24; j1 < 55; j1++)
    {
        new_random = oldrand[j1] - oldrand[j1-24];
        if(new_random < 0.0) new_random = new_random + 1.0;
        oldrand[j1] = new_random;
    }
}

int flip(prob)
/* Flip a biased coin - true if heads */
float prob;
{
    float randomperc();

    if(randomperc() <= prob)
        return(1);
    else
        return(0);
}

inirandomnormaldeviate()
/* initialization routine for randomnormaldeviate */
{
    rndcalcfalg = 1;
}

```

```

double noise(mu ,sigma)
/* normal noise with specified mean & std dev: mu & sigma */
double mu, sigma;
{
    double randomnormaldeviate();

    return((randomnormaldeviate()*sigma) + mu);
}

randomize()
/* Get seed number for random and start it up */
{
    float randomseed;
    int j1;
    for(j1=0; j1<=54; j1++)
        oldrand[j1] = 0.0;
    jrand=0;
    randomseed=0.5;
    warmup_random(randomseed);
}

double randomnormaldeviate()
/* random normal deviate after ACM algorithm 267 / Box-Muller Method */
{
    double sqrt(), log(), sin(), cos();
    float randomperc();
    double t, rndx1;
    if(rndcalcflag)
    {
        rndx1 = sqrt(- 2.0*log((double) randomperc()));
        t = 6.2831853072 * (double) randomperc();
        rndx2 = rndx1 * sin(t);
        rndcalcflag = 0;
        return(rndx1 * cos(t));
    }
    else
    {
        rndcalcflag = 1;
        return(rndx2);
    }
}

float randomperc()
/* Fetch a single random number between 0.0 and 1.0 - Subtractive Method */
/* See Knuth, D. (1969), v. 2 for details */
/* name changed from random() to avoid library conflicts on some machines*/
{
    jrand++;
    if(jrand >= 55)
    {
        jrand = 1;
        advance_random();
    }
    return((float) oldrand[jrand]);
}

```

```

int rnd(low, high)
/* Pick a random integer between low and high */
int low,high;
{
    int i;
    float randomperc();

    if(low >= high)
        i = low;
    else
    {
        i = (randomperc() * (high - low + 1)) + low;
        if(i > high) i = high;
    }
    return(i);
}

float rndreal(lo ,hi)
/* real random number between specified limits */
float lo, hi;
{
    return((randomperc() * (hi - lo)) + lo);
}

warmup_random(random_seed)
/* Get random off and running */
float random_seed;
{
    int j1, ii;
    double new_random, prev_random;

    oldrand[54] = random_seed;
    new_random = 0.000000001;
    prev_random = random_seed;
    for(j1 = 1 ; j1 <= 54; j1++)
    {
        ii = (21*j1)%54;
        oldrand[ii] = new_random;
        new_random = prev_random-new_random;
        if(new_random<0.0) new_random = new_random + 1.0;
        prev_random = oldrand[ii];
    }

    advance_random();
    advance_random();
    advance_random();

    jrand = 0;
}

```