Introducción a la Computación Evolutiva

Dr. Carlos A. Coello Coello

Departamento de Computación

CINVESTAV-IPN

Av. IPN No. 2508

Col. San Pedro Zacatenco

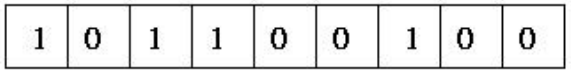
México, D.F. 07300

email: ccoello@cs.cinvestav.mx

http: //delta.cs.cinvestav.mx/~ccoello

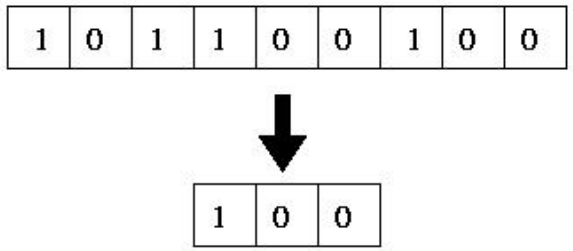
Conceptos de CE

- Denominamos **cromosoma** a una estructura de datos que contiene una cadena de parámetros de diseño o genes. Esta estructura de datos puede almacenarse, por ejemplo, como una cadena de bits o un arreglo de enteros.



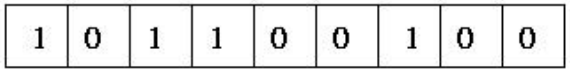
Conceptos de CE

- Se llama **gene** a una subsección de un cromosoma que (usualmente) codifica el valor de un solo parámetro.



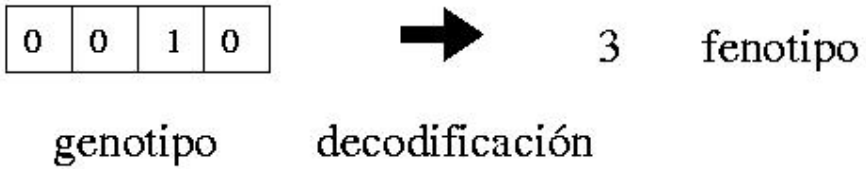
Conceptos de CE

- Se denomina **genotipo** a la codificación (por ejemplo, binaria) de los parámetros que representan una solución del problema a resolverse.



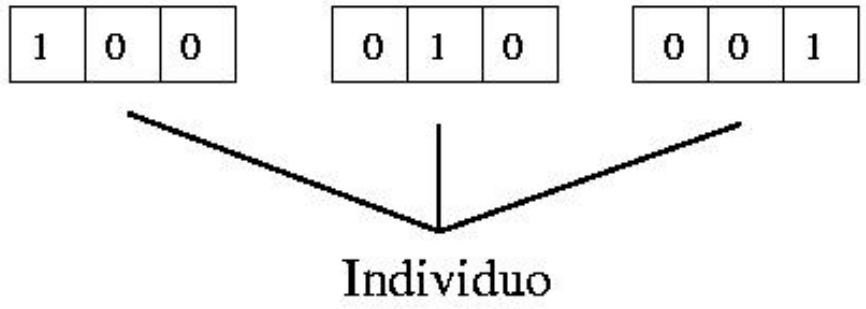
Conceptos de CE

- Se denomina **fenotipo** a la decodificación del cromosoma. Es decir, a los valores obtenidos al pasar de la representación (binaria) a la usada por la función objetivo.



Conceptos de CE

- Se denomina **individuo** a un solo miembro de la población de soluciones potenciales a un problema. Cada individuo contiene un cromosoma (o de manera más general, un genoma) que representa una solución posible al problema a resolverse.



Conceptos de CE

- Se denomina **aptitud** al valor que se asigna a cada individuo y que indica qué tan bueno es éste con respecto a los demás para la solución de un problema.

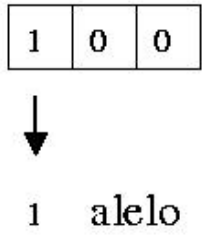
$$\text{Si } f(x) = x^2 \text{ entonces } f(1010_2) = 100$$

Conceptos de CE

- Se llama **paisaje de aptitud** (*fitness landscape*) a la hipersuperficie que se obtiene al aplicar la función de aptitud a cada punto del espacio de búsqueda.

Conceptos de CE

- Se denomina **alelo** a cada valor posible que puede adquirir una cierta posición genética. Si se usa representación binaria, un alelo puede valer 0 ó 1.



Conceptos de CE

- Llamamos **generación** a una iteración de la medida de aptitud y a la creación de una nueva población por medio de operadores de reproducción.

Conceptos de CE

- Una población puede subdividirse en grupos a los que se denomina subpoblaciones. Normalmente, sólo pueden cruzarse entre sí los individuos que pertenezcan a la misma subpoblación.

Conceptos de CE

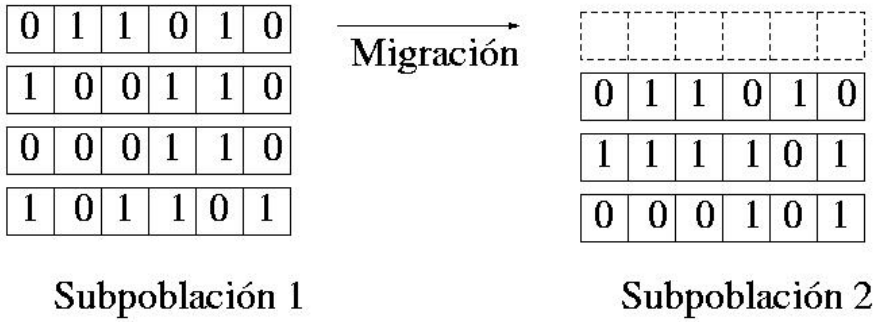
- En los esquemas con subpoblaciones, suele permitirse la migración de una subpoblación a otra (sobre todo en el contexto de AGs paralelos).

Conceptos de CE

- Al hecho de permitir la cruza sólo entre individuos de la misma subpoblación se le llama **especiación** en una emulación del fenómeno natural del mismo nombre.

Conceptos de CE

- Se llama **migración** a la transferencia de (los genes de) un individuo de una subpoblación a otra.



Conceptos de CE

- Hay un tipo de población usada en CE en la que cualquier individuo puede reproducirse con otro con una probabilidad que depende sólo de su aptitud. Se le llama **población panmítica**.

Conceptos de CE

- Lo opuesto de la población panmítica es permitir la reproducción sólo entre individuos de la misma subpoblación. La mayor parte de los AEs convencionales usan poblaciones panmíticas.

Conceptos de CE

- Debido a ruidos estocásticos, los AEs tienden a converger a una sola solución. Para evitar eso, y mantener la diversidad, existen técnicas que permiten crear distintos **nichos** para los individuos.

Conceptos de CE

- Se llama **epístasis** a la interacción entre los diferentes genes de un cromosoma. Se refiere a la medida en que la contribución de aptitud de un gene depende de los valores de los otros genes.

Conceptos de CE

- Cuando un problema tiene poca **epístasis** (o ninguna), su solución es trivial (un algoritmo escalando la colina es suficiente para resolverlo).

Conceptos de CE

- Cuando un problema tiene una **epístasis** elevada, el problema será **deceptivo**, por lo que será muy difícil de resolver por un AE.

Conceptos de CE

- Se llama **bloque constructor** a un grupo pequeño y compacto de genes que han co-evolucionado de tal forma que su introducción en cualquier cromosoma tiene una alta probabilidad de incrementar la aptitud de dicho cromosoma.

Conceptos de CE

- Se llama **decepción** a la condición donde la combinación de buenos bloques constructores llevan a una reducción de aptitud, en vez de un incremento. Este fenómeno fue sugerido originalmente por Goldberg para explicar el mal desempeño del AG en algunos problemas.

Conceptos de CE

- Se llama **operador de reproducción** a todo aquel mecanismo que influencia la forma en que se pasa la información genética de padres a hijos. Los operadores de reproducción caen en 3 amplias categorías:

  a) Cruza
  b) Mutación
  c) Reordenamiento

Conceptos de CE

- La **cruza** es un operador que forma un nuevo cromosoma combinando partes de cada uno de sus cromosomas padres.

Conceptos de CE

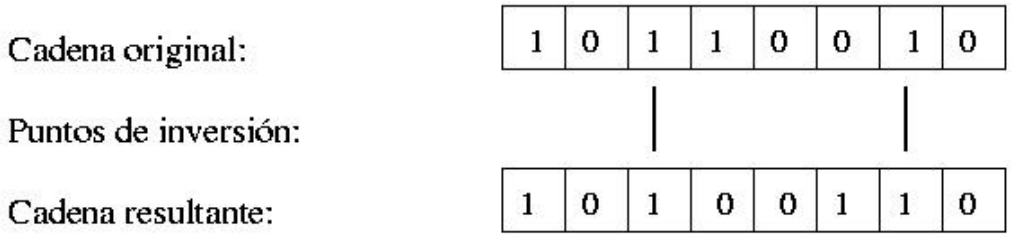
- Se denomina **mutación** a un operador que forma un nuevo cromosoma a través de alteraciones (usualmente pequeñas) de los valores de los genes de un solo cromosoma padre.

Conceptos de CE

- Un **operador de reordenamiento** es aquél que cambia el orden de los genes de un cromosoma, con la esperanza de juntar los genes que se encuentren relacionados, facilitando así la producción de bloques constructores.

Conceptos de CE

- La **inversión** es un ejemplo de un operador de reordenamiento en el que se invierte el orden de todos los genes comprendidos entre 2 puntos seleccionados al azar en el cromosoma.



Conceptos de CE

- En un AG, cuando una población no tiene **variedad requisito**, la cruza no será útil como operador de búsqueda, porque tendrá propensión a simplemente regenerar a los padres.

Conceptos de CE

- Es importante aclarar que en los AGs los operadores de reproducción actúan sobre los **genotipos** y no sobre los **fenotipos** de los individuos.

Conceptos de CE

- Se denomina **elitismo** al mecanismo utilizado en algunos AEs para asegurar que los cromosomas de los miembros más aptos de una población se pasen a la siguiente generación sin ser alterados por ningún operador genético.

Conceptos de CE

- Usar **elitismo** asegura que la aptitud máxima de la población nunca se reducirá de una generación a la siguiente. Sin embargo, no necesariamente mejora la posibilidad de localizar el óptimo global de una función.

Conceptos de CE

- Cuando se atraviesa un espacio de búsqueda, se denomina **explotación** al proceso de usar la información obtenida de los puntos visitados previamente para determinar qué lugares resulta más conveniente visitar a continuación.

Conceptos de CE

- Se denomina **exploración** al proceso de visitar regiones del espacio de búsqueda completamente nuevas, para ver si puede encontrarse algo prometedor.

Conceptos de CE

- La **exploración** involucra grandes saltos hacia lo desconocido.

  La **explotación** normalmente involucra movimientos finos.

Conceptos de CE

- La **explotación** es buena para encontrar óptimos locales.

  La **exploración** es buena para evitar quedar atrapado en óptimos locales.

Conceptos de CE

- Se denomina **esquema** a un patrón de valores de genes de un cromosoma que puede incluir estados 'no importa' (*don't care*).

Conceptos de CE

- Usando un alfabeto binario, los esquemas se forman del alfabeto $\{0, 1, \#\}$. Por ejemplo, el cromosoma 0110 es una instancia del esquema #1#0 (donde # significa 'no importa').

Técnicas de Representación

- Cadenas Binarias (Tradicional)
- Códigos de Gray (Binaria)
- Punto Flotante (Binaria)
- Punto Flotante (Real)

Técnicas de Representación

- Punto Flotante (Entera)
- Expresiones S en LISP (Programación Genética)
- Listas Binarias de Longitud Variable (*messy-GA*)
- Híbridos (AG Estructurado)

Cadenas Binarias

- Sugerida por Holland en su libro
- Además de ser una representación "universal" para cualquier tipo de alfabetos, Holland justificó el uso de representación binaria con un argumento teórico.

Cadenas Binarias

Holland favorece el uso de muchos genes con pocos alelos posibles en vez de contar con pocos genes con muchos alelos posibles.

Teóricamente, la representación binaria favorece la diversidad y la formación de buenos bloques constructores.

La representación binaria es también "natural" para las computadoras.

Cadenas Binarias

Según Holland, la representación binaria tiene también una justificación biológica: en los cromosomas naturales, es más usual que hayan muchas posiciones y pocos alelos por posición que pocas posiciones y muchos alelos por posición.

Problemas con las Cadenas Binarias

- Escalabilidad
- Epístasis
- Representación natural
- Soluciones ilegales

Códigos de Gray

Motivación: Problemas con el mapeo entre genotipo y fenotipo al usar representación binaria.

Ejemplo:
5 (decimal) = 101
6 (decimal) = 110

Diferencia de 1 en el espacio fenotípico, y de 2 en el genotípico (distancia de Hamming).

Códigos de Gray

A este fenómeno se le conoce como **risco de Hamming** (*Hamming cliff*).

Este fenómeno es el que ha propiciado propuestas de representaciones que mantengan la propiedad de adyacencia entre 2 valores consecutivos. De entre ellas, los denominados **códigos de Gray** se cuentan entre las más populares.

Códigos de Gray

Decimal	Binario	Código de Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111

Códigos de Gray

Se asume: $\mathbf{b} =< b_1, ..., b_m >$ es un número binario, y
$\mathbf{g} =< g_1, \ldots, g_m >$ es un número de Gray.

procedure Binario-a-Gray
begin
$g_1 = b_1$
for $k = 2$ **to** m **do**
$g_k = b_{k-1}$ **XOR** b_k
end

Códigos de Gray

procedure Gray-a-Binario
begin
valor = g_1
b_1 = valor
for $k = 2$ **to** m **do**
begin
if $g_k = 1$ **then** valor = **NOT** valor
b_k = valor
end
end

Representación de Punto Flotante

- Binaria
- IEEE
- Real
- Entera

Uso de Cadenas Binarias

- Podemos discretizar el rango de cada variable fijando una precisión deseable y posteriormente se tratan esos valores como si fueran enteros.

  Ejemplo:

  $$\text{-}1.5 \leq x \leq 2.0$$

  Precisión: 3 dígitos

  $$\text{Tamaño} = (\text{int})\ [log_2[(l_{sup} - l_{inf}) * 10^{\text{precisión}}] + 0.9]$$

  Tamaño = 12 bits

Uso de Cadenas Binarias

Críticas:

- Representación compacta
- Decodificación eficiente
- Problemas ante alta dimensionalidad
- Precisión limitada

Notación Estándar de la IEEE

- Podemos usar cualquier formato estándar del IEEE:

Signo	Exponente	Mantisa
0	1 0 0 0 1 0 1 1	0 1 0 0 . . . 0
1 bit	8 bits	23 bits

Notación en exceso -127

Rango numérico: 2^{-126} a 2^{126}

Notación Estándar de la IEEE

Críticas:

- Decodificación costosa
- Mapeo muy complejo entre el genotipo y el fenotipo
- Muy susceptible a cambios pequeños en algunos campos (p.ej. el exponente) y poco susceptible a cambios mayores en otros (p.ej. la mantisa)

Representación Real

- Podemos usar directamente números reales en cada gene:

1.54	-0.29	7.43	-2.15

Realmente estaríamos operando a nivel fenotípico.

Representación Real

Críticas:

- Requiere de operadores genéticos especiales
- Mutación normalmente más alta
- Los teóricos argumentan que las cardinalidades más bajas son las mejores
- La práctica parece indicar lo contrario

Representación Entera

- Podemos usar:

7	2	9	3	2	0	1

$= 72.93201$

o algo como:

72945	12902	13001	12000

Representación Entera

Críticas:

- Limitados por la precisión
- Puede requerir operadores especiales
- Compromiso entre la representación real y la binaria
- Ahorros de memoria

Programación Genética

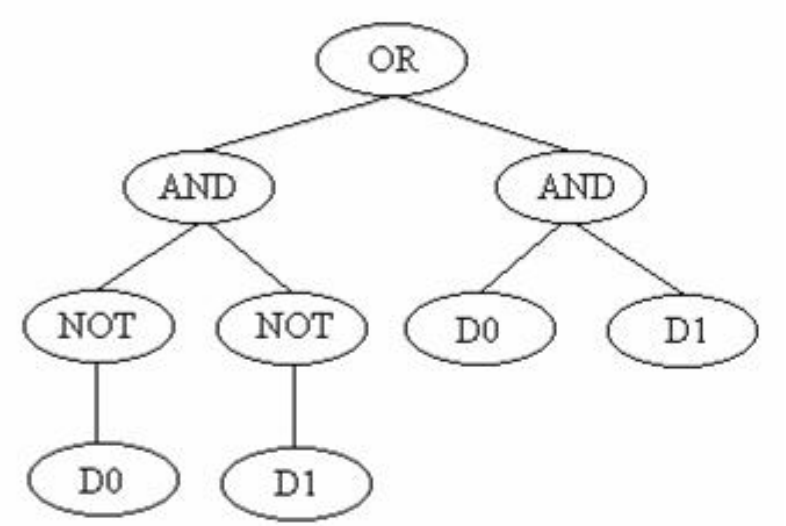
- Uso de estructuras de árboles para representar programas de computadora
- Se predetermina la máxima profundidad de los árboles, pero no su topología precisa
- El tamaño, forma y contenido de los árboles puede cambiar dinámicamente durante el proceso evolutivo

Programación Genética

- Funciones más utilizadas:

1) Operaciones aritméticas ($+, -, *, etc.$)
2) Funciones matemáticas (seno, coseno, exp, log)
3) Operaciones Booleanas (AND, OR, NOT)
4) Operadores condicionales (IF-THEN-ELSE)
5) Funciones que causan iteraciones (DO-UNTIL)
6) Funciones que causan recursión
7) Cualquier función específica del dominio definido

Programación Genética



La expresión S equivalente es:

(OR (AND (NOT D0)(NOT D1))(AND (D0D1))

Algoritmos Genéticos Desordenados

- Desarrollados por Deb y Goldberg hacia fines de los 1980s.
- Motivación: resolver problemas "deceptivos" con un AG.
- Características: longitud cromosómica y tamaño de población variables.
- Se le llama AG desordenado, porque se contrapone al AG "ordenado" que tiene longitud y tamaño de población fijos.

Algoritmos Genéticos Desordenados

- Idea fundamental: representar esquemas en vez de cromosomas.
- Funcionamiento: Comenzar con cromosomas cortos, identificar un conjunto de buenos bloques constructores y después incrementar la longitud del cromosoma para propagar estos buenos bloques constructores a lo largo del resto de la cadena.

Algoritmos Genéticos Desordenados

Ejemplos de cadenas válidas:

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

(1,1)	(1,0)	(1,1)	(4,1)	(4,0)

El número de la izquierda es la posición cromosómica (locus) y el de la derecha el valor del bit.

Algoritmos Genéticos Desordenados

Peculiaridades de la representación:

- Algunas posiciones pueden ser asignadas a más de un bit **(sobre-especificación)**
- Otras posiciones pueden no ser asignadas a ningún bit **(sub-especificación)**

Algoritmos Genéticos Desordenados

¿Cómo lidiar con la **sobre-especificación**?

- Imponer un orden determinístico de evaluación. Por ejemplo, de izquierda a derecha. De esta manera, se ignoran los valores posteriores al primero detectado para una cierta posición cromosómica.
  Ejemplo: si tenemos la cadena

  $$\{ (1,0) (2,0) (4,1) (4,0) \}$$

  tomamos el valor de 1 para el cuarto bit

Algoritmos Genéticos Desordenados

¿Cómo lidiar con la **sub-especificación**?

- Tenemos que evaluar la aptitud de cadenas parcialmente definidas.
- Podemos ver una cadena como un esquema candidado. Ejemplo: si tenemos la cadena

$$\{ (1,0)\ (2,0)\ (4,1)\ (4,0) \}$$

se trata del esquema candidato: $00*1$

Algoritmos Genéticos Desordenados

¿Cómo lidiar con la **sub-especificación**?

- Hay varios métodos:

  1) Promedios

  2) Plantillas competitivas

Algoritmos Genéticos Desordenados

Método de los Promedios:

- Generar aleatoriamente valores para los lugares faltantes un cierto número de veces
- La aptitud de la cadena sub-especificada será el promedio de aptitud de estas muestras

Algoritmos Genéticos Desordenados

Método de los Promedios:

- Motivación: se intenta calcular el promedio del esquema candidato.
- Problema: La varianza de esta aptitud promedio frecuentemente será demasiado alta para que se pueda obtener un valor significativo mediante un muestreo aleatorio.

Algoritmos Genéticos Desordenados

Método de las Plantillas Competitivas:

- Usar un *hillclimber* para obtener un óptimo local.
- Al correr el mGA, se evalúan las cadenas sub-especificadas llenando los bits faltantes con el óptimo local. De esa manera, es posible calcular la aptitud de cualquier esquema candidato.

Algoritmos Genéticos Desordenados

Método de las Plantillas Competitivas:

- Motivación: Por definición, un óptimo local es una cadena que no puede mejorarse con el cambio de un solo bit. Por tanto, si los bits definidos de un esquema candidato mejoran el óptimo local, vale la pena efectuar una mayor exploración.
- Problema: El método es víctima de la explosión combinatoria.

Algoritmos Genéticos Desordenados

Operan en 2 fases:

- Fase Primordial
- Fase Yuxtaposicional

Algoritmos Genéticos Desordenados

El objetivo de la **fase primordial** es generar esquemas cortos que sirven como los bloques constructores en la **fase yuxtaposicional** en que éstos se combinan.

Problema: ¿Cómo decidir qué tan largos deben ser estos "esquemas cortos"?

Algoritmos Genéticos Desordenados

La solución que plantearon Goldberg y sus colegas al problema de la longitud de los esquemas cortos fue adivinar el "orden" de dichos esquemas.
El "orden" de un esquema es el número de posiciones fijas que tiene. Por ejemplo: $o(0\#\#10) = 3$

Algoritmos Genéticos Desordenados

Si se adivina el orden del esquema (al que llamaremos k), entonces se procede a generarse por enumeración todos los esquemas de ese orden y de la longitud requerida, l.
Por ejemplo, si l=8 y k=3, tenemos:

$$\{(1, 0)\ (2, 0)\ (3, 0)\}$$
$$\{(1, 0)\ (2, 0)\ (3, 1)\}$$
$$\vdots$$
$$\vdots$$
$$\{(1, 1)\ (2, 1)\ (3, 1)\}$$
$$\{(1, 0)\ (2, 0)\ (4, 0)\}$$

$\{(1, 0)\ (2, 0)\ (4, 1)\}$

:

:

$\{(6, 1)\ (7, 1)\ (8, 1)\}$

Algoritmos Genéticos Desordenados

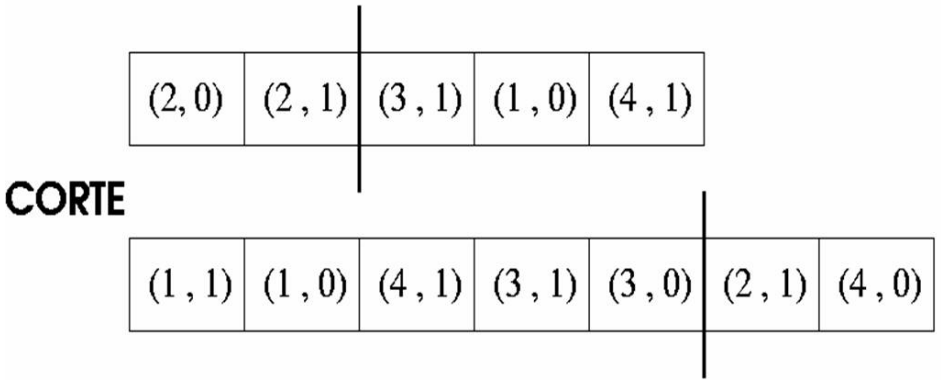
Estos esquemas cortos (de orden predefinido) se generan en la fase primordial y luego evaluamos sus aptitudes. Después de eso aplicamos sólo selección a la población (sin cruza ni mutación) para propagar los buenos bloques constructores, y se borra la mitad de la población a intervalos regulares.

Algoritmos Genéticos Desordenados

Después de un cierto número (predefinido) de generaciones, entramos a la fase yuxtaposicional. A partir de este punto, el tamaño de la población permanecerá fijo y usaremos selección y 2 operadores llamados **corte y unión**. Debido a la naturaleza de este algoritmo, estos operadores siempre producen cadenas válidas.

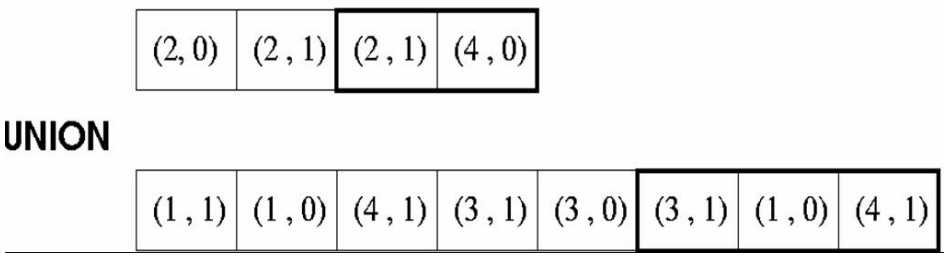
Algoritmos Genéticos Desordenados

Operador de Corte:



Algoritmos Genéticos Desordenados

Operador de Unión:



Algoritmos Genéticos Desordenados

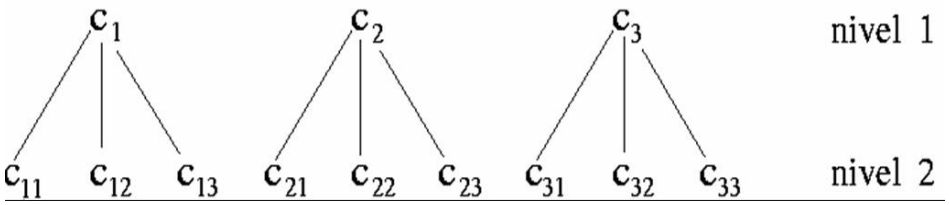
- El AG desordenado está diseñado para resolver problemas sumamente difíciles para un AG tradicional (p.ej. deceptivos).
- A pesar de ser una propuesta interesante, ha tenido poco uso práctico porque suele ser víctima de la "maldición de la dimensionalidad".

Algoritmo Genético Estructurado

- Representación híbrida propuesta por Dasgupta a mediados de los 1990s.
- Es un compromiso entre la representación lineal del AG tradicional y la de árbol usada por la programación genética.
- Usa una representación jerárquica con un mecanismo de dominancia similar al de los diploides.

Algoritmo Genético Estructurado

El AG estructurado codifica estructuras genéticas de varios niveles (grafos dirigidos o árboles) como se muestra a continuación:



Algoritmo Genético Estructurado

Los genes en cualquier nivel pueden ser **pasivos** o **activos**, pero los genes de alto nivel activan o desactivan conjuntos de genes de más bajo nivel, lo que significa que cualquier cambio pequeño en un alto nivel se magnifica en los niveles inferiores.

Algoritmo Genético Estructurado

La idea principal del AG estructurado es que los genes de alto nivel deben **explorar** las áreas potenciales del espacio de búsqueda y los de bajo nivel deben **explotar** ese sub-espacio. El uso principal de este tipo de técnica es para funciones dinámicas.

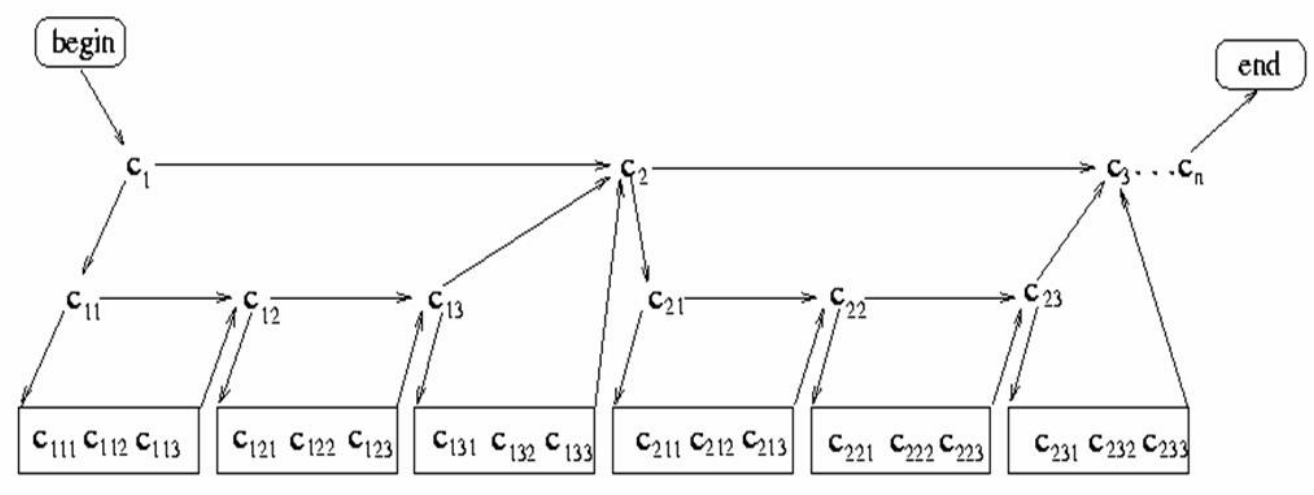
Algoritmo Genético Estructurado

A pesar de tener una estructura jerárquica, se codifica como un cromosoma lineal de longitud fija:

$$(c_1 \quad c_2 \quad c_3 \quad c_{11} \quad c_{12} \quad c_{13} \quad c_{21} \quad c_{22} \quad c_{23} \quad c_{31} \quad c_{32} \quad c_{33})$$

Algoritmo Genético Estructurado

Sin embargo, el AG estructurado requiere de una estructura de datos más complicada que la del AG tradicional:



Algoritmo Genético Estructurado

Problemas:

- Implementación más complicada
- Mapeo entre genotipo y fenotipo extremadamente complejo
- Se degrada su desempeño conforme se aumenta el número de jerarquías utilizadas

Recomendaciones para diseño de buenas representaciones

- Palmer (1994) hizo varias recomendaciones en el contexto de representaciones de árbol, en torno los puntos clave a tomarse en cuenta al proponer una representación:
- 1) Nuestra codificación debe poder representar todos los fenotipos posibles.

Recomendaciones para diseño de buenas representaciones

- 2) Nuestra codificación no debe tener sesgos, de manera que todos los individuos posibles se encuentren representados de manera equitativa en el conjunto de todos los genotipos posibles.
- 3) Nuestra codificación no debería permitir soluciones infactibles.

Recomendaciones para diseño de buenas representaciones

- 4) La decodificación del genotipo al fenotipo debiera ser simple.
- 5) Una codificación debe poseer localidad (o sea, cambios pequeños en el genotipo debieran resultar en cambios pequeños en el fenotipo).

Recomendaciones para diseño de buenas representaciones

- Ronald (1995) hizo también sugerencias en torno a cómo elegir una buena representación:
- 1) Las codificaciones deben ajustarse a un conjunto de operadores genéticos de tal forma que se preserven (o transmitan) los bloques constructores de padres a hijos.

Recomendaciones para diseño de buenas representaciones

- 2) Una buena codificación debe minimizar la epístasis.
- 3) Deben preferirse las soluciones factibles.
- 4) El problema debe estar representado a un nivel correcto de abstracción.

Recomendaciones para diseño de buenas representaciones

- 5) Las codificaciones deben explotar un mapeo apropiado del genotipo al fenotipo en caso de que no sea posible producir un mapeo simple entre estos 2 espacios.
- 6) No debieran usarse formas isomórficas, en las cuales el fenotipo de un individuo es codificado con más de un genotipo (esto ha sido debatido).