

Capítol 4: Algorismes Genètics (continuació)

Ref.

An Introduction to Genetic Algorithms

Melanie Mitchell

MIT Press, 1997.

5. Aplicacions

Evolució de xarxes neurals

- Els AG poden ser aplicats en les xarxes neurals per l'evolució de
 - els pesos en una arquitectura donada
 - l'arquitectura pròpiament dita
 - la regla d'aprenentatge

5. Aplicacions

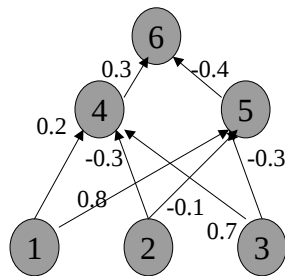
XN. Evolució de pesos

- Es tracta d'usar els AG enlloc de l'algorisme de backpropagation
- Es va provar per classificar son sota aigua (*lofargramas*)
 - La classificació és: senyals interessant / no interessants
 - L'algorisme de backpropagation presentava un problema de mínims locals importats
 - L'algorisme de backpropagation exigeix un aprenentatge supervisat
 - Dades: 236 senyals de lofargramas i les classificacions fetes per experts.
- Xarxa:
 - unitats d'input: 4, corresponent als paràmetres usats pels experts en la classificació
 - unitat d'output: 1 (interessant/no interessat)
 - unitats ocultes: 2 nivells de 7 + 10 unitats
 - Estructura feedforward, totalment connectada $\Rightarrow 108 \text{ connexions} + 18 \text{ thresholds (bias)} = 126 \text{ pesos}$

5. Aplicacions

XN. Evolució de pesos

- Algorisme genètic: codificació
 - 1 cromosoma = 1 vector de 126 pesos
 - 1 gen = 1 nombre real
 - Ordre dels pesos dins del cromosoma: d'esquerra a dreta, i de dalt a baix
Exemple: (0.3 -0.4 0.2 0.8 -0.3 -0.1 0.7 -0.3)



5. Aplicacions

XN. Evolució de pesos

- Funció d'adaptació:
 - Es calcula la sortida per a tots els exemples
 - Es calcula la suma dels errors quadràtics de tots els exemples. Aquest és el valor d'adaptació
- Inicialització:
 - 50 cromosomes
 - valors aleatoris dels pesos entre [-1.0, 1.0]
- Selecció: mètode del rang

Veurem el mètode del rang més endavant.

5. Aplicacions

XN. Evolució de pesos

- Creuament:
 - Seleccionar 2 pares
 - Per a tota unitat oculta o de sortida del descendent, seleccionar un dels pares de forma aleatòria i copiar
 - Només es genera un descendent.
 - Exemple:
$$\begin{array}{l} (0.3 \ -0.4 \ 0.2 \ 0.8 \ -0.3 \ -0.1 \ 0.7 \ -0.3) \\ (0.7 \ -0.9 \ 0.3 \ 0.4 \ 0.8 \ -0.2 \ 0.1 \ 0.5) \end{array} \implies (0.7 \ -0.9 \ 0.2 \ 0.4 \ -0.3 \ -0.2 \ 0.7 \ 0.5)$$

5. Aplicacions

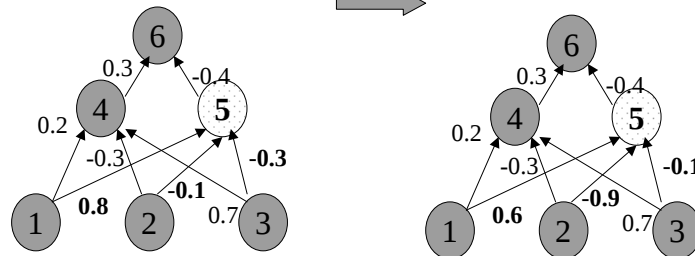
XN. Evolució de pesos

- Mutació

- Seleccionar una unitat oculta o de sortida
- Alterar els seus pesos d'entrada amb un valor aleatori entre [-1.0, 1.0]
- Exemple:

(0.3 -0.4 0.2 **0.8** -0.3 **-0.1** 0.7 **-0.3**)

(0.3 -0.4 0.2 **0.6** -0.3 **-0.9** 0.7 **-0.1**)



5. Aplicacions

XN. Evolució de pesos

- Resultats
 - Algorisme genètic: 200 generacions (* 50 cromosomes = 10.000 avaluacions)
 - Backpropagation: 5.000 èpoques
 - AG més ràpid i millors pesos.
 - Però amb mètodes de backpropagation més acurats, els AG són pitjors.
- ⇒ És millor reservar els algorismes genètics per realitzar aprenentatge no supervisat.

5. Aplicacions

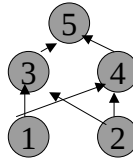
XN. Evolució de l'arquitectura

- L'arquitectura d'una xarxa neural és fonamental per l'èxit d'aquesta. ... Però depèn de l'habilitat del dissenyador.
 - ⇒ És interessant tenir una forma automàtica per determinar l'estructura d'una x.n. per a una aplicació determinada
- Existeixen dos aproximacions per optimitzar l'estructura d'una x.n. usant algorismes genètics:
 - Codificació directe: una arquitectura es codifica directament en un cromosoma.
 - Codificació gramatical: l'AG evoluciona gramàtiques que poden usar-se per desenvolupar arquitectures.

5. Aplicacions

XN. Codificació directe

- Restriccions: xarxa feed forward amb nombre d'unitats conegudes
⇒ L'AG decideix la connexió topològica
- Codificació:
 - Mitjançant una matriu: A= hi ha connexió; 0= no hi ha connexió
 - A: indica que el pes es pot aprendre
 - Exemple
0 0 0 0 0
0 0 0 0 0
A A 0 0 0
A A 0 0 0
0 0 A A 0
- Funció d'adaptació:
 - S'executa la xarxa amb backpropagation per un cert nombre d'èpoques
 - La suma de l'error quadràtic és la funció d'adaptació



5. Aplicacions

XN. Codificació directe

- Selecció: proporcional a l'adaptació
- Creuament:
 - Es selecciona aleatòriament una fila de les matrius dels pares i s'intercanvien
 - $P_c=0.6$
 - 2 descendents
- Mutació: $P_m=0.005$ per bit.
- Resultats:
 - Es van trobar topologies per 3 problemes: XOR, 4-quadrants, codificació/decodificació
- Problema: escalabilitat del mètode.
 - Com obtenir una funció d'adaptació?
 - Quin és el cost de l'avaluació de la funció d'adaptació?
 - Representació de xarxes recurrents

El problema de la codificació/decodificació consisteix en que la xarxa neural reconegui una entrada (i.e. entrada = sortida); però l'estructura de la xarxa té capes ocultes amb menor nombre d'unitats que la d'entrada, per la qual cosa necessàriament la XN està implementant una codificació i una decodificació.

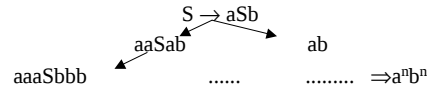
5. Aplicacions

XN. Codificació gramatical

- És una alternativa a la codificació directa que intenta resoldre els problemes d'escalabilitat.
- Gramàtica:
 - Genotipus, del qual es deriva la xarxa.
 - Conjunt de regles que es poden aplicar a la producció d'una estructura
 - Exemple:

$S \rightarrow aSb$
 $S \rightarrow \epsilon$

a, b : símbols terminals
 S : símbol no terminal
 ϵ : cadena buida
- Generació de l'estructura

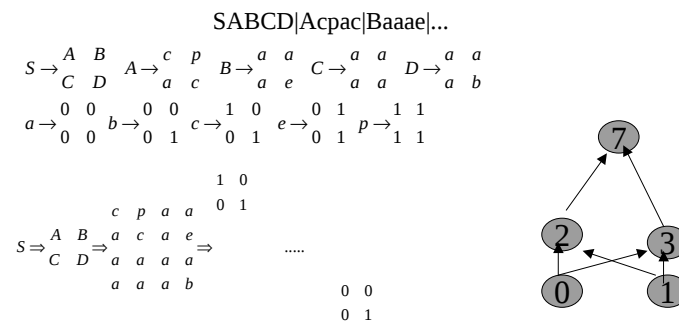


Les gramàtiques s'apliquen en les frases del llenguatge natural, en l'elaboració de programes, en arquitectures de XN.

5. Aplicacions

XN. Codificació gramatical

- Codificació: 5 locus:
 - Primer: la part esquerra de la regla gramatical
 - Segon a cinquè: símbols de la part dreta de la regla gramatical
 - Al.lels: símbols A-Z i a-z
 - Exemple:



Estructura de xarxa per una funció XOR.

5. Aplicacions

XN. Codificació gramatical

- Inicialització:
 - Primers símbols corresponents a la regla d'inci (S)
 - Resta, inicialitzats de forma aleatòria.
- Funció d'adaptació:
 - Entrenar la xarxa amb backpropagation
 - La suma d'errors quadràtics és el valor d'adaptació
- Selecció: proporcional al valor d'adaptació
- Creuament: multipunt
- Mutació:
 - Substitució aleatòria d'un símbol en el cromosoma
 - $P_m = f(\text{distància de Hamming entre els dos pares})$
- Resultats:
 - AG amb codificació gramatical superen la codificació directa en correctesa i en velocitat
 - Escalabilitat millor
 - Cal comprovar amb tasques més complexes.

Mutació: Més distants són els pares, menys mutació.

Escalabilitat: requereix menys cromosomes per codificar regles de construcció que per codificar les arquitectures pròpiament dites (mètode directe).

5. Aplicacions

XN. Evolució de la regla d'aprenentatge

- Restriccions: XN feedforward, totalment connectades, sense capes ocultes.
- Suposició: els pesos s'actualitzen d'acord amb una regla que té la forma:

$$\Delta w_{ij} = f(a_i, o_j, t_j, w_{ij})$$

a_i : input, o_j : output, t_j : patró, w_{ij} : pes

- S'assumeix que aquesta f és una funció lineal de les variables a_i , o_j , t_j , w_{ij} , i els parells possibles entre elles:

$$\Delta w_{ij} = k_0(k_1 w_{ij} + k_2 a_i + k_3 o_j + k_4 t_j + k_5 w_{ij} a_i + k_6 w_{ij} o_j + k_7 w_{ij} t_j + k_8 a_i o_j + k_9 a_i t_j + k_{10} o_j t_j)$$

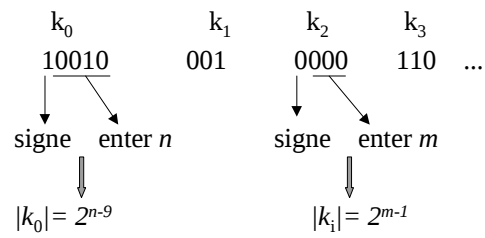
- La regla delta és: $w_{ij} = w_{ij} + \alpha a_i (t_j - o_j)$
- Amb l'AG s'intenta trobar la regla delta o de millors a partir de l'expressió més general.
- L'algorisme genètic ha d'obtenir, doncs, els valors dels k_i .

Es pot veure fàcilment que el paràmetre d'aprenentatge alfa és k_0 .

5. Aplicacions

XN. Evolució de la regla d'aprenentatge

- Codificació:



5. Aplicacions

XN. Evolució de la regla d'aprenentatge

- Funció d'adaptació:
 - Es tracta d'implementar les XN i avaluar-les.
 - Atès que són XN sense capes ocultes, només poden fer funcions de classificacions linealment separables.
 - Es generen 30 possibles classificacions amb un sol output i variant el nombre d'inputs.
 - Mètode per conèixer el valor d'adaptació:
 - Es seleccionen 20 classificacions
 - Per a cada classificació,
 - S'agafen 12 exemples
 - Es crea la xarxa que correspon a la classificació
 - S'inicialitzen els pesos de forma aleatòria
 - S'entrena la xarxa per 10 èpoques
 - La mitjana dels errors de la xarxa per les 20 classificacions és la mesura d'adaptació de la regla d'aprenentatge.

La mitjana d'errors es transforma en %.

5. Aplicacions

XN. Evolució de la regla d'aprenentatge

- Creuament:
 - Dos punts de creuament
 - $P_c = 0.8$
- Mutació: $P_m = 0.01$
- Resultats:
 - 40 regles d'aprenentatge
 - 1000 generacions
 - 1 de cada 10 intents es descobreix la regla delta.
 - L'adaptació de la regla delta resulta en 98%.
- És pot estendre el mètode a xarxes amb més capes?
- Es pot estendre el mètode a xarxes recurrents?
- I amb aprenentatge no supervisat?

La mitjana d'errors es transforma en %.

5. Aplicacions

Generació d'horaris universitaris

- Horaris de cursos universitaris:
 - Aules, professors, ensenyaments
 - Restriccions: no posar totes les hores d'una assignatura el mateix dia
 - ...
- Horaris d'exàmens:
 - Aules, assignatures, professors vigilants.
- Cada universitat té la seva pròpia problemàtica lligada a la seva idiosincràsia.
- Úniques restriccions universals:
 - No hi ha d'haver conflictes (un alumne no pot estar en 2 exàmens al mateix temps)
 - No ha d'haver aules sobresaturades
- Qüestions d'optimització (segon ordre, no obligatòries):
 - Ningú té dos exàmens de forma consecutiva.,

Ref.

A genetic algorithm for university timetabling

E. Burke, D. Elliman, R. Weare.

El problema d'assignar hores als exàmens és una reducció del problema original d'elaborar els horaris d'un curs escolar en una universitat.

5. Aplicacions

Generació d'horaris universitaris

- Codificació:
 - 1 examen = 1 gen
Examen1 Examen2 Examen3
 - Per a cada examen s'especifica el període i l'aula
- Inicialització:
 - Per a cada individu
 - Generar de forma aleatòria l'ordre dels exàmens
 - Per a cada examen:
 - Trobar el primer període en el qual es pot col·locar sense conflicte
 - Trobar l'aula més gran possible. Si no es troba, provar amb el període següent més apropiat.

5. Aplicacions

Generació d'horaris universitaris

- Funció d'adaptació:
 - Duració de l'horari
 - Nombre de conflictes de segon ordre
 - Mal aprofitament de la capacitat de les aules.

$$f(\text{duració}) + g(\text{nombre de conflictes de 2n ordre}) + h_1\left(\sum_{\text{períodes}} \sum_{\text{aules}} h_2(\text{seients buits})\right)$$

F,g,h1,h2: paràmetres que es poden ajustar d'acord amb els interessos de la institució. Per exemple:

$$f(\text{duració}) = (\text{duració} - 10)^2 * 50$$

$$g(x) = x * \text{ADJ, on ADJ és una constant (paràmetre)}$$

$$h_2(y) = y^2$$

$$h_1(z) = \frac{z}{1000}$$

5. Aplicacions

Generació d'horaris universitaris

- Selecció: basada en l'adaptació.
- Creuament: especial per assegurar que els descendents compleixen les restriccions

Seleccionar dos pares

Seleccionar dos punts de creuament aleatoris: x, y

Agafar els exàmens del pare 1, en les posicions $1..x-1$ i $y..n$

Per a cada examen $x..y$ del pare 2, col·locar-lo segons les prioritats següents:

Mateix període, mateixa aula.

Mateix període, diferent aula (la més gran possible).

Període més proper i aula més semblant.

n és el nombre d'exàmens.

5. Aplicacions

Generació d'horaris universitaris

- Mutació:
 - Seleccionar un cromosoma
 - Seleccionar un examen de forma aleatòria
 - Agafar un nou període d'examen
 - Col.locar l'examen d'acord a les prioritats següents:
 - Nou període, aula més gran possible
 - Període més proper (i aula) després del període seleccionat.
- Resultats:
 - Depenen de les constants ADJ
 - Depenen de la duració del període d'exàmens total.

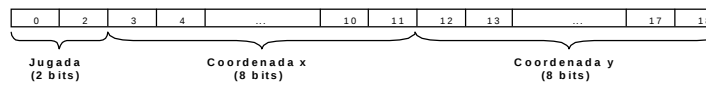
n és el nombre d'exàmens.

5. Aplicacions Futbol.

- Els cromosomes codifiquen possibles jugades.
- Cada possible jugada es codifica amb un codi de jugada i una posició.
- El codi de jugada és un enter codificat amb dos bits.

00 RES
01 ANAR
10 XUTAR
11 --

- La posició de la jugada esta formada per dos números reals que equivalen a la coordenada x i la coordenada y respectivament.



Ref.

Apunts Pep Lluís de la Rosa. Curs 1998-1999.

5. Aplicacions

Futbol.

- Per poder representar nombres de doble precisió amb 8 bits, es realitza un escalat de coordenades: els cromosomes només codifiquen valors reals entre $[-1.37, 1.37]$ que són les dimensions reals del camp.
- Aquesta representació en 18 bits ($2+8+8$) suposa un arbre d'exploració de 2^{18} possibilitats (262.144).
- Si no es realitzés aquest escalat de coordenades caldria representar els números reals amb més bits per no perdre precisió.
 - Si s'utilitzés un *long* (32bits) cada cromosoma es codificaria en 66bits el que equival a $7,378697629484e+19$ possibilitats
 - Si s'utilitzés un *double* (64bits) l'arbre d'exploració tindria $1,361129467684e+39$ possibilitats.
- Tot i que els algorismes genètics són capaços d'explorar arbres molt grans, aquests no son gaire adequats per l'aplicació del jugador de JavaSoccer degut al temps de convergència. Per tant un arbre d'exploració relativament petit permetrà als jugadors emprar menys temps en convergència de l'algorisme cap a la solució, és a dir, cap a la millor jugada.

5. Aplicacions

Modelització d'ecosistemes

- L'evolució no afecta a una espècie de forma aïllada sinó a ecologies on diferents espècies interaccionen.
 - Exemples:
 - Dilema del presoner: quan les estratègies juguen entre sí.
 - Xarxes d'ordenació: quan els hostes i els paràsits competeixen entre sí.
- ECHO:
 - Intent de capturar les característiques generals de totes els ecosistemes.
 - Plataforma per experimentar com els canvis en el model afecten a la subreabundància d'una espècies, etc.

5. Aplicacions

Modelització d'ecosistemes

- Elements:
 - Recursos:
 - Fonts d'energia dels organismes
 - Representats per lletres de l'alfabet.
 - Agents:
 - Cadascun amb un genotipus i fenotipus diferent
 - Té una energia interna.
 - L'energia es gasta pel moviment, per comunicació o simplement per metabolisme (estant aturat).
 - Es pot reproduir quan té suficient energia per crear una còpia del seu propi genoma.
 - Si l'energia cau sota un determinat llindar, l'agent mor i els recursos tornen al lloc on viu.
 - Si un agent no obté recursos en un lloc, tria un altre lloc de forma aleatòria.

5. Aplicacions

Modelització d'ecosistemes

- Genotipus
 - Regla que governa els tipus i quantitats de recursos que l'agent necessita per sobreviure i reproduir-se
 - Regla que governa els tipus i quantitats de recursos que l'agent pot agafar de l'entorn
 - Regla que especifica com es comunica amb els altres agents
 - Característiques físiques visibles pels altres agents.
- Canvis en la població: són una conseqüència de les trobades aleatòries entre agents
- Depenent de les seves regles i l'aparença física que rep de l'altre, quan dos agents es troben poden:
 - Combatre: el perdedor mor i el guanyador es queda els seus recursos
 - Negociar: excedents de recursos, ...
 - Aparellar-se:
 - Creuament: 2 punts, 2 descendents que substitueixen las pares

5. Aplicacions

Modelització d'ecosistemes

- No hi ha una mesura explícita d'adaptació.
- La taxa de reproducció emergeix per la interacció dels agents

... Falta molt perquè el sistema s'assembli a un real. Futur:

- Estudiar l'evolució de mecanismes de comunicació
- Estendre el model per representar comunitats connectades
- Estudiar la dinàmica de l'evolució
- Crear un teorema sobre l'adaptació

6. Temes avançats

Quan s'han d'usar els AG?

- Quan l'espai de cerca és gran
 - L'espai de cerca no és suau ni unimodal
 - L'espai de cerca no es coneix bé
 - La funció d'adaptació és sorollosa
 - La tasca no requereix una solució òptima global
- ⇒ Un AG funcionarà tan bé com qualsevol altre mètode dèbil o fins i tot millor.

Però els AG depenen en gran mesura del mètode de codificació de les solucions, dels operadors (selecció, creuament, mutació i d'altres) i dels paràmetres.

Espai de cerca:

Si l'espai no és gran, es pot buscar la solució mitjançant una cerca exhaustiva, de forma que es garantirà que es troba la solució millor. Mentre que els AG poden caure en una convergència prematura.

Suau i unimodal:

Espai suau: una colina (versus pic).

Unimodal: consisteix en un sol pic, per exemple.

Si l'espai de cerca és suau i unimodal, un algorisme basat en el gradient ascendent, com el mètode "steepest ascent hill-climbing", poden ser més eficients que els AG.

Si l'espai és coneix bé, un mètode de cerca basat en heurístiques específiques del domini pot proporcionar millors resultats que un AG.

Si la funció d'adaptació és sorollosa, els AG són bons, proporcionen uns resultats robustos, comparats amb altres mètode de tipus hill-climbing que poden extravar-se.

Òptim: la tasca necessita una solució suficientment bona, però no cal que sigui la millor en termes globals.

6. Temes avançats

Codificació d'un problema

- Codificació binària
 - Variacions: codificació grisa, codificació basada en diploids
 - Codificacions de cadenes llargues permeten un grau de paral·lelisme més elevat
 - Codificacions curtes permeten una major quantitat d'esquemes.
 - És una codificació poc natural i no sempre aplicable
- Codificació basada en un conjunt de caràcters o amb valors reals
- Codificació basada en arbres
 - Els arbres poden créixer de forma descontrolada, perdent estructura en les solucions
 - Poden ser difícils d'interpretar

Totes les codificacions tenen pros i contres.

⇒ Dexiar a l'AG adaptar la codificació al problema

Normalment, els GA usen una codificació binària de longitud i ordre fix.

Un esquema és un patró que identifica al conjunt de cromosomes que pertanyen a un hiperpla de recerca.

Intuitivament: identifica un conjunt de cromosomes que responen a un determinat patró.

Exemple: *11* és un esquema que representa els cromosomes:

0110

1110

0111

1111

Cada codificació depèn del problema a resoldre i no hi ha una "guia d'usuari" per escollir-ne una.

6. Temes avançats

Adaptació de la codificació

- Justificació:
 - Problema del “linkatge”: locus que estiguin relacionats s’haurien de mantenir junts en el moment dels creuaments. Però, a priori, es deconeix quin són aquests locus.
 - La representació basada en una longitud fixa, limina la complexitat de les solucions.
- Solucions
 - Operador d’inversió
 - Evolució dels punts de creuament
 - AG “messy” (desordenats)

6. Temes avançats

Operador d'inversió

- Operador específic per resoldre el problema del “linkatge”
- Basat en el fet que un gen és independent de la seva posició en el cromosoma
- Invertint part del cromosoma, es manté, doncs, la semàntica original
- Es tracta d'evolucionar l'ordre dels bits en el cromosoma.
- Cada al·lel es representa amb la seva posició real, per a que aquesta sigui avaluada en el moment de calcular l'adaptació del cromosoma.
Exemple: 00010101 es codificaria com:
 $\{(1,0),(2,0),(3,0),(4,1),(5,0),(6,1),(7,0),(8,1)\}$
- L'operador d'inversió: selecciona dos punts del cromosoma i els canvia d'ordre:
Exemple: canvi entre el 3 i el 6:
 $\{(1,0),(2,0),(6,1),(4,1),(5,0),(3,0),(7,0),(8,1)\}$

Amb aquesta codificació s'aconsegueix que la posició real d'un al·lel sempre sigui coneguda.

L'operador d'inversió no canvia doncs l'adaptació del cromosoma, ja que en el moment de calcular-la es posaran en ordre els al·lells d'acord amb la seva posició real.

La idea darrera d'aquest operador és que es produeixin ordres que responguin a esquemes amb més probabilitats de sobreviure. Per exemple:

1) l'esquema corresponent a l'exemple original és 00**01**

2) el nou esquema és: 001****.

El nou esquema té més probabilitats de sobreviure a un creuament de punt fix (ja veurem més endavant que aquest tipus de creuaments afavoreix els extrems).

6. Temes avançats

Operador d'inversió. Problemes

- Suposem el creuament dels següents cromosomes:

$\{(1,0),(2,0),(6,1),(5,0),(4,1),(3,0),(7,0),(8,1)\}$

$\{(5,1),(2,0),(3,1),(4,1),(1,1),(8,1),(6,0),(7,0)\}$



$\{(1,0),(2,0),(6,1),(4,1),(1,1),(8,1),(6,0),(7,0)\}$

$\{(5,1),(2,0),(3,1),(5,0),(4,1),(3,0),(7,0),(8,1)\}$

- Resultats:

- Descendent 1:

- Té dos còpies per la posició 1 i 6
- No té cap còpia per les posicions 3 i 5

- Descendent 2:

- Té dos còpies per les posicions 3 i 5
- No té cap còpia per les posicions 1 i 6

6. Temes avançats

Operador d'inversió. Solucions

- Permetre només el creuament entre cromosomes amb les mateixes permutacions
- Usar una aproximació de tipus “màster/esclau”:
 - escollir un pare com a màster,
 - reordenar temporalment l'altre pare (esclau) perquè tingui el mateix ordre que el màster.

Qualsevol de les dues solucions té les seves pròpies limitacions.

6. Temes avançats

Evolució dels punts de creuament

- Es tracta d'evolucionar les posicions de creuament
- S'associa a cada solució un estring anomenat "plantilla de cromosoma" que representa les posicions de creuament (1 → creuament, 0 → no creuament).

Exemple: un cromosoma que es pugui creuar després de les posicions 4 i 7 es representaria com:

10011111:000100100

Usant una simplificació basada en !:

1001!111!1

- Aquests punts de creuament són heredats pels descendents:

1 0 0 1! 1 1 1! 1	→	1 0 0 1! 0 0! 1! 0
0 0 0 0 0 0! 0 0		0 0 0 0 1 1 0 1

- L'operador de mutació actua tant com en els cromosomes com en les seves plantilles.

6. Temes avançats

AG “messy”

- Es tracta de construir cromosomes cada cop més grans i adaptats a partir d'altres més petits.
- Cada bit és etiquetat amb la seva posició (locus) real
- Sota-especificació: no tots els cromosomes tenen totes les posicions
Exemple: en un problema de 4 bits: $\{(1,0),(2,0),(4,1)\}$
- Sobre-especificació: alguns cromosomes poden tenir posicions repetides.
Exemple: en un problema de 4 bits: $\{(1,0),(2,0),(4,1),(4,0)\}$
- Funció de fitness:
 - Sobre-especificació: First-come-first-served
 - Sota-especificació: Mètode de plantilles competitives.

AG "messy" està biològicament motivada: la natura no va contruir un DNA de longitud $5.0 \cdot 10^9$ directament i va sortir l'home, sinó que formes més simples de vida van anar evolucionant cap a aquest DNA complex.

FCFS: el primer valor que es troba per una posició, és la que s'agafa.

A part del mètode de plantilles competitives, hi ha d'altres, com el mètode de la mitjana, però que tenen resultats pitjors.

6. Temes avançats

AG “messy”. Fitness per sota-especificació

- Es consideren esquemes enlloc de solucions
Exemple: $\{(1,0),(2,0),(4,1)\} \rightarrow 00^*1$
- Es tracta, doncs, d’avaluar un cromosoma parcial.
- Passos:
 - Trobar un òptim local inicial mitjançant una tècnica de hill-climbing
 - Executar l’AG, omplenant les posicions les cromosomes sota-especificats a partir dels resultats obtinguts mitjançant l’òptim local inicial.

6. Temes avançats

AG “messy”. Funcionament i problemes

- Funcionament en dues fases:
 - Primordial
 - Enriquiment de la població amb esquemes petits
 - Quin és la mida k ideal d'aquests esquemes?
 - Juxtaposicional.
 - Combinar els esquemes de la fase anterior de forma convenient: cut + splice
 - Operador cut: talla un cromosoma en un punt aleatori

$$\{(2,0),(3,0),(1,1),(4,1),(6,0)\} \longrightarrow \begin{matrix} \{(2,0),(3,0)\} \\ \{(1,1),(4,1),(6,0)\} \end{matrix}$$
 - Operador splice (): enganxa dos cromosomes

$$\begin{matrix} \{(1,1),(2,1),(3,1)\} \\ \{(1,0),(4,1),(3,0)\} \end{matrix} \longrightarrow \{(1,1),(2,1),(3,1), (1,0),(4,1),(3,0)\}$$
- Problemes:
 - Coneixement de la mida mínima d'un esquema (k)
 - Combinatòria enorme d'aquests k esquemes

6. Temes avançats

Mètodes de selecció

- Qüestions:
 - Com seleccionar els individus pe reproducció?
 - Quants descendents s'han de generar?
- Objectiu: filtrar els individus per obtenir una adaptació millor.
- Equilibri entre explotació (creuament) i exploració (mutació).
 - Una selecció molt forta pot significar que individus subòptims dominin la població, reduint la diversitat per canvis i progrés
 - Una selecció molt dèbil provocarà una evolució molt lenta.
- Mètodes:
 - Selecció proporcional a l'adaptació
 - Sigma Scaling
 - Elitisme
 - Selecció de Boltzmann
 - Selecció per rang (rank)
 - Selecció per torneig (*tournament*)
 - Selecció d'estat estable (*steady-state*)

Exploració: generar individus que pertanyen a hiperplans de cerca diferents, per la qual cosa interessa tenir una gran variabilitat dintre de la població (probabilitat de mutació gran).

Explotació: Buscar ràpidament, dintre dels hiperplans de cerca actuals (als associats a la població actual) un valor òptim (possiblement local).

Per primar l'explotació interess que hi hagi poca variabilitat dintre de la població actual (probabilitat de mutació baixa).

6. Temes avançats

Selecció proporcional a l'adaptació

- Mètode
 - Suma el valor de l'adaptació total dels N individus de la població. (T)
 - Repetir N cops:
 - Escollir aleatòriament un enter r entre 0 i T
 - Recórrer els individus de la població, sumant el seu valor d'adaptació fins que la suma sigui superior o igual a r . Aquest és l'individu escollit.
- Problema: convergència prematura

De fet és una família de mètodes. El més conegut és el que es motra aquí: la ruleta. Però també hi ha d'altres, com el mostreig estocàstic universal.

6. Temes avançats

Sigma Scaling

- Objectiu: evitar la convergència prematura
- Mètode: establir una correspondència entre el valor brut d'adaptació a valors esperats.
- Càlcul del valor esperat:

$$ExpVal(i,t) = \begin{cases} 1 + \frac{f(i) - \bar{f}(t)}{2\sigma(t)} & \text{si } \sigma(t) \neq 0 \\ 1.0 & \text{si } \sigma(t) = 0 \end{cases}$$

$ExpVal(i,t)$	valor esperat per l'individu i al temps t
$f(i)$	valor d'adaptació
$\bar{f}(t)$	mitjana de valors d'adaptació
$\sigma(t)$	desviació estàndard de la funció d'adaptació

De fet, el mètode de "sigma scaling" pertany a la família de mètodes d'"escalring", l'objectiu del qual és usar sempre valors esperats més que valors bruts.

Si el valor esperat cau per sota de 0, s'inicialitza a 0.1 de forma que individus amb una adaptació molt minça tinguin alguna oportunitat de sobreviure.

6. Temes avançats

Elitisme

- Es tracta de retenir els millors individus de cada generació.

6. Temes avançats

Selecció de Boltzmann

- Una temperatura que contínuament es fa variar, controla la taxa de selecció.
 - Inici: temperatura alta → la pressió de selecció és baixa, tots els individus tenen una probabilitat raonable de reproduir-se
 - Disminució de la temperatura → augment de la pressió de selecció
- Exemple d'implementació:

$$ExpVal(i,t) = \frac{e^{f(i)/T}}{\langle e^{f(i)/T} \rangle}$$

T: temperatura

$\langle \rangle$: mitjana de la població

6. Temes avançats

Selecció per rang

- Els individus s'ordenen d'acord amb el seu grau d'adaptació
- El valor esperat de cada individu depèn de la seva posició en el resultat de l'ordenació (rang)
- Dos individus en les posicions i i $i+1$ poden tenir el mateix valor esperat, malgrat que el seu grau d'adaptació absolut sigui molt distant.
- Exemple (mètode per rang lineal):

$$ExpVal(i, t) = Min + (Max - Min) \frac{rank(i, t) - 1}{N - 1}$$

N Nombre d'individus

Min Valor esperat per l'individu que ocupa la posició 1

Max Valor esperat per l'individu que ocupa la posició N

rank(i,t) Posició que ocupa l'individu i dins de la classificació

6. Temes avançats

Selecció per torneig

- S'escullen dos individus de forma aleatòria
- Es genera un nombre aleatori r entre 0 i 1
- Si $r < k$, el millor dels dos individus s'escull per ser pare
- Si $r \geq k$, el pitjor dels dos individus és el pare.
- Els pares es retornen a la població, per poder ser re-escollits.

k és un paràmetre del l'algorisme. Per exemple: 0.75.

6. Temes avançats

Selecció d'estat estable

- Només una fracció d'una generació és substituïda
- Acostuma a substituir-se els que són pitjor (pitjor valor d'adaptació).
- Gap generacional: fracció dels nous individus en la població.

6. Temes avançats

Creuament. Posició única de creuament

- És el mètode més senzill.
- Problemes:
 - Bias posicional:
 - Per exemple, si 0111 ($\in *11*$ i $\in ***1$) és una solució òptima al problema, la població següent el conté:

0110
 1110
 0111
 1111
 - El creuament pot fer perdre l'esquema, dificultant consegüentment la seva convergència.

0110 ($\in *11*$)
 0001 ($\in ***1$)



0101 ($\notin *11*$)
 0010 ($\notin ***1$)
 - Efecte de punt final: els segments intercanviats entre els dos pares, sempre contenen els finals dels cromosomes.

6. Temes avançats

Creuament. Altres mètodes

- Posició doble de creuament
 - Redueix el problema del bias posicional:
 - la ruptura d'esquema llargs és menys probable
 - es poden combinar més esquemes
 - Redueix el problema de l'efecte de punt final, ja que es poden intercanviar esquemes que no continguin extrems dels cromosomes.
- Creuament uniforme parametritzat: l'intercanvi succeeix en cada posició amb una probabilitat p

Normalment: $0.5 \leq p \leq 0.8$

6. Temes avançats

Mutació.

- La mutació és fonamental per evitar que la població es fixi
- L'equilibri amb l'operador de creuament, i un mecanisme de selecció adequat la complementen.

... Però hi ha pocs mètodes.

6. Temes avançats

Espècies, migracions, reciclatge i altres.

- Problema: convergència prematura:
 - Pèrdua d'al·lells crítics degut a la selecció
 - Alteració d'un esquema degut al creuament
 - Fixació dels paràmetres d'exploració vs. exploració.
- Solucions:
 - Definir diverses espècies. L'evolució de cada espècies es comporta bàsicament com l'evolució de la població d'un sol AG.
 - Exemple: 4 espècies
 - Espècie I: espècie més adaptada. No pateix ni creuament ni mutacions. El cromosoma de l'espècie I és la millor solució al problema.
 - Espècie II: s'encarrega de l'exploració de nous hiperplans de cerca
 - Espècie III: explota els hiperplans de cerca, cercant de forma ràpida u'n òptim
 - Espècie IV: l'exploració i l'explotació tenen un mateix pes.

Ref.

Implementació d'un algorisme genètic basat en la migració i la selecció artificial.

GAMAS

Pere Ridao

6. Temes avançats

Espècies, migracions, reciclatge i altres.

- Migració:
 - A cada generació, un número aleatori d'individus migra
 - de l'espècie II a la III
 - de l'espècie III a la IV
 - de l'espècie IV a la II
 - La freqüència de migració decreix a mida que l'algorisme evoluciona.
- Selecció artificial:
 - A cada generació, els individus més adaptats de les espècies II, III, IV són capturats sobre l'espècie I.
 - Un individu de l'espècie I només es substituït quan hi ha un altre de la resta d'espècies més adaptat.

La migració permet, per exemple, que cromosomes de l'espècie II saltin a l'espècie III, introduint nous esquemes que seran explotats (òptims).

6. Temes avançats

Espècies, migracions, reciclatge i altres.

- Reciclatge:
 - Després d'un nombre predeterminat de generacions, l'espècie IV és totalment repoblada per l'espècie I.
 - Això vol dir reintroduir al procés d'evolució individus molt adaptats que emigraran a altres espècies

⇒ Més possibilitats de trobar òptims.
- Biaix inicial de les poblacions: cada espècie té una població inicial esbiaixada per assegurar que s'exploren diferents hiperespais en els primers instants.

La migració permet, per exemple, que cromosomes de l'espècie II saltin a l'espècie III, introduint nous esquemes que seran explotats (òptims).

6. Temes avançats

Altres operadors.

- *Crowding*:
 - Un descendent substitueix a l'individu més similar a ell en la població.
 - Evita tenir individus molt similars
 - Relacionat amb la funció d'adaptació compartida: el valor d'adaptació d'un individu disminueix amb la presència d'altres similars en la població.
- Estratègies d'aparellament
 - Només creuar parelles d'individus similars.
 - No permetre creuar parelles d'individus similars
 - Aparellament restringit en l'espai: la població evoluciona en diferents gralles espacials, i només es poden creuar individus de gralles veïnes.

Crowding (multitud) el podriem traduir com operador d'amuntonament, o d'allaut de gent... ?

El creuament d'individus similars porta a la distinció d'espècies o grups d'aparellament.

Evitant el creuament d'individus similars s'afavoreix la diversitat.

6. Temes avançats

Paràmetres

- Els paràmetres dels AG són: la grandària de la població, la probabilitat de creuament, la probabilitat de mutació, longitud dels individus, nombre de generacions i funció d'adaptació.
- Consells segons DeJong:
 - Població de 50-100 individus
 - $P_c = 0.6$
 - $P_m = 0.001$ (per bit)
- Consells segons Schaffer:
 - Població de 20-30 individus
 - $P_c = 0.75 - 0.95$
 - $P_m = 0.01$ (per bit)
- Els paràmetres es poden optimitzar usant un AG
 - AG metanivell (Grefenstette)
- Malgrat si trobem els paràmetres inicials correctes, és necessari adaptar-los durant l'execució de l'AG.
 - Com es realitza aquesta adaptació?