

Algoritmi Genetici

Algoritmi Genetici (*GAs*): classe di modelli di calcolo, motivati da metafora naturale, di tipo casualizzato

- Furono proposti da John Holland (Univeristà del Michigan) tra la fine degli anni 60 e l'inizio degli anni 70
- I *GAs* (Holland, 1975, Goldberg, 1989) sono algoritmi di ricerca che si ispirano ai meccanismi della selezione naturale e della riproduzione sessuale
- I *GAs* simulano l'evoluzione di una popolazione di individui, che rappresentano soluzioni candidate di uno specifico problema, favorendo la sopravvivenza e la riproduzione dei migliori

Terminologia

Molti testi usano termini in analogia a quelli della 'reale' biologia

Gli organismi viventi sono formati da cellule: ogni cellula contiene lo stesso numero di '**cromosomi**' (es., la cellula umana contiene 23 coppie di cromosomi)

Ogni cromosoma può essere visto come una stringa di '**DNA**'

La stringa è divisa in blocchi funzionali detti '**geni**'

Ogni gene è responsabile di un '**tratto**' dell' individuo (es. il colore degli occhi)

I possibili valori del tratto sono chiamati '**alleli**' (es. occhi azzurri, marroni, neri)

Terminologia

Il materiale genetico, cioè l'insieme di tutti i cromosomi contenuti nella cellula, è detto '**genoma**'

La riproduzione può avvenire mediante la '**ricombinazione**' (**crossover**), cioè lo scambio di geni tra due cromosomi

Variazioni casuali in un cromosoma sono dette '**mutazione**'

La capacità di un organismo di adattarsi e riprodursi è detta '**fitness**'

Algoritmi Genetici per l'ottimizzazione

Scopo: risolvere con algoritmi genetici problemi di ricerca (ottimizzazione)

Istanza: $\langle S, f \rangle$, dove S è l'insieme delle soluzioni ammissibili e $f : S \rightarrow \mathbb{N}$ è la funzione obiettivo

Domanda: trova $s \in S$ tale che $f(s) \geq f(x) \forall x \in S$ (problema di max)

Negli algoritmi genetici il termine **cromosoma** si riferisce alla generica **soluzione ammissibile** detta **individuo** o **genotipo**

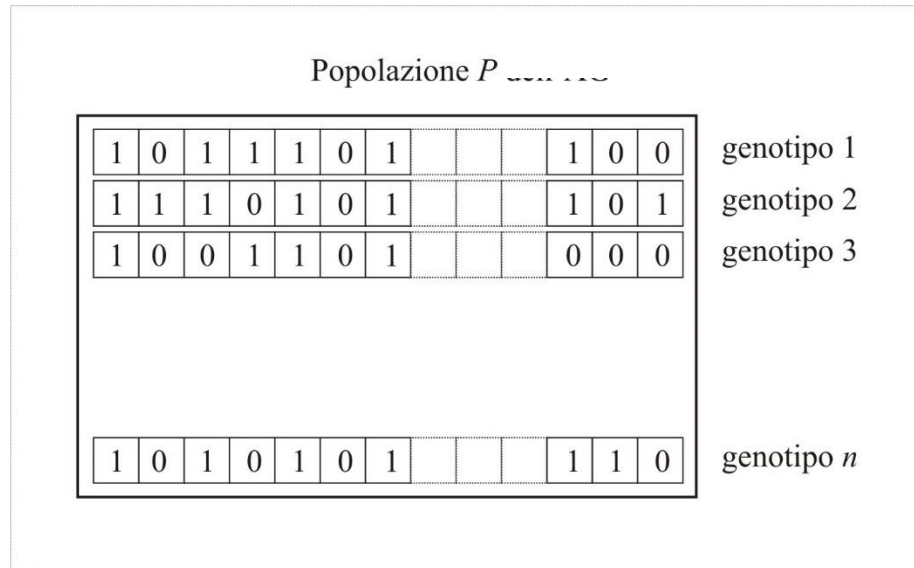
Un insieme di individui (soluzioni ammissibili) è detta **popolazione**

Se l'individuo viene codificato come una **stringa binaria** allora

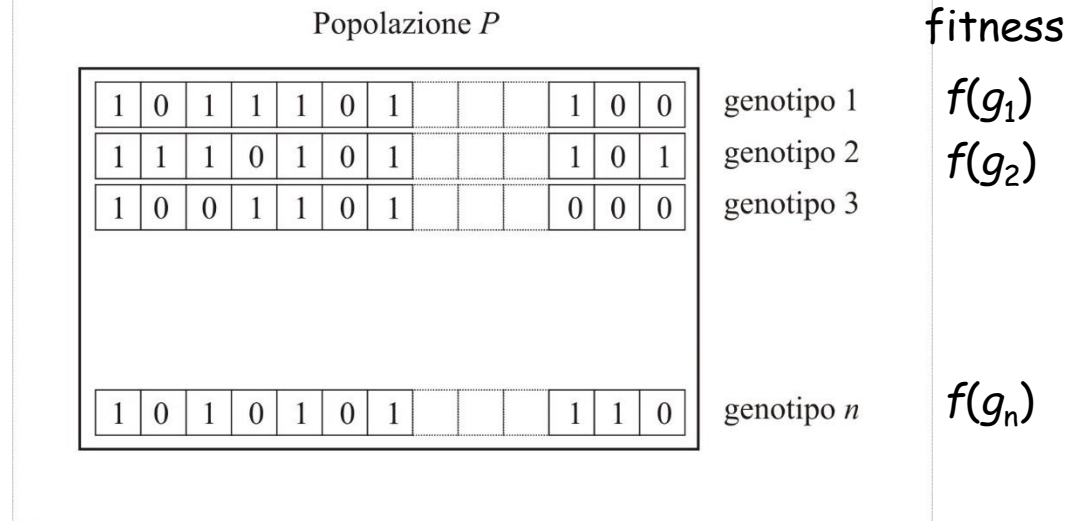
- i geni sono i singoli bit,
- gli alleli valgono '0' o '1' e
- la funzione di fitness corrisponde alla funzione obiettivo f

L'originario modello di Holland

L'originario modello di Holland opera su una popolazione P di n stringhe di bit di lunghezza l fissata che vengono fatte evolvere con regole generalmente probabilistiche



L'originario modello di Holland



La funzione di fitness valuta la bontà degli individui g_i della popolazione P nel risolvere il problema dato:

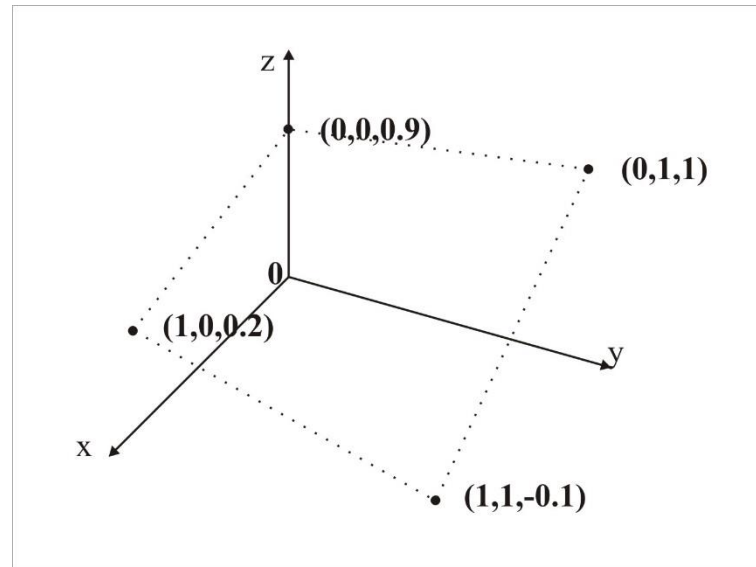
$$f : P \rightarrow (-\infty, +\infty); \quad f(g_i) = f_i$$

L'insieme delle stringhe binarie di lunghezza l ha 2^l elementi; tale insieme rappresenta lo **spazio di ricerca** (search space) del GA

L'insieme di valori di fitness sui punti dello spazio di ricerca è detto **paesaggio d'idoneità** (fitness landscape)

Esempio con genotipi binari di 2 bit

- Il numero di stringhe binarie di lunghezza 2 è $2^l = 2^2 = 4$
- Lo spazio di ricerca del GA è
$$S = \{(0,0), (0,1), (1,0), (1,1)\}$$
- I valori di fitness sui punti di S definiscono il paesaggio d'idoneità del GA



L'Algoritmo

ALGORITMO Genetico

P = popolazione iniziale

While *< not Fine >* **do**

*P = popolazione ottenuta aggiornando la
popolazione corrente P con regole stocastiche R_1, \dots, R_m*

Ogni iterazione in questo processo è detta '**generazione**'

L'esecuzione di un algoritmo genetico prevede tipicamente un ordine di 10^2 - 10^3 generazioni

Poiché l'algoritmo è probabilistico, diverse esecuzioni potranno portare a risultati diversi

Istruzioni probabilistiche

ALGORITMO Genetico

P = popolazione iniziale

while *< not Fine >* **do**

*P = popolazione ottenuta aggiornando la
popolazione corrente P con regole stocastiche R_1, \dots, R_m*

In genere le istruzioni probabilistiche contenute nelle **regole stocastiche** di un algoritmo genetico richiedono di estrarre a caso una alternativa, all'interno di un insieme A , in accordo ad una data distribuzione di probabilità D , non necessariamente uniforme

random(A , D)

Es., l'esecuzione di

$S = \mathbf{random} (\{0,1\} \{D(0)=0.3, D(1)=0.7\})$

ha l'effetto di porre in S il valore '0' con probabilità 0.3 o '1' con probabilità 0.7

Operatori genetici

ALGORITMO Genetico

P = popolazione iniziale

While *< not Fine >* **do**

*P = popolazione ottenuta aggiornando la
popolazione corrente P con regole stocastiche R_1, \dots, R_m*

Una volta che la funzione di fitness ha determinato il valore di bontà di ogni individuo (genotipo) della popolazione, una nuova popolazione di individui viene creata applicando le regole stocastiche o **operatori**

Gli **operatori** si ispirano alla selezione naturale e alla genetica

Gli operatori proposti da Holland sono:

- **Selezione** (ispirato alla selezione naturale)
 - **Crossover** (ispirato alla genetica)
 - **Mutazione** (ispirato alla genetica)
- } **operatori genetici**

L' operatore di selezione

Selezione: individuare gli individui (genotipi) che potranno riprodursi dando luogo a nuovi individui

La selezione naturale Darwiniana sostiene che gli individui più "forti" abbiano maggiori probabilità di sopravvivere nell'ambiente in cui vivono e quindi maggiore probabilità di riprodursi

Nel contesto degli *GAs* di Holland, gli individui più forti sono quelli con fitness più alta (infatti risolvono meglio degli altri il problema di ricerca dato)

Gli individui con fitness più alta vengono privilegiati nella fase di **selezione**: essi avranno maggiore probabilità di riprodursi

L'operatore di selezione **proporzionale**

Holland propose un metodo di selezione **proporzionale al valore di fitness**;

sia f_i il valore di fitness del genotipo g_i , la probabilità che g_i sia selezionato per la riproduzione è:

$$p_{s,i} = \frac{f_i}{\sum_{j=1}^n f_j} \quad (\text{problema di max})$$

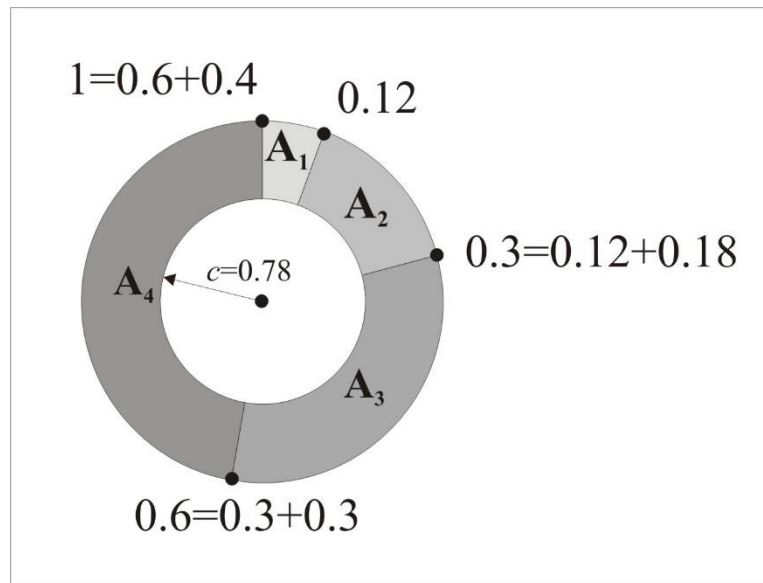
Tali probabilità sono utilizzate per costruire una sorta di **roulette**

Implementare l'operatore di selezione **proporzionale**

Ad ogni individuo x_1, \dots, x_n , con probabilità di selezione $p_{s,1}, \dots, p_{s,n}$ si associa l'intervallo $(0, p_{s,1}]$, $(p_{s,1}, p_{s,1} + p_{s,2}]$, ..., $(p_{s,1} + p_{s,2} + \dots + p_{s,n-1}, 1]$

L'operatore di selezione genera un numero casuale $c \in (0,1]$ e seleziona l'individuo il cui intervallo contiene c

Implementare l'operatore di selezione **proporzionale**



I quattro individui A_1 , A_2 , A_3 e A_4 , con probabilità di selezione 0.12, 0.18, 0.3 e 0.4, occupano uno "spicchio" di roulette di ampiezza pari alla propria probabilità di selezione

Si associa l'individuo A_1 all'intervallo $(0, 0.18]$, A_2 all'intervallo $(0.18, 0.3]$, A_3 all'intervallo $(0.3, 0.6]$, A_4 all'intervallo $(0.6, 1.0]$

L'operatore di selezione genera il numero casuale $c = 0.78$ e l'individuo A_4 viene selezionato

L'algoritmo di selezione **proporzionale**

Procedure **Select** (x_1, \dots, x_n, m)

P1 for $k = 1$ **to** m **do**
$$Y_k \leftarrow \text{random} \left(\{x_i \mid i = 1, \dots, n\}, \left\{ D(x_i) = \frac{f_i}{\sum_{j=1}^n f_j} \right\} \right)$$

end for

P2 return Y_1, \dots, Y_m

In genere la procedura $\text{Select}(x, m)$ viene richiamata con il secondo argomento pari ad n , vengono cioè selezionati per la riproduzione n individui

In tal modo si producono più copie identiche degli individui con fitness maggiore (selezione con rigenerazione)

Tali individui avranno la possibilità di riprodursi più volte

II Mating Pool

Procedure **Select** (x_1, \dots, x_n, m)

P1 for $k = 1$ **to** m **do**
$$Y_k \leftarrow \text{random} \left(\{x_i \mid i = 1, \dots, n\}, \left\{ D(x_i) = \frac{f_i}{\sum_{j=1}^n f_j} \right\} \right)$$

end for

P2 return Y_1, \dots, Y_m

- Ogni volta che un individuo della popolazione è selezionato ne viene creata una copia; tale copia è inserita nel così detto **mating pool** (piscina d'accoppiamento) **MP**
- Quando il mating pool **MP** è riempito con esattamente n copie di individui della popolazione, n nuovi discendenti vengono creati applicando gli **operatori genetici**

L' operatore di **crossover**

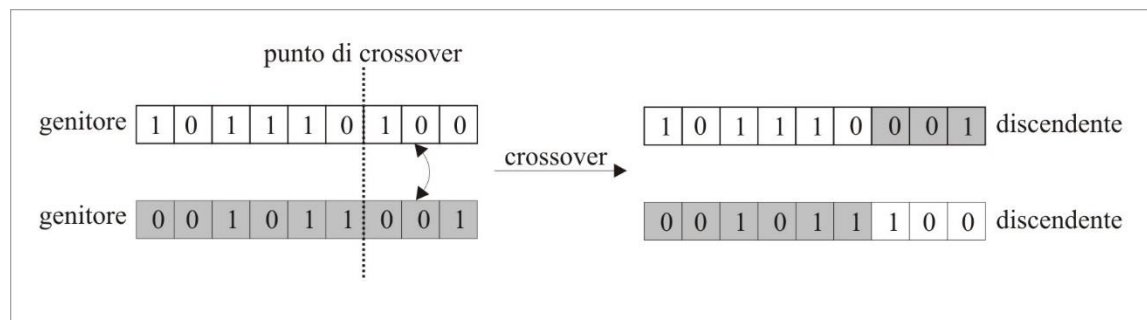
Si scelgono a caso due individui $x=x[1] \dots x[l]$ e $y=y[1] \dots y[l]$ nel mating pool (genitori) e un punto di taglio (punto di crossover) $k \in [1, l]$

Le porzioni di genotipo alla destra del punto di crossover vengono scambiate generando i due discendenti

$$x' = x[1] \dots x[k] y[k+1] \dots y[l] \quad e$$

$$y' = y[1] \dots y[k] x[k+1] \dots x[l]$$

L'operatore di crossover è applicato, in accordo a una prefissata probabilità p_c , $n/2$ volte in modo da ottenere n discendenti
nel caso in cui il crossover non sia applicato, i discendenti coincidono con i genitori



L'algoritmo di **crossover**

Procedure **Riproduci** (x_1, \dots, x_n, p_c)

P1 $MP \leftarrow \{x_1, \dots, x_n\}$

P2 for $k = 1$ **to** $n/2$ **do**

$x_1 \leftarrow \text{random}(MP)$; $MP \leftarrow MP / \{x_1\}$

$x_2 \leftarrow \text{random}(MP)$; $MP \leftarrow MP / \{x_2\}$

$c \leftarrow \text{random}(\{0, 1\}, \{D(0) = 1 - p_c, D(1) = p_c\})$

if $c = 1$ **then**

$(Y_{2k-1}, Y_{2k}) \leftarrow \text{crossover}(x_1, x_2, l)$

else

$(Y_{2k-1}, Y_{2k}) \leftarrow (x_1, x_2)$

end for

P3 return Y_1, \dots, Y_n

Procedure **crossover** (x, y, l)

P1 $k \leftarrow \text{random}(\{1, 2, \dots, l\})$

P2 return $x[1] \dots x[k] y[k+1] \dots y[l], y[1] \dots y[k] x[k+1] \dots x[l]$

L' operatore di **mutazione**

Una volta che due discendenti sono stati generati per mezzo del crossover, i valori dei bit dei nuovi individui vengono cambiati da 0 in 1 o viceversa in funzione di una prefissata (e usualmente piccola) probabilità p_m ,

Procedure **Mutazione** (x_1, \dots, x_n, p_m)

P1 for $i = 1$ **to** n **do**

$x_i \leftarrow$ **Muta** (x_i, l, p_m)

P2 return x_1, \dots, x_n

Procedure **Muta** (x, l, p_m)

P1 for $k = 1$ **to** l **do**

$x[k] \leftarrow$ **random** ($\{x_i[k], \overline{x_i[k]}\}, D(x_i[k]) = 1 - p_m, D(\overline{x_i[k]}) = p_m$)

P3 return x

L' operatore di mutazione

Come il crossover rappresenta una metafora della riproduzione sessuale, l'operatore di mutazione modella il fenomeno genetico della rara variazione di elementi del genotipo negli esseri viventi durante la riproduzione sessuale

Procedure **Muta** (x , l , p_m)

P1 for $k = 1$ to l **do**

$x[k] \leftarrow \text{random}(\{x_i[k], \overline{x_i[k]}\}, D(x_i[k]) = 1 - p_m, D(\overline{x_i[k]}) = p_m)$

P3 return x



L'Algoritmo GA

GA ($x_1, \dots, x_n, l, p_c, p_m$)

P1 $t \leftarrow 0$; $P(t) \leftarrow \{x_1, \dots, x_n\}$

P4 while $< not Fine >$ **do**

$t \leftarrow t+1$

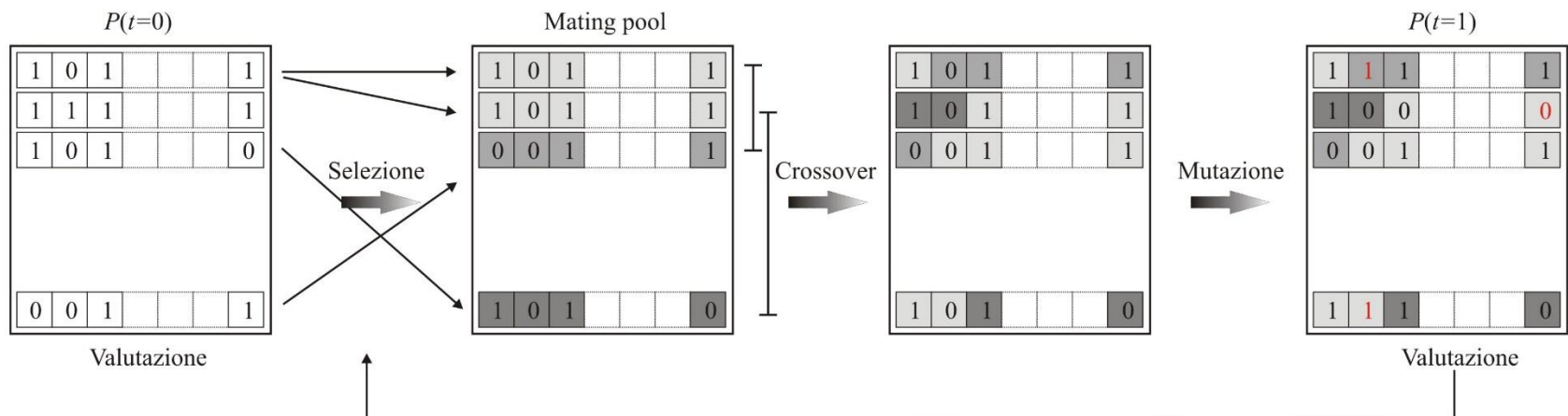
$P(t) \leftarrow \text{Select}(P(t-1), n)$

$P(t) \leftarrow \text{Riproduci}(P(t), p_c)$

$P(t) \leftarrow \text{Mutazione}(P(t), p_m)$

end while

P5 return ...



L'Algoritmo GA per l'ottimizzazione

```
GA( $x_1, \dots, x_n, l, p_c, p_m$ )
P1  $t \leftarrow 0$ ;  $P(t) \leftarrow \{x_1, \dots, x_n\}$ 
P2  $x^* \leftarrow \operatorname{argmax}\{f(x_i) \mid x_i \in P(t)\}$ ;
P3  $f^* \leftarrow \max\{f(x_i) \mid x_i \in P(t)\}$ ;
P4 while  $\langle \text{not Fine} \rangle$  do
     $t \leftarrow t + 1$ 
     $P(t) \leftarrow \text{Select}(P(t-1), n)$ 
     $P(t) \leftarrow \text{Riproduci}(P(t), p_c)$ 
    if  $\max\{f(x_i) \mid x_i \in P(t)\} > f^*$  then
         $x^* \leftarrow \operatorname{argmax}\{f(x_i) \mid x_i \in P(t)\}$ 
     $P(t) \leftarrow \text{Mutazione}(P(t), p_m)$ 
    if  $\max\{f(x_i) \mid x_i \in P(t)\} > f^*$  then
         $x^* \leftarrow \operatorname{argmax}\{f(x_i) \mid x_i \in P(t)\}$ 
    end while
P5 return  $x^*$ 
```

Algoritmi Genetici per l'ottimizzazione

Schemino riassuntivo

• Individuo	Soluzione di un problema
• Popolazione	Insieme di soluzioni
• Fitness	Qualità di una soluzione
• Cromosoma	Rappresentazione di una soluzione
• Gene	Componente di una rappresentazione
• Crossover, Mutazione	Operatori per la ricerca di soluzioni
• Selezione Naturale	Riutilizzo di buone soluzioni

Analisi dei GAs

Analisi: stimare qualche comportamento medio

(es. dopo quante generazioni si riesce a determinare la soluzione ammissibile migliore)

Anche se un algoritmo genetico è descritto da un semplice programma probabilistico, l'analisi teorica del suo comportamento è molto complessa

Discutiamo qui il primo approccio, proposto da Holland nel 1975

Analisi dei GAs

Consideriamo preliminarmente il seguente problema

E' data al tempo t una popolazione di n simboli (con ripetizione) scelti in $\{0,1\}$, ed è fissata una funzione di fitness $f : \{0,1\} \rightarrow \mathbb{N}^+$

Supponiamo che la popolazione al tempo $t+1$ sia ottenuta applicando l'operatore di **SELEZIONE**

Sapendo che la popolazione al tempo t contiene $m(t)$ volte 1, qual è il valore atteso $E[m(t+1)]$ del numero di 1 al tempo $t+1$?

Analisi dei GAs

Sapendo che la popolazione al tempo t contiene $m(t)$ volte 1, qual è il valore atteso $E[m(t+1)]$ del numero di 1 al tempo $t+1$?

La probabilità di scegliere '1' al tempo ' $t+1$ ' è

$$\sum_{x_i=1} p_{s,i} = \sum_{x_i=1} \frac{f_i(1)}{\sum_{j=1}^n f_j} = \frac{f(1) \cdot m(t)}{f(1) \cdot m(t) + f(0) \cdot (n - m(t))}$$

quindi

$$E[m(t+1)] = \sum_{i=1}^n \frac{\sum_{x_i=1} f_i(1)}{\sum_{j=1}^n f_j} \cdot 1 = n \cdot \frac{f(1) \cdot m(t)}{f(1) \cdot m(t) + f(0) \cdot (n - m(t))}$$

Analisi dei GAs

Sapendo che la popolazione al tempo t contiene $m(t)$ volte 1, qual è il valore atteso $E[m(t+1)]$ del numero di 1 al tempo $t+1$?

$$E[m(t+1)] = \sum_{i=1}^n \frac{\sum_{x_i=1} f_i(1)}{\sum_{j=1}^n f_j} \cdot 1 = n \cdot \frac{f(1) \cdot m(t)}{f(1) \cdot m(t) + f(0) \cdot (n - m(t))}$$

Osservando che la fitness media al tempo t $\bar{f}(t)$ vale $\frac{1}{n} \sum_{j=1}^n f_j$

ricaviamo $E[m(t+1)] = \frac{f(1)}{\bar{f}(t)} \cdot m(t)$

Osservazioni

Se $f(1) \gg \bar{f}(t)$, allora il numero di '1' al tempo ' $t+1$ ' sarà verosimilmente aumentato con un buon fattore moltiplicativo

Se $f(1) \ll \bar{f}(t)$, allora il numero di '1' al tempo ' $t+1$ ' sarà verosimilmente considerevolmente diminuito

Analisi dei GAs

La teoria di Holland si riferisce ad algoritmi genetici, con popolazioni di n cromosomi (w_1, \dots, w_n) , dove ogni cromosoma è una parola di l bits, $x[1] \dots x[l]$, e la fitness è una funzione $f : \{0,1\}^l \rightarrow \mathbb{N}^+$

Definizione Chiamiamo *Schema* l'insieme di parole in cui sono fissati gli alleli di alcuni geni

Lo schema viene descritto da una parola di lunghezza ' l ' su $\{0,1,*\}$, dove il simbolo '*' sta per '0' o '1'.

Es. Lo schema $0*1*$ individua le parole che hanno '0' al primo posto e '1' al terzo posto:

0010, 0011, 0110, 0111.

Analisi dei GAs

Fissiamo ora uno schema H ; sia data una popolazione (w_1, \dots, w_n) al tempo t e sia $m(H, t)$ il numero di parole H presenti nella popolazione.

Qual è il numero atteso $E[m(H, t+1)]$ di parole di H nella popolazione al tempo $t+1$, se si è applicata una operazione di *Selezione, CrossOver, Mutazione*?

Analisi dei GAs

Selezione

La probabilità che venga selezionata una parola di H è

$$\frac{\sum_{w \in H} f(w_j)}{\sum_j f(w_j)} = \frac{\bar{f}(H, t) \cdot m(H, t)}{\bar{f}(t) \cdot n} \quad \text{dove}$$

$$\bar{f}(t) = \frac{\sum_j f(w_j)}{n} \quad \text{è la fitness media della popolazione al tempo } t$$

$$\bar{f}(H, t) = \frac{\sum_{w_j \in H} f(w_j)}{m(H, t)} \quad \text{è la fitness media dello schema nella popolazione al tempo } t$$

Analisi dei GAs

Selezione

La probabilità che venga selezionata una parola di H è

$$\frac{\sum_{w \in H} f(w_j)}{\sum_j f(w_j)} = \frac{\bar{f}(H, t) \cdot m(H, t)}{\bar{f}(t) \cdot n} \quad \text{quindi}$$

$$\begin{aligned} E[m(H, t+1)] &= \sum_{i=1}^n \frac{\sum_{w_j \in H} f(w_j)}{\sum_{j=1}^n f_j} \cdot 1 = n \cdot \frac{\sum_{w_j \in H} f(w_j)}{\sum_{j=1}^n f_j} \\ &= \frac{\bar{f}(H, t) \cdot m(H, t)}{\bar{f}(t)} \end{aligned}$$

Analisi dei GAs

Selezione

$$E[m(H, t+1)] = \frac{\bar{f}(H, t) \cdot m(H, t)}{\bar{f}(t)}$$

Osservazioni

Se $\bar{f}(H, t) \gg \bar{f}(t)$, allora il numero di parole dello schema al tempo ' $t+1$ ' sarà verosimilmente aumentato con un fattore moltiplicativo

Se $\bar{f}(H, t) \ll \bar{f}(t)$, allora il numero di parole dello schema al tempo ' $t+1$ ' sarà verosimilmente considerevolmente diminuito

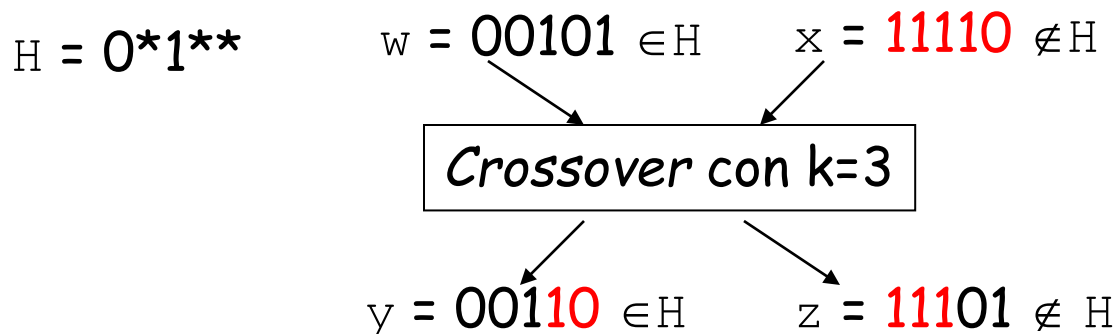
Analisi dei GAs

Crossover

Dato uno schema $H \in \{0,1,*\}^+$, sia 'w' una parola dello schema

L'applicazione dell'operazione di *Crossover* può trasformare la parola w dello schema in una parola w' non appartenente allo schema: diremo che in questo caso lo schema viene distrutto

Lo schema **non** viene distrutto se nell'applicazione del *Crossover* viene selezionata in w una posizione k che ha alla sua sinistra (o alla sua destra) solo simboli *



Analisi dei GAs

Crossover

Definizione Chiamiamo *lunghezza definita* di uno schema 'H' la distanza $\delta(H)$ tra i simboli '0' o '1' più esterni misurata come
$$\text{pos}(\text{elem}_{dx}) - \text{pos}(\text{elem}_{sx})$$

Es. $\delta(*0*1**) = 4 - 2 = 2$

Indicando con p_c la probabilità che l'algoritmo genetico decida di applicare l'operazione di *Crossover*, risulta

$$\Pr(\text{Crossover applicato, schema } H \text{ distrutto}) \leq P_c \cdot \frac{\delta(H)}{l-1}$$

quindi

$$\Pr(\text{schema } H \text{ non distrutto}) \geq 1 - P_c \cdot \frac{\delta(H)}{l-1}$$

Analisi dei GAs

Mutazione

Dato uno schema $H \in \{0,1,*\}^+$, sia w una parola dello schema

L'applicazione dell'operatore di *Mutazione* può trasformare la parola w dello schema in una parola w' non appartenente allo schema qualora esso modifichi i simboli che nello schema corrispondono a 0 o 1 (non ad $*$)

Definizione Chiamiamo *ordine* di uno schema H il numero $\text{ord}(H)$ di simboli '0' o '1'

Es. $\text{ord}(**0*1*) = 2$

Indicando con p_m la probabilità di mutazione, risulta

$$\mathbf{Pr}(\text{schema } H \text{ non distrutto}) = (1 - p_m)^{\text{ord}(H)}$$

Analisi dei GAs

Teorema dello Schema (Holland '75)

Se in una popolazione, che al tempo t ha $m(H, t)$ parole in uno schema H , vengono applicate le operazioni di *SELEZIONE*, *CROSSOVER*, *MUTAZIONE*, il numero atteso $E[m(H, t+1)]$ di parole nello schema H alla generazione successiva verifica:

$$E[m(H, t+1)] \geq \frac{\bar{f}(H, t)}{\bar{f}(t)} \cdot \left(1 - P_{CROSS} \cdot \frac{\delta(H)}{l-1}\right) \cdot (1 - P_{MUT})^{ord(H)} \cdot m(H, t)$$

dove $\bar{f}(t)$ è la fitness media della popolazione, mentre $\bar{f}(H, t)$ è la fitness media delle parole di H nella popolazione

Problema della codifica

Affinché risultino efficaci gli operatori genetici devono tener conto della struttura dello spazio delle soluzioni

Codifica di Gray:

- I numeri interi da 0 a 2^n-1 sono codificati con una stringa binaria di lunghezza n .
- Interi consecutivi hanno una rappresentazione che differisce per un solo bit

Es.

0 000

1 001

2 011

3 010

4 110

5 111

6 101

7 100

Vantaggio:

- Una qualsiasi inversione di un bit produce piccoli cambiamenti

Svantaggio:

- Quando il cambiamento è grande è maggiore che con una normale codifica binaria

Problema della codifica

Es. TSP

la struttura **naturale** dello spazio delle soluzioni è definita dallo spazio delle permutazioni di n elementi e

non dallo spazio delle stringhe binarie (anche se l'insieme di vettori caratteristici degli insiemi di lati che formano cicli hamiltoniani in un grafo dato è un sottoinsieme dello spazio delle stringhe binarie)

E' quindi necessario adattare gli operatori genetici (crossover e mutazione) in modo che possano (sempre) produrre soluzioni ammissibili del problema

Esercizio di codifica

Trovare un operatore di crossover che prese due permutazioni π_1 e π_2 produca due permutazioni figlie

Trovare un operatore di mutazione che presa una permutazione π produca una permutazione mutata

Problema della selezione

Selezione proporzionale

$$p_{s,i} = \frac{f_i}{\sum_{j=1}^n f_j}$$

- *Convergenza prematura*
 - Se un individuo i ha fitness molto maggiore della media della popolazione ma molto minore della massima fitness possibile, tenderà ad essere sempre selezionato e quindi a generare una popolazione "mediocre"
- *Stagnazione*
 - Dopo un certo numero di generazioni, tutti gli individui hanno una buona fitness e quindi tendono ad avere la stessa probabilità di essere selezionati

Problema della selezione

Selezione per rango (rank selection)

- Si ordinano gli individui in ordine decrescente di fitness
- si attribuisce una probabilità decrescente (ad es. linearmente o esponenzialmente) in funzione della posizione in classifica, indipendentemente dal valore della fitness

Vantaggi

- Non si ha convergenza prematura (nessun individuo ha probabilità molto maggiore degli altri di essere selezionato)
- Non c'è stagnazione: le probabilità sono sempre ugualmente distribuite

Svantaggi

- Computazionalmente pesante
- Nota: non è biologicamente plausibile (...but who care?)

Problema della selezione

Selezione tramite torneo (tournament selection)

- Per ogni individuo da selezionare, si seleziona un gruppo di individui e si clona il migliore (torneo)

Vantaggi

- Non si ha convergenza prematura (se il gruppo è ristretto nessun individuo ha probabilità molto maggiore degli altri di essere selezionato)
- Non c'è stagnazione: le probabilità sono sempre ugualmente distribuite

Problema della selezione

Selezione elitista (elitist selection)

- Almeno una copia dell'individuo migliore viene mantenuta nella generazione successiva.

Vantaggi

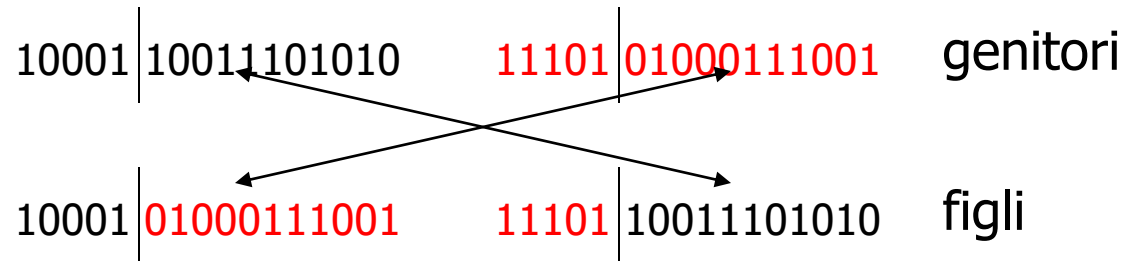
- Non si perdono soluzioni buone nei processi "casuali" di selezione

Svantaggi

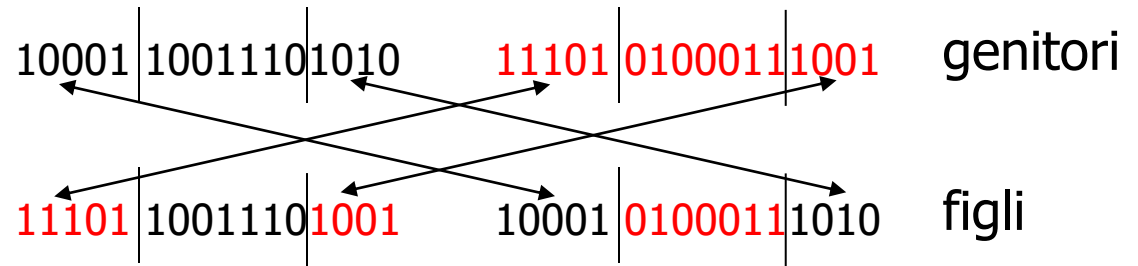
- Si possono raggiungere massimi locali se i caratteri di tale individuo diventano "dominanti"

Tipi di crossover

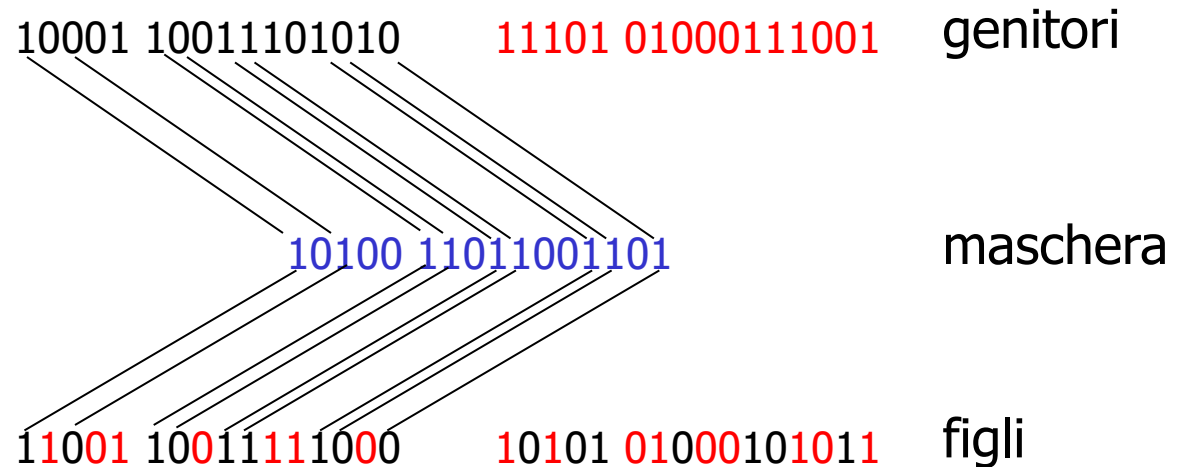
Singolo punto



Doppio punto



Uniforme



Condizione di arresto


Il processo viene arrestato:

- quando la fitness media cresce al di sotto di una soglia (quando, cioè, tutti gli individui sono simili tra loro)
- dopo un numero definito di iterazioni

GOLEM: Generator Of Life Embedded into MMOs

- MMOs: complex and costly system from both a game design and a technical point of view:
 - They have many open issues, among which **gameplay too repetitive**
- **Goal:** incrementing players thrill by supplying plenty of unpredictable monsters



- 
- **Main idea:** exploit GAs to make monsters evolve and reproduce in unforeseen species

A. Guarneri, D. Maggiorini, L.A. Ripamonti, M. Trubian, "GOLEM: Generator Of Life Embedded into MMOs". In Proceedings of the Twelfth European Conference on the Synthesis and Simulation of Living Systems: Advances in Artificial Life, ECAL, pp. 585–592, 2013

GOLEM algorithm

- For each generation (till max.):
 1. **Selection:**
 - Death by old age?
 - Selects candidates and evaluates «challenge ratings» (low/medium/high/invincible)
 2. **Crossover:**
 - Uniform crossover
 3. **Mutation:**
 - Random change of ONE gene (90% probability to happen!!)
 4. **Evaluation**
 - Are whelps suited for their habitat? (death/malus/ etc.)
 5. **(ending)**



Esercizio

Istanza: la funzione $f(x) \equiv y = x$

Questione: trova $\max f(x)$ con $x \in [0,255]$

Scrivere un *GA* che risolve il problema

Scegliere

- la dimensione della popolazione
- la lunghezza del genotipo
- la funzione di fitness
- la probabilità di crossover
- la probabilità di mutazione

GAs e WWW

Librerie "free" per *Gas*

PGAPack: <https://github.com/eddelbuettel/pgapack/blob/master/README>

Galib: <http://lancet.mit.edu/ga/>

- implementa i *GAs* di Holland e altri modelli proposti successivamente
- gira sotto Linux
- è presente una versione parallela dei *GAs*

Conclusioni

I *GAs* hanno ricevuto considerevole attenzione come algoritmi di ricerca,

hanno applicazione in varie aree:

- ottimizzazione
- apprendimento automatico
- simulazione nella genetica delle popolazioni
- problematiche di cooperazione in sistemi multi-agenti
- inferenza: generazione automatica di regole logiche