



Introduzione agli Algoritmi Genetici

Prof. Pedro Pablo González Pérez

Departamento de Matemáticas Aplicadas y Sistemas
Universidad Autónoma Metropolitana, México D.F.

pgonzalez@correo.cua.uam.mx

Cosa sono gli Algoritmi Genetici (GA)? (I)

- Sono algoritmi di ricerca ispirati ai meccanismi della selezione naturale e della genetica naturale.
- Lavorano su una popolazione di stringhe (elementi, creature artificiali, componenti, ...) combinando la sopravvivenza dei più adatti con uno scambio di informazione aleatorio e strutturato.
- Una nuova generazione di stringhe viene creata usando bits e blocchi dai più adatti della vecchia generazione.
- Combinano meccanismi aleatori con informazione accumulata per migliorare l'esecuzione.

Cosa sono gli Algoritmi Genetici (GA)? (II)

- Sono stati sviluppati da John Holland, i suoi colleghi e i suoi studenti alla University of Michigan.
- Sono stati concepiti per: (1) astrarre e spiegare i processi adattativi dei sistemi naturali e (2) costruire sistemi artificiali che includono importanti meccanismi dei sistemi naturali.
- Offrono un bilancio fra l'efficienza (buon funzionamento) e l'efficacia (capacità di ottenere l'effetto desiderato) necessarie per la sopravvivenza in diversi ambienti.
- Forniscono una ricerca robusta in uno spazio complesso.

Principali aree generiche di applicazione

- Ricerca in uno spazio di soluzioni
- Ottimizzazione
- Apprendimento
- Programmazione genetica

Caratteristiche essenziali degli algoritmi genetici

- Lavorano con **una codifica dell'insieme di parametri**, non lavorano con i propri parametri.
- Eseguono una ricerca fra una popolazione di elementi.
- Usano come informazione una **funzione obiettivo**, non usano nessun'altra conoscenza ausiliaria.
- Applicano regole di transizione probabilistiche, non deterministiche.

Alcune funzioni obiettivo conosciute nell'argomento "Reti Neurali Artificiali (RNA)"

- Somma pesata: $\sum W_{ij} * X_i$
- Distanza: $d(W, X)$
- Funzione sigmoide: $1/(1 + e^{-A})$
- $E_p = (1/2) \sum_j (D_{jp} - O_{jp})^2$

Un primo esempio preso dalle RNA (I)

La funzione obiettivo

Considerare la seguente funzione obiettivo, che desideriamo massimizzare:

$$A = -W_1 * X_1 + W_2 * X_2 - W_3 * X_3 + W_4 * X_4 + W_5 * X_5 - 70$$

dove $X, W \in \mathbb{R}^5$, il pattern di ingresso X ha i valori fissi $X = (2, 4, 1, 3, 2)$ e W_i prende valori interi nell'intervallo $[0, 31]$.

- Dove avevamo già visto questa funzione?
- Quale potrebbe essere una conseguenza di massimizzare la funzione A ?
- Quale potrebbe essere una conseguenza di minimizzare la funzione A ?

Un primo esempio preso dalle RNA (II)

La popolazione

- Generare vettori aleatori $W = (W_1, W_2, W_3, W_4, W_5)$ dove la codifica dell'insieme di parametri sarebbe $WStringa = W_1Stringa + W_2Stringa + W_3Stringa + W_4Stringa + W_5Stringa$.
- Per codificare ogni W_i occorre una stringa binaria di lunghezza pari a 5 (ad esempio, a $W_i = 25$ corrisponde la $W_iStringa = 11001$).
- La popolazione verrà composta da stringhe di lunghezza pari a 25, dove il primo sottoinsieme di lunghezza 5 corrisponde alla codifica del parametro W_1 , il secondo sottoinsieme di lunghezza 5 alla codifica del parametro W_2 e così via.

Un primo esempio preso dalle RNA (III)

La popolazione iniziale di stringhe e i valori di fitness

No.	Stringa	Valori	Fitness
1	1001101110001110101110101	19 14 7 11 21	16
2	1101101011101010111110001	27 11 21 15 17	-22
3	0010101101011011010001101	5 13 13 20 13	45
4	1110110111001001100110110	29 23 4 25 22	33
5	0110100110101010101100110	13 6 21 11 6	-31
6	1110001101110000011110110	28 13 24 7 22	-59
7	0100110001011111011110111	9 17 15 23 23	80
8	0111010011011010111101100	14 19 13 15 12	34

Gli Operatori di un Algoritmo Genetico Semplice

- Selezione (*selection*)
- Incrocio (*crossover*)
- Mutazione (*mutation*)

Selezione (I)

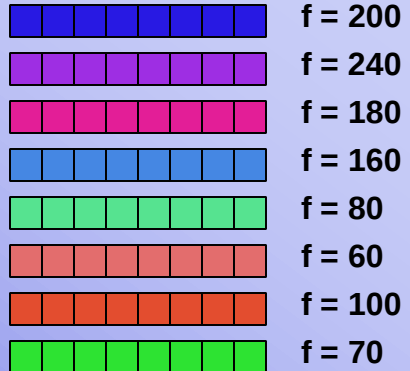
- Le stringhe individuali sono copiate considerando i loro valori della **funzione obiettivo f** (*fitness function*).
- Le stringhe con maggiore valore della funzione obiettivo hanno una maggiore probabilità di contribuire con uno o più discendenti nella prossima generazione.
- L'operatore di selezione costituisce una versione artificiale della selezione naturale, e potrebbe essere visto come una sopravvivenza Darwiniana se le stringhe fossero considerate come "creature". Ci sono molti modi di implementare algoritmicamente questo operatore.

Selezione (II)

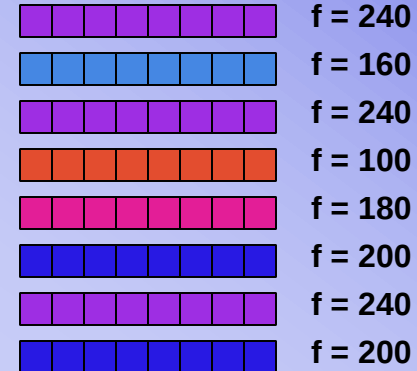
- La probabilità che una stringa sia scelta per l'operatore di riproduzione è $P = M/N$, dove M è il valore della funzione obiettivo per la stringa e N è la somma dei valori della funzione obiettivo su tutte le stringhe della popolazione (generazione attuale).
- Per ogni stringa scelta dall'operatore di riproduzione, una copia viene fatta. L'insieme di tutte le copie costituirà una nuova popolazione sulla quale verranno eseguite le azioni degli altri operatori.

Selezione (III)

Generazione t



Operatore di
selezione



Un primo esempio preso dalle RNA (IV)

La selezione

No.	Stringa	Valori	Fitness	AFitness	AFitness/ FitnessT
1	1001101110001110101110101	19 14 7 11 21	16	96	13.04%
2	1101101011101010111110001	27 11 21 15 17	-22	58	7.88%
3	0010101101011011010001101	5 13 13 20 13	45	125	16.98%
4	1110110111001001100110110	29 23 4 25 22	33	113	15.35%
5	0110100110101010101100110	13 6 21 11 6	-31	49	6.65%
6	1110001101110000011110110	28 13 24 7 22	-59	21	2.85%
7	0100110001011111011110111	9 17 15 23 23	80	160	21.73%
8	0111010011011010111101100	14 19 13 15 12	34	114	15.48%
	Total			736	≈ 100%

Un primo esempio preso dalle RNA (V)

La selezione: il metodo della *roulette pesata*

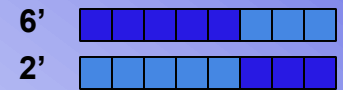
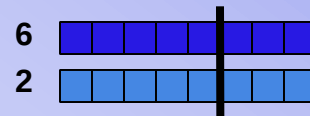
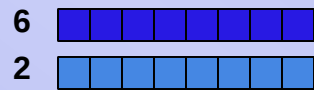
No.	Stringa	Fitness	AFitness
3	0010101101011011010001101	45	125
7	0100110001011111011110111	80	160
1	1001101110001110101110101	16	96
7	0100110001011111011110111	80	160
4	1110110111001001100110110	33	113
2	1101101011101010111110001	-22	58
8	0111010011011010111101100	34	114
2	1101101011101010111110001	-22	58

Incrocio (I)

- Dopo l'applicazione dell'operatore di selezione, l'operatore di incrocio esegue due passi: (1) la generazione aleatoria di coppie e (2) la scelta aleatoria del sito di incrocio per ogni coppia e l'esecuzione dell'incrocio.
- In (2) una posizione intera aleatoria k è scelta fra 1 e la lunghezza della stringa meno 1 $[1, l-1]$. Due nuove stringhe vengono create scambiando tutti i caratteri fra le posizioni $k+1$ e l compresa.

Incrocio (II)

1		$f = 240$
2		$f = 160$
3		$f = 240$
4		$f = 100$
5		$f = 180$
6		$f = 200$
7		$f = 240$
8		$f = 200$



lunghezza della stringa: $l = 8$

sito di incrocio: $k = 5$

Un primo esempio preso dalle RNA (VI)

Incrocio

No.	Selected chromosome	Fitness	AFitness
4	1110110111001001100110110	33	113
7	0100110001011111011110111	80	160

Crossover point = 10

4: 1110110111001001100110110

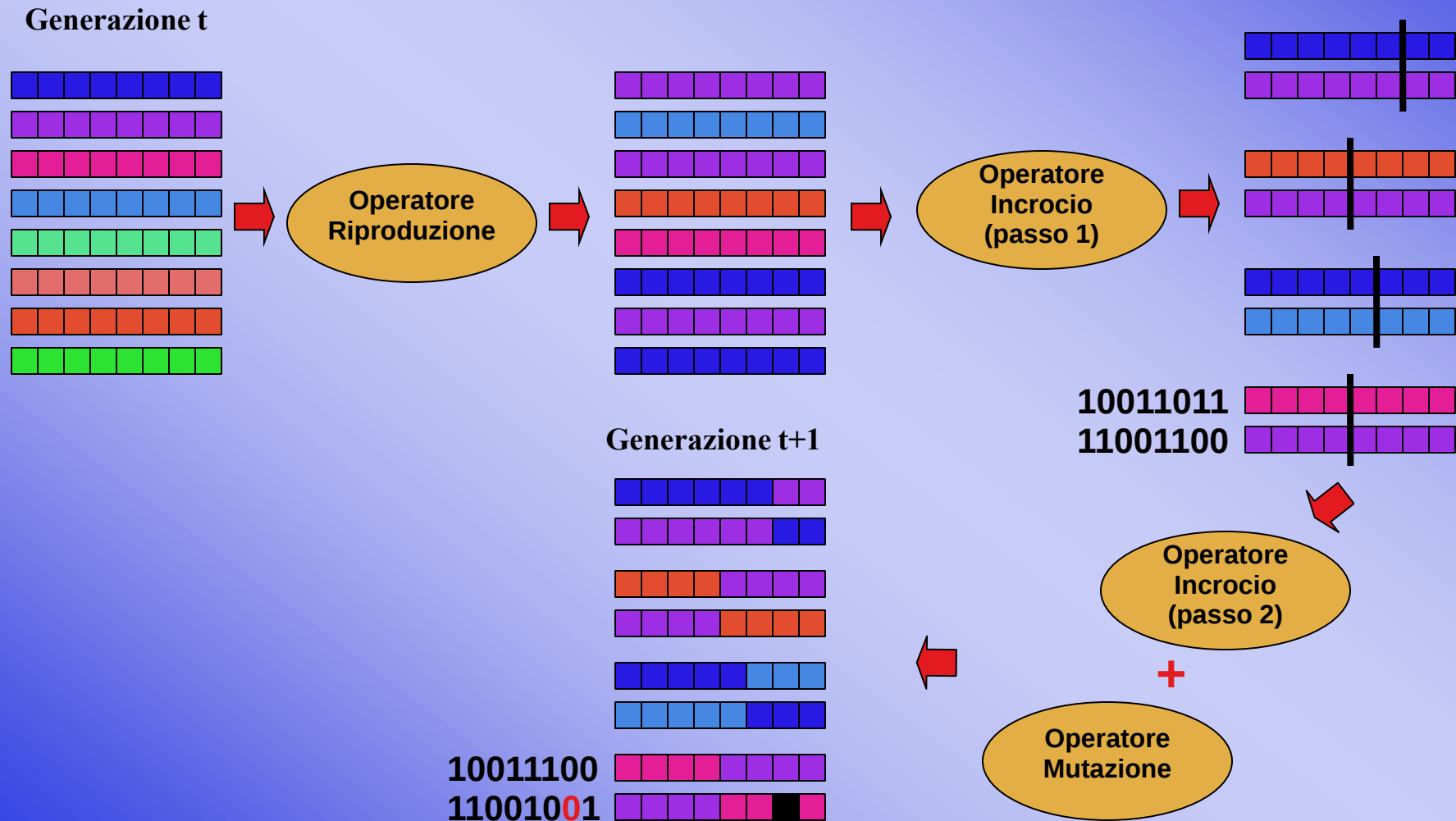
7: 0100110001011111011110111

No.	Offsprings	Fitness	AFitness
4'	1110110111011111011110111		
7'	0100110001001001100110110		

Mutazione

- La mutazione è l'alterazione aleatoria del valore di una unica posizione della stringa con probabilità p_m .
- La probabilità di mutazione p_m è un valore molto basso. La frequenza di mutazione potrebbe essere scelta nell'ordine di una mutazione per mille caratteri trasferiti ($p_m = 0.001$).
- In un GA l'operatore di mutazione viene invocato dall'operatore di incrocio quando si inizia il trasferimento di caratteri dalle stringhe genitori alle stringhe figlie.

Gli Operatori di un Algoritmo Genetico Semplice



Funzionamento di un algoritmo genetico (I)

The genetic algorithm creates a population of chromosomes and then applies the selection, crossover and mutation operators to the strings in the current population to generate a new population. A simple GA works as shown in figure above and its algorithm is as follows:

1. Encoding of the problem in a binary string
2. Random generation of a population of n strings of length n (where each string or chromosome represents a candidate solution to the problem)
3. Calculate the value of fitness function $f(x)$ for each string x in the population
4. Repeat the following steps until n offsprings have been created:

Funzionamento di un algoritmo genetico (II)

- Selection of the pair of parents strings from the current population according to their share in the population global fitness
- With crossover probability p_c , crossover the pair of selected strings at a randomly crossover point to form two offsprings
- With mutation probability p_m , mutate the two offspring
- Copy the two new strings in the population

5. Replace the old population with the new population.
6. Go to 3.

Un Paragone fra la terminologia genetica e la terminologia GA

Genetica	Algoritmo Genetico
Cromosoma	Stringa
Gene	Caratteristica, carattere o rivelatore
Allele	Valore della caratteristica
Locus	Posizione nella stringa
Genotipo	Struttura
Fenotipo	Struttura decodificata

Stringhe e sotto-stringhe come idee e nozioni rispettivamente (I)

- Ogni stringa in una popolazione potrebbe essere considerata come una “idea” o prescrizione per eseguire un compito particolare.
- Le sotto-stringhe dentro ogni stringa contengono diverse “nozioni” su che cosa è importante per il compito da eseguire.
- In questo senso, la popolazione contiene non solo un insieme di n “idee”, ma molto più di ciò: contiene un gran numero di “nozioni” per eseguire il compito.

Stringhe e sotto-stringhe come idee e nozioni rispettivamente (II)

- Gli algoritmi genetici approfittano di questa ricchezza di informazione quando: (1) riproducono nozioni di grande qualità in accordo con la loro esecuzione e (2) incrociano queste nozioni con altre nozioni adeguate che provengono da altre stringhe.
- In questo senso, quale sarebbe un'idea innovativa?

Schemi (I)

- Quando una stringa può essere considerata una "stringa rappresentativa" di altre classi di stringhe?
- E' necessaria l'esistenza di similitudini in certe posizioni della stringa.
- Uno schema è un pattern di similitudine che descrive un sottoinsieme di stringhe con analogie in certe posizioni della stringa.
- Considerare l'alfabeto binario $\{0,1\}$ e il corrispondente alfabeto esteso $\{0,1,*\}$, dove il simbolo $*$ è un simbolo jolly (può valere indifferentemente 0 o 1).
- Adesso possiamo creare stringhe (schemi) dall'alfabeto ternario $\{0,1,*\}$.

Schemi (II)

- Il significato dello schema si traduce in un *pattern matching device*.
- Fra uno schema e una stringa particolare c'è una corrispondenza se in tutte le posizioni nello schema un **1** corrisponde a un **1** nella stringa, un **0** corrisponde a un **0**, e un ***** corrisponde a **qualsiasi degli altri due elementi**.
- Per esempio, lo schema **1*10*1** descrive il sottoinsieme con 4 elementi **{101001, 101011, 111001, 111011}**.
- Il simbolo ***** è solo un meta-simbolo cioè, un simbolo sugli altri simboli. Questo non è mai processato dal GA.

Schemi (III)

- Per un alfabeto di cardinalità (numero di caratteri nell'alfabeto) k e stringhe con lunghezza l , ci sono $(k+1)^l$ schemi.
- Una stringa particolare contiene 2^l schemi. Una popolazione con n elementi contiene fra 2^l e $n2^l$ schemi, dipendendo dalla diversità della popolazione.
- L'insieme composto dalle stringhe, dai valori della funzione obiettivo (*fitness values*) e dalle similitudini fra le stringhe nella popolazione fornisce una ricchezza di nuova informazione che contribuisce a guidare la ricerca sullo spazio degli elementi.

Schemi (IV)

- Fra i 2^l e $n2^l$ schemi contenuti in una popolazione, quanti sono realmente processati in modo utile dall'algoritmo genetico?
- Quali sono gli effetti della riproduzione, dell'incrocio e della mutazione sulla crescita o perdita degli schemi importanti da una generazione a un'altra?
- Secondo l'operatore di selezione le stringhe più adeguate hanno una maggiore probabilità di essere selezionate come parte della nuova generazione.
Allora, cosa dovrebbe capitare con i patterns di similitudine?

Schemi (V)

- Cosa accade a uno schema particolare quando l'operatore di incrocio viene applicato?
- L'operatore di incrocio potrebbe tagliare lo schema o lasciarlo intatto, a seconda dove si trova il punto di incrocio.
- Quali di questi due schemi ha una maggiore probabilità di sopravvivere all'operatore di incrocio, $1^{***}0$ o $^{**}11^{*}$?
- Gli schemi con lunghezza di definizione corta hanno maggiore probabilità di rimanere intatti dopo l'azione dell'operatore di incrocio.
- La mutazione, come conseguenza della sua bassa probabilità, non corrompe molto spesso uno schema particolare.

Fondamenti Matematici (I)

- Si riprende il concetto di schema o *pattern* di similitudine.
- Gli schemi con lunghezza di definizione corta sono visti come **blocchetti di costruzione** (*building blocks*).
- Gli algoritmi genetici approfittano in parallelo della grande quantità di similitudini contenuta nei blocchetti di costruzione.
- Come contare gli schemi rappresentati dentro una popolazione di stringhe?
- Quale sarà la crescita o la perdita degli schemi in una generazione particolare?
- Quali sono gli effetti degli operatori di selezione, incrocio e mutazione su uno schema particolare?

Fondamenti Matematici (II)

Gli effetti della selezione, incrocio e mutazione sugli schemi

- Consideriamo che le stringhe vengano costruite su un alfabeto binario $\{0,1\}$.
- L'elemento $A = a_1a_2a_3\dots a_l$ rappresenta una stringa A con l caratteri (geni) a_i , dove ognuno può prendere un valore di 0 o 1.
- $A(t)$ rappresenta una popolazione di stringhe A_j , $j = 1, 2, \dots, n$ nel tempo o generazione t
- Consideriamo uno schema H preso da un alfabeto ternario $V = \{0,1,*\}$.

Fondamenti Matematici (III)

Gli effetti della selezione, incrocio e mutazione sugli schemi

- Una stringa particolare A_k membro di una popolazione $A(t)$ costituisce un esempio di uno schema H se il valore di ogni caratteristica a_i ($i = 1, 2, \dots, l$) nella stringa corrisponde al valore di ogni posizione h_i nello schema.
- Non tutti gli schemi vengono creati allo stesso, ci sono schemi più specifici degli altri.
- L'ordine (o grado) di uno schema H , denotato da $o(H)$, è il numero di posizioni fisse (in un alfabeto binario, sarebbe il numero di 0 e 1) presenti nel *template*.

Fondamenti Matematici (IV)

Gli effetti della selezione, incrocio e mutazione sugli schemi

- La lunghezza di uno schema H , denotata da $\delta(H)$ è la distanza fra la prima e l'ultima posizione fissa della stringa. Uno schema con una sola posizione fissa ha un $\delta(H) = 0$.
- L'effetto della selezione sulla crescita o perdita di uno schema in una popolazione viene determinato dall'equazione:

$$m(H, t+1) = m(H, t)f(H)/f'$$

dove: $m(H, t)$ sono gli m esempi di uno schema particolare H contenuti in una popolazione $A(t)$; $f(H)$ è il fitness medio delle stringhe che rappresentano lo schema H nella generazione t , e f' è il fitness medio di tutta la popolazione, $f' = \sum f_j/n$.

Fondamenti Matematici (V)

Gli effetti della selezione, incrocio e mutazione sugli schemi

- Uno schema particolare cresce così come cresce la relazione $f(H)/f'$.
- Uno schema con valori di *fitness* sopra la media della popolazione riceverà un aumento nel numero di stringhe che rappresentano lo schema nella prossima generazione.
- Uno schema con valori di *fitness* sotto la media della popolazione riceverà un decremento nel numero di stringhe che rappresentano lo schema nella prossima generazione.

Fondamenti Matematici (VI)

Gli effetti della selezione, incrocio e mutazione sugli schemi

- Uno schema H viene distrutto da un operatore di incrocio con una probabilità $p_d = \delta(H)/(l - 1)$ e sopravvive con una probabilità $p_s = 1 - p_d$
- Uno schema sopravvive quando il sito di incrocio cade al di fuori della lunghezza di definizione.
- Se l'incrocio fosse eseguito da una scelta aleatoria, cioè con una probabilità p_c in una coppia particolare, la probabilità di sopravvivenza potrebbe venire determinata dalla disequazione: $p_s \geq 1 - p_c \delta(H)/(l - 1)$

Fondamenti Matematici (VII)

Gli effetti della selezione, incrocio e mutazione sugli schemi

- Quando si assume l'indipendenza fra le operazioni di riproduzione e incrocio, l'effetto combinato di tutti e due gli operatori su uno schema viene determinato dall'equazione:

$$m(H, t+1) = m(H, t)f(H)/f'[1 - p_c \delta(H)/(l - 1)]$$

- Il valore di una caratteristica in una stringa sopravvive con una probabilità $1 - p_m$, mentre uno schema sopravvive quando ognuna delle $o(H)$ posizioni fisse nella stringa sopravvive.

Fondamenti Matematici (VIII)

Gli effetti della selezione, incrocio e mutazione sugli schemi

- La probabilità di sopravvivenza alla mutazione per tutto lo schema di ordine $o(H)$ sarebbe $(1 - p_m)^{o(H)}$
- Per valori molto piccoli di p_m ($p_m \ll 1$), la probabilità di sopravvivenza dello schema può essere approssimata dall'espressione $1 - o(H)p_m$

Apprendimento con Algoritmi Genetici (I)

Sistemi classificatori

Un sistema classificatore (SC) è un sistema di apprendimento automatico che usa un algoritmo genetico per imparare le regole che gli permettono guidare la sua esecuzione in un ambiente arbitrario. Un SC è composto da tre componenti principali:

- Un sistema di regole e messaggi (sistema di produzione). Le regole sono conosciute come classificatori.
- Un pagamento o compensazione da un sistema di credito.
- Un algoritmo genetico.

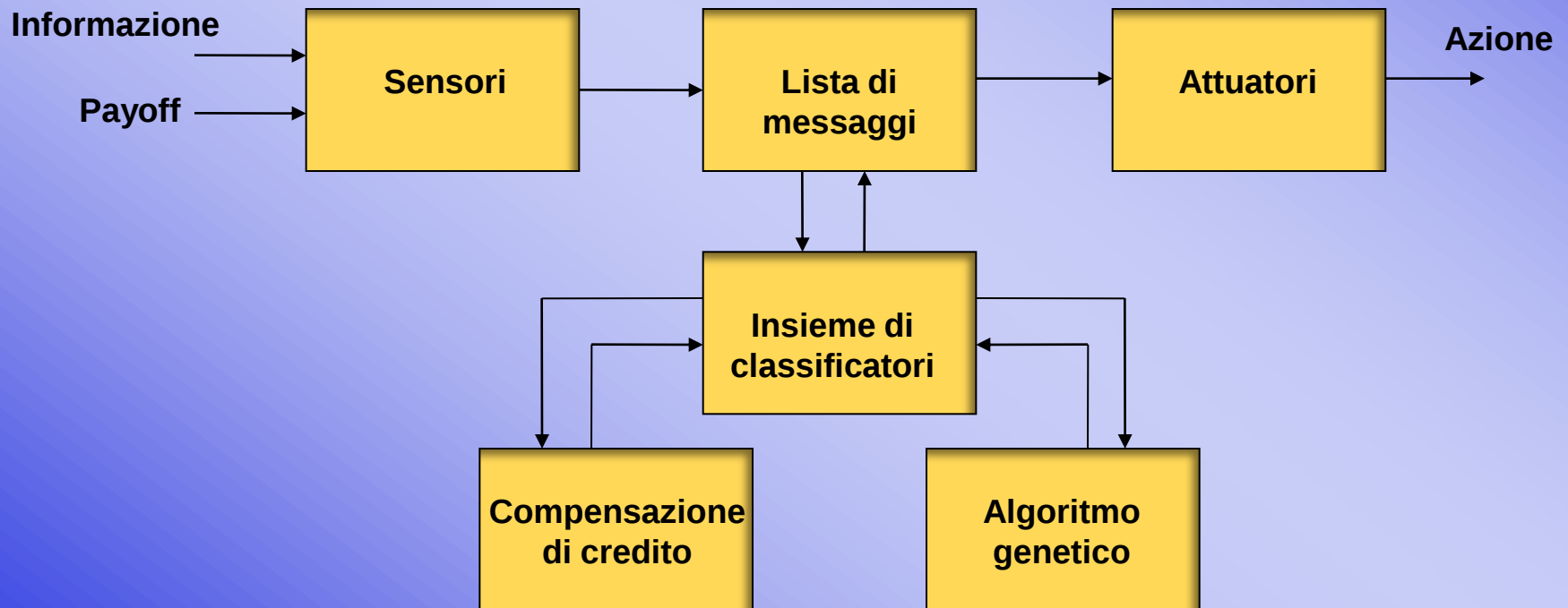
Apprendimento con Algoritmi Genetici (II)

Componenti di un sistema esperto e la sua corrispondenza con i componenti di un sistema classificatore semplice

Sistema esperto	Sistema classificatore
Interfaccia di entrata-uscita	Sensori-attuatori
Memoria di lavoro	Lista di messaggi
Base di Conoscenza (regole di produzione)	Insieme di classificatori
Meccanismo di inferenza	Pagamento o compensazione da un sistema di credito
Modulo di apprendimento	Algoritmo genetico

Apprendimento con Algoritmi Genetici (III)

Componenti di un sistema classificatore



Apprendimento con Algoritmi Genetici (IV)

Sistemi classificatori

- Un messaggio in un SC è una stringa di lunghezza finita su un alfabeto finito $\{0, 1\}$ (ad esempio, **110010**, **100101**).
- Un classificatore è un tipo speciale di regola di produzione (**se <condizione> allora <messaggio>**).
- La condizione del classificatore è usata per fare un riconoscimento di patterns semplice, e ammette il simbolo # (jolly) (**<condizione> $\in \{0, 1, \#\}$**).

Apprendimento con Algoritmi Genetici (V)

Sistemi classificatori

- Quando un classificatore è riuscito a soddisfare la sua condizione, esso diventa un candidato a porre il suo messaggio sulla lista di messaggi.
- Tutti i classificatori che sono riusciti a soddisfare la loro condizione entrano in una gara nella quale i classificatori più adeguati vengono scelti per collocare i loro messaggi sulla lista di messaggi.
- Ogni classificatore viene caratterizzato da una forza S , la quale è una misura del suo valore, e quando riesce a soddisfare la sua condizione fa una offerta proporzionale alla sua forza.

Apprendimento con Algoritmi Genetici (VI)

Sistemi classificatori

- Dopo che un classificatore è stato scelto per attivarsi, esso deve pagare la sua offerta a tutti gli altri classificatori che hanno contribuito alla sua attivazione.
- Il ruolo dell'algoritmo genetico consiste in creare nuovi classificatori che possano contribuire a migliorare l'esecuzione del sistema, mantenendo un elevato livello di esecuzione on-line.
- In questo tipo di applicazione non è desiderabile la completa selezione e rimpiazzo di una nuova popolazione in ogni generazione.

Principali Aree di Applicazione degli Algoritmi Genetici

- **Informatica**
- **Telecomunicazioni**
- **Ricerca operativa (ottimizzazione di funzioni)**
- **Biologia e medicina**
- **Processamento di immagini e riconoscimento di patterns**
- **Applicazioni ibride (reti neurali, sistemi esperti, robotica, sistemi paralleli)**

Tools e simulatori di Algoritmi Genetici

Tool	Description	http
Genetic algorithm viewer	Demonstration applet of the functioning of a Genetic Algorithm	http://www.rennard.org/alife/english/gavgb.html
Genetic Algorithms Toolkit	It is a general purpose genetic algorithm toolkit where the user can define and run his own optimization problems	http://www.aridolan.com/ga/gaa/gaa.html#Download

Riferimenti

- **M. Mitchell, An Introduction to Genetic Algorithms, The MIT Press, 1998.**
- **D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, 1° ed., Addison-Wesley Professional, 1989.**