Computazione evoluzionistica: algoritmi genetici e strategie evolutive - Fondamenti ed esempi di applicazione nel campo della Computer-Vision, Biomeccanica ed Elaborazione di Immagini Biomediche"

Cerveri Pietro
Dipartimento di Bioingegneria – Politecnico di Milano
25,27 Ottobre 2005

Evoluzione

- L'evoluzione è un processo continuo distribuito su una scala temporale ampia che cambia una popolazione di organismi generando prole via via migliore attraverso la riproduzione

Computazione evoluzionistica

- Termine generico che indica una gamma di sistemi di risoluzione dei problemi basati sull'utilizzo del calcolatore e affini ai processi evolutivi.
- Algoritmi genetici, Programmazione Evolutiva, le Strategie Evolutive, i Sistemi Classificatori e la Programmazione Genetica.

3/50

Un problema è risolto da un processo evoluzionistico che emula l'evoluzione naturale nel ricercare la migliore soluzione

- 1960: Ingo Rechenberg introduce l'idea della computazione evoluzionistica nel suo lavoro "Evolution strategies"
- L. Fogel 1962 (San Diego, CA): Programmazione evoluzionistica
- 1975: John Holland inventa gli algoritmi genetici e pubblica il libro "Adaption in Natural and Artificial Systems"
- 1992: John Koza propone gli algoritmi genetici per fare evolvere programmi software che effetuano specifici compiti. Tale metodo viene chiamato da lui Programmazione Genetici

4/50

Sono stati proposti per diverse applicazioni:

- Ottimizzazione (e.g., addestramento di reti neurali, minimizzazioni di funzioni costo, layout di circuiti, job shop scheduling, . . .)
- Predizione (e.g., previsioni del tempo, disposizione spaziale di proteine, . . .)
- Classificazione (e.g., verifica di truffe, verifica di qualità, . . .)
- Economia (e.g., strategie d'offerta, valutazione del mercato, . . .)
- Ecologia (e.g., competizione biologica, coevoluzione ospite-parassita, . . .)
- Programmazione automatica (e.g., generatori automatici di codice,..)

In generale sono da considerarsi indicati per:

- Ricerca in spazi di dimesionalità elevata, multimodali, e non smooth
- Funzioni con rapporto S/N elevato di cui non conosco la formulazione analitica o tale formulazione è estremente complessa
- Convergenza sub-ottima ma in un tempo ragionevole

5/50

Terminologia

- Basi (azotate): elementi fondamentali: Adenina, Citosina, Guanina, Timina (ed Uracile).
- DNA: è una struttura ad elica costituita da due filamenti di basi azotate. Le basi sui due filamenti sono accoppiate secondo: A-T e C-G;
- Cromosoma: informazione codificante del DNA che caratterizza un organismo
- Gene: Blocco elementare di DNA che codifica informazione (e.g.,: colore occhi)
- Allele: Uno dei possibili valori di espressione di un gene (e.g., marrone, blu, . . .)
- Tratto: La caratteristica fisica codificata da un gene
- Genoma: Un insieme completo di materiale genetico in un organismo
- Genotipo: Un particolare insieme di geni all'interno del genoma
- Fenotipo: La realizzazione fisica di un genotipo (e.g., una persona)
- Fitness: Una misura di succeso in vita di un organismo
- Ricombinazione: Cromosomi parentali scambiano tra loro materiale genetico per generare una nuova prole
- Mutazione: Errore che avviene durante la replicazione del DNA dai genitori

6/50

Matematicamente….

- Cromosoma: La codifica di una possibile soluzione per un dato problema di solito rappresentata tramite un array di bit o caratteri
- Gene: Un singolo bit o insieme di bit che codifica una parte della soluzione
- Allele: Uno degli elementi utilizzati per codificare i geni
- Fitness: Valutazione della soluzione attuale

Per rappresentare un processo di apprendimento come evoluzione

- Ricombinazione: Generare nuove soluzioni "mescolando" 2 o più soluzioni esistenti
- Mutazione: Cambiamenti casuali nella soluzione

7/50

Calcolo evolutivo

- Codifica la potenziale soluzione di uno specifico problema in una struttura dati simile al cromosoma
- Applica il processo di ricombinazione a queste strutture per preservare l'informazione critica

8/50

Ricerca per la risoluzione di problemi
Opt F(x1,x2,x3,…)

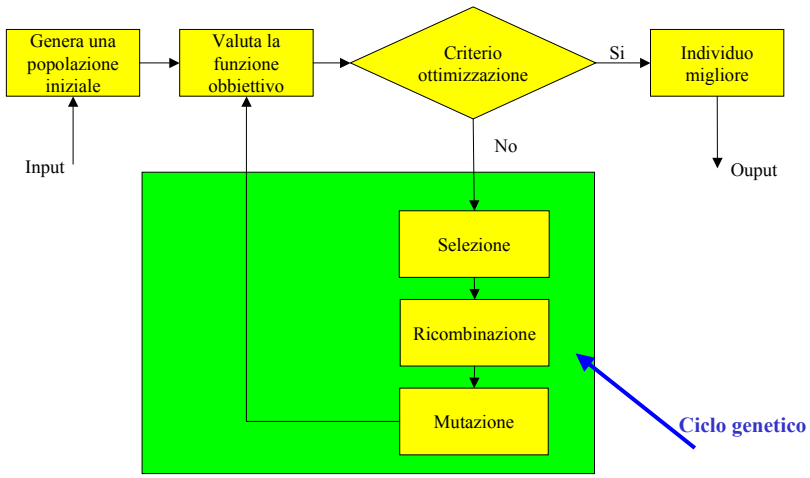
- Lo spazio delle soluzioni possibili contiene una o più soluzioni accettabili, eventualmente alcune ottime e altre sub-ottime.
- Gli algoritmi di ricerca servono a localizzare tali soluzioni evitando possibilmente di esplorare TUTTE le soluzioni possibili
- Epistasi: interazione tra le variabili del problema

9/50

ricerca di uno stato
    es: quadrato magico (sudoku)
ricerca di un cammino
    es: percorso tra due stati (teorema)
ricerca della soluzione ottima
    es: minimizzazione di funzioni
ricerca di una soluzione qualunque
    es: cammino tra due luoghi
ricerca di una soluzione subottima
    es: cammino quando la ricerca è un costo
ricerca in ambiente avverso
    es: giochi – esiste un avversario

10/50

Algoritmo genetico



11/50

Algoritmo genetico

1. [Inizia] Genera una popolazione random di n cromosomi (soluzioni plausibili - genitori)
2. [Fitness] Valuta la fitness f(x) di ciascun cromosoma x nella popolazione corrente
3. [Nuova popolazione] Crea una nuova popolazione (prole) attraverso:
   (a) [Selezione] Seleziona i cromosomi genitori in base alla loro fitness (criterio)
   (b) [Ricombinazione] Con una certa probabilità di ricombinazione incrocia due genitori per generare un nuovo figlio. Se la ricombinazione ha probabilità nulla, il figlio è una copia esatta dei genitori
   (c) [Mutazione] Con una certa probabilità di mutazione cambia gli elementi che costituiscono il figlio
   (d) [Accettazione] Introduce il nuovo figlio nella popolazione
4. [Verifica] Se la condizione di uscita è soddisfatta, la soluzione del problema è rappresentata dall'elemento nella popolazione che presenta la migliore fitness
5. [Ciclo] Vai al passo 2

12/50

Problemi intrinseci

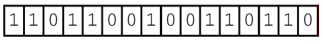
- Come creare i cromosomi e che tipo di codifica utilizzare
- Come selezionare i genitori per la ricombinazione nella speranza che i genitori migliori produrranno migliore prole
- Come definire la ricombinazione e la mutazione

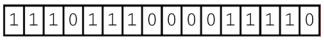
Codifica del cromosoma

Il primo passo per sviluppare un algoritmo genetico consiste nel definire la codifica della soluzione:

- un cromosoma deve contenere l'informazione sulla soluzione che rappresenta
- La codifica dipende principalmente dal problema da risolvere (e.g. numeri interi, numeri reali, permutazione, alberi di parsing,…)
- La modalità usuale di codifica consiste nel utilizzare una stringa binaria: ciascun bit nella stringa rappresenta una qualche caratteristica della soluzione

Il nostro cromosoma protrebbe essere quindi:

Cromosoma 1: 

Cromosoma 2: 

Es: rappresentazione binaria di un numero reale

Codifica implicita

- La codifica binaria è la più comune codifica implicita (principalmente perché la prima ricerca di GA aveva utilizzato tale codifica)
  - Codifica binaria: ogni cromosoma è una stringa di bits (1 o 0)
  - L'implementazione di operatori genetici risulta essere immediata
  - Tuttavia non è sempre naturale per molti problemi

Esempio: problema dello zaino

Date un certo numero di oggetti caratterizzati ciascuno da un valore e una dimensione, massimizzare il valore degli oggetti in uno zaino che ha una capacità fissata

In questo caso, ogni bit può rappresentare il fatto che un'oggetto è inserito o no nello zaino

Codifica binaria (8 bits)

Numero intero compreso tra 0 e 255

Genotipo Fenotipo

1	0	1	0	0	0	1	1

= 163

$1*2^7 + 0*2^6+1*2^5+0*2^4+0*2^3+0*2^2+1*2^1+1*2^0=$

128 + 32 + 2 + 1 = 163

Numero reale compreso tra 0 e 10

Genotipo Fenotipo

1	0	1	0	0	0	1	1

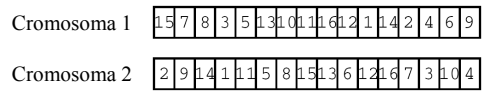
= 6.3922

0 + 163/255*(10-0) = 6.3922

Codifica di permutazioni

La codifica di permutazione è utilizzata per problemi di ordinamento

- Il cromosoma è una stringa di numeri che rappresenta la posizione in una sequenza
- La ricombinazione e la mutazione devono essere determinate in modo da produrre prole consistente



Esempio: problema del commesso viaggiatore (Traveling salesman problem)

- È dato un insieme di città e le corrispondenti distanze a coppie. Il commesso viaggiatore deve visitarle tutte ma non vuole viaggiare più del necessario. Scopo: Trovare una sequenza di città che minimizza la distanza percorsa.
  - Il cromosoma descrive l'ordine delle città visitate

Codifica esplicita

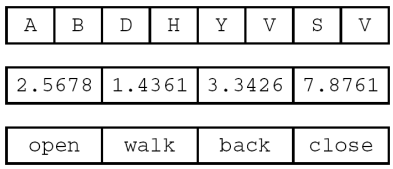
La codifica esplicita o diretta è indicata in problemi dove le entità coinvolte sono caratterizzate dove complessità

- Il cromosoma è una sequenza di valori connessi al problema (numeri, caratteri, stringhe di caratteri, …)
- Rappresenta la scelta migliore per ottimizzazione in spazi multidimensionali e/o multimodali. Gli operatori di ricombinazione e mutazione richiedono specifici controlli di consistenza

Esempio: addestramento di reti neurali (determinazione dei pesi)

Una rete neurale è specificata attraverso una architettura. Trovare i pesi tra i neuroni che producono il comportamento desiderato della rete

I numeri reali nel cromosoma rappresentano i pesi della rete neurale.



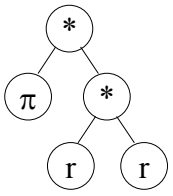
Codifica per albero

La codifica ad albero è utilizzata principalmente per evolvere programmi o espressioni matematiche (Programmazione genetica)

- Il cromosoma è un albero di oggetti, come funzioni o comandi in un linguaggio di programmazione
- Il linguaggio LISP è spesso utilizzato per questo scopo. La definizione di ricombinazione e mutazione risulta più semplice

Esempio: Trovare una funzione che approssimi la relazione tra coppie di valori

- Gli ingressi e le uscite sono dati. Lo scopo consiste nel determinare una funzione che produrrà le migliori uscite per tutti gli ingressi
- Il cromosoma è una funzione rappresentata in un albero come sequenza di operatori e valori



Cromosoma: (πr^2)

Selezione

- In accordo alla teoria evoluzionistica di Darwin il migliore cromosoma sopravvive per creare nuova prole. Per selezionare il migliore cromosoma esistono diversi criteri:
  - Ordinamento (priorità diretta ai cromosomi che si sono espressi meglio)
  - Casualità (metodo della roulette, solo tra i cromosomi migliori)
  - Probabilità crescente con l'incremento di espressione
  - …

Elitismo

Quando si crea una nuova popolazione tramite ricombinazione e mutazione corriamo il rischio di perdere il migliore cromosoma ottenuto fino a quel momento. Elitismo è la procedura che salvaguardia il passato copiando per primo nella nuova popolazione il cromosoma migliore

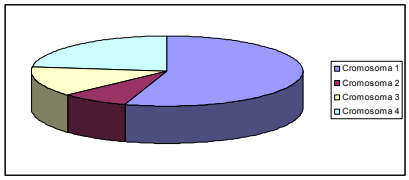
Definizioni

- **Pressione della selezione:**
  - Probabilità del migliore individuo a essere selezionato in rapporto alla probabilità media di selezione di tutti gli individui
- **Polarizzazione della selezione :**
  - Differenza in modulo tra la fitness normalizzata di un individuo e la probabilità attesa di riproduzione
- **Diffusione:**
  - Range dei possibili valori per il numero di figli prodotti da un individuo
- **Perdita di diversità:**
  - Proporzione degli individui di una popolazione che non è selezionata durante la fase di selezione
- **Intensità della selezione**
  - Valore medio atteso della fitness di una popolazione dopo aver applicato un metodo di selezione ad una distribuzione Gaussiana
- **Varianza della selezione**
  - Valore di varianza atteso per la distribuzione di fitness di una popolazione dopo aver applicato un metodo di selezione ad una distribuzione Gaussiana

Selezione tramite roulette

Gli individui sono selezionati proporzionalmente alla loro fitness. Migliore essa è e più alta è la probabilità di selezione

1. Si immagini una roulette dove sono sistemati tutti i cromosomi della popolazione
2. La dimensione della sezione nella roulette è proporzionale al valore di fitness di ciascun cromosoma
3. La pallina viene lanciata all'interno della roulette e il cromosoma in corrispondenza del quale si ferma è quello selezionato

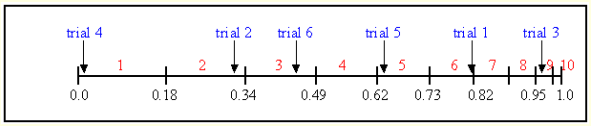


Roulette

Numero di individui	1	2	3	4	5	6	7	8	9	10	11
Fitness	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2	0.0
Probabilità di selezione	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.06	0.03	0.02	0.00

- **Generazione casuale di 6 numeri**
  - 0.81, 0.32, 0.96, 0.01, 0.65, 0.42

Individuo



- **Dopo la selezione la nuova popolazione è costituita dai seguenti individui**
  - 1, 2, 3, 5, 6, 9.
- **Non è polarizzata**

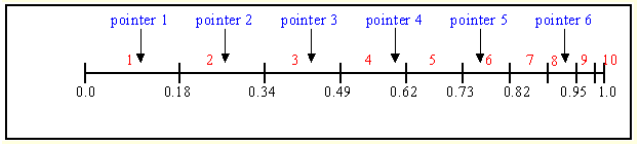
23/50

Stochastic universal sampling

Numero di individui	1	2	3	4	5	6	7	8	9	10	11
Fitness	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2	0.0
Probabilità di selezione	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.06	0.03	0.02	0.00

- **Generazione 6 puntatori equispaziati**

Individuo



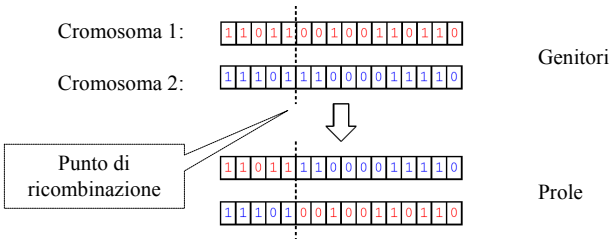
- **Dopo la selezione la nuova popolazione è costituita dai seguenti individui**
  - 1, 2, 3, 6, 8.
- **Non è polarizzata**

24/50

Operatore di Ricombinazione

Il processo di ricombinazione opera su geni selezionati dal cromosoma genitore e crea nuova prole:

1. Selezione random di un **punto di ricombinazione** all'interno del cromosoma
2. Copia tutti i geni precedenti questo punto dal primo genitore e poi copia tutti i geni successivi a questo punto dal cromosoma del secondo genitore



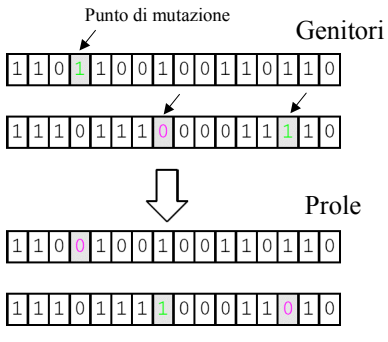
Nota: la ricombinazione dipende principalmente dalla codifica dei cromosomi. Una ricombinazione specifica per un dato problema può migliorare o ridurre le prestazioni di un algoritmo genetico

25/50

Operatore di Mutazione

Dopo la ricombinazione interviene il processo di mutazione

Cambia in modo casuale un numero contenuto di bits da 0 a 1 o viceversa da 1 a 0



26/50

Mutazione

Note

- La mutazione è concepita per evitare la possibile caduta delle soluzioni della popolazione in ottimi locali
- Anche la mutazione dipende dal tipo di codifica dei cromosomi (e.g. quando codifichiamo permutazioni, la mutazione può essere effettuata come scambio tra due geni

27/50

Ricombinazione/mutazione (I)

Per la codifica binaria esistono diverse modalità:

- Ricombinazione da un singolo punto: selezionato un punto all'interno del cromosoma, la stringa binaria compresa tra l'inizio e il punto di ricombinazione è copiata dal primo genitore, il resto è copiato dal secondo genitore
- Ricombinazione da due punti: selezionati i due punti all'interno del cromosoma, la stringa binaria compresa tra l'inizio e il primo punto di ricombinazione è copiata dal primo genitore, la parte compresa tra il primo e il secondo punto è copiata dal secondo genitore, la parte compresa tra i secondo punto e la fine del cromosoma è copiata ancora dal primo genitore
- Ricombinazione uniforme: i bit sono copiati casualmente dal primo o dal secondo genitore
- Ricombinazione aritmetica: operatori algebrici (e.g., logica AND, OR, ..)
- Mutazione: inversione dei bit selezionati

28/50

Ricombinazione/mutazione (II)

Per la codifica esplicita tramite numeri reali esistono diverse modalità

- Ricombinazione:
  - **discreta**

    Individuo 1: [12 25 5] estrazione 1: 2 2 1 Figlio 1: [123 4 5]
    Individuo 2: [123 4 34] estrazione 1: 1 2 1 Figlio 2: [12 4 5]
  - **intermedia**

    figlio = padre1 + α(padre2-padre1)
  - **lineare**
- Mutazione: perturbare i valori aggiungendo un rumore casuale Spesso è utilizzata una distribuzione Gaussiana $N(\sigma,0)$ - $X'_i=X_i+N(\sigma,0)$
  - Media nulla
  - Deviazione standard σ
  - (1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0) $\Rightarrow$ (0.9 2.2 3.1 4.2 5.5 6.0 6.8 8.1 9.1)

29/50

Ricombinazione/mutazione (III)

Per la codifica di permutazione occorre garantire consistenza:

- Ricombinazione da un singolo punto: selezionato un punto all'interno del cromosoma, la permutazione compresa tra l'inizio e il punto di ricombinazione è copiata dal primo genitore, poi il secondo genitore is analizzato selezionando i numeri mancanti
  - (1 2 3 4 5 **6** 7 8 9) + (4 5 3 6 8 9 7 2 1) $\Rightarrow$ (1 2 3 4 5 6 8 9 7)
- Mutazione per cambiamento d'ordine:
  - (1 2 3 4 5 6 8 9 7) $\Rightarrow$ (1 **8** 3 4 5 6 **2** 9 7)

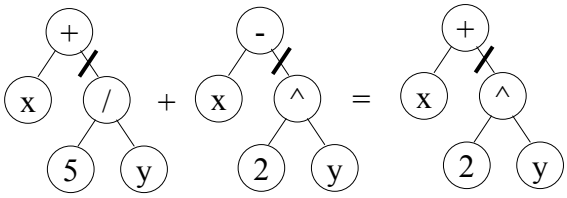
30/50

Ricombinazione/mutazione (IV)

Per la codifica per albero occorre selezionare specifici operatori:

Ricombinazione: in ciascun genitore è selezionato un punto di ricombinazione. I genitori sono divisi nel punto di ricombinazione e le parti al disotto di quel punto vengono scambiate per produrre una nuovo figlio

Mutazione: operatori, numeri, variabili sono cambiate in modo casuale



31/50

Problemi e regole euristiche

- Funzione costo
- Popolazioni di dimensioni ridotte di solito garantiscono convergenza veloce. Dimensioni ampie evitano ottimi locali ma il costo computazionale può essere critico. Di solito 30-50 individui rappresentano un trade-off adeguato
- Il tasso di ricombinazione dovrebbe essere alto intorno al 90%
- Il tasso di mutazione dovrebbe essere basso; i migliori tassi sembrano essere circa 1% per allele
- La selezione casuale e ordinamento sono di solito indicate come le più efficaci. Il metodo di **elitismo** dovrebbe essere implementato se non si prevedono altri meccanismi che tengono memoria delle soluzioni migliori
- Convergenza prematura

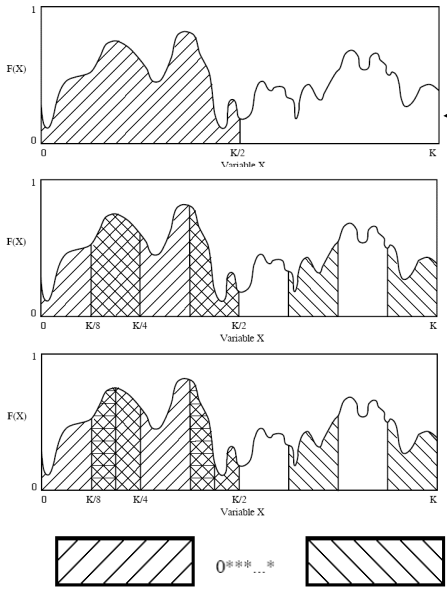
32/50

“Schema” - Holland (1975)

- Una stringa S contiene sotto-stringhe. Questo può essere espresso tramite un template di similarità (schema) H
- Esempio: lo schema **11 rappresenta le stringhe 1111, 0011, 1011 and 0111.
- Al contrario, 1011 and 1101 contengono schemi comuni 1***, ***1 e 1**1.
- Lo schema rappresenta una partizione dello spazio di ricerca

Parallelismo implicito

- Dato che 2 stringhe hanno 3 schemi in comune ognuna producendo un totale di 6 schemi
- Analizzare una stringa di bit quindi equivale a processare un numero elevato di schemi
- In generale, una stringa di lunghezza L presenta 3^L schemi



Esempio: max(f)

La partizione 0****..* che codifica soluzione da 0 a K/2 produce risultati **mediamente** migliori della partizione 1****..* che codifica soluzione da K/2 a K

L'iperpiano 0-K/2 avra' una probabilità superiore di essere campionato

Il campionamento delle partizioni non è affetto dai valori ottimi locali

Problema: non garantisce di raggiungere comunque il valore ottimo globale se questo è per esempio isolato. Le soluzioni via via trovate sono comunque globalmente competitive

Teorema degli schemi - Holland (1975)

- Valore atteso di valutazione del numero di schemi tra una generazione e la successiva

$$M\left(H_i, t+1\right) \geq M\left(H_i, t\right)\left[\frac{f\left(H_i\right)}{\overline{f}}\right]\left[1-p_c\frac{\delta\left(H_i\right)}{L-1}\right]\left[\left(1-p_m\right)^{o\left(H_i\right)}\right]$$

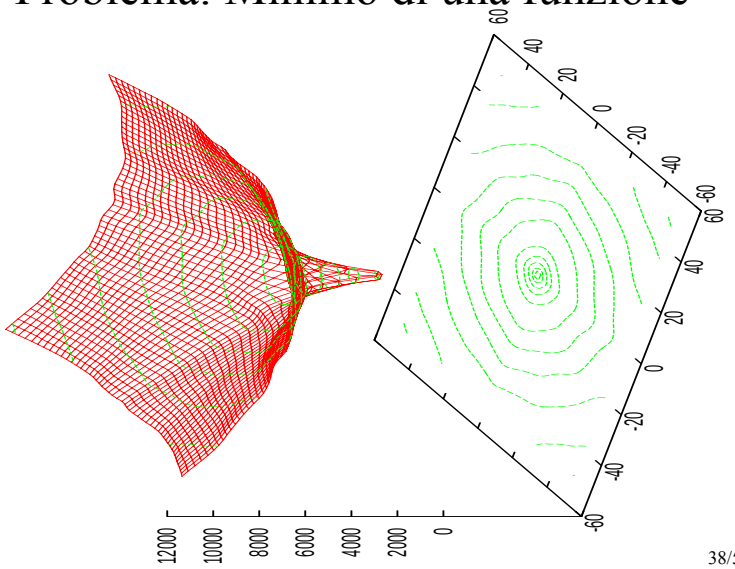
- Fitness media del *i*^{esimo} schema $f\left(H_i\right)$
- Fitness media della popolazione $\overline{f}$
- Lunghezza del *i*^{esimo} schema $\delta\left(H_i\right)$
- Lunghezza delle stringhe nello spazio di ricerca L
- Numero di bits validi nello schema $o\left(H_i\right)$
- Tasso di ricombinazione p_c
- Tasso di mutazione p_m

Sintesi

- Al disopra del valore medio di fitness schemi di lunghezza ridotta hanno una probabilità di sopravvivenza elevata e cresceranno almeno esponenzialmente nella popolazione
- Schemi di lunghezza ridotta sono chiamati “building blocks”; essi rappresentano soluzioni parziali del problema e la loro analisi (processing) conduce alla soluzione ottima

37/50

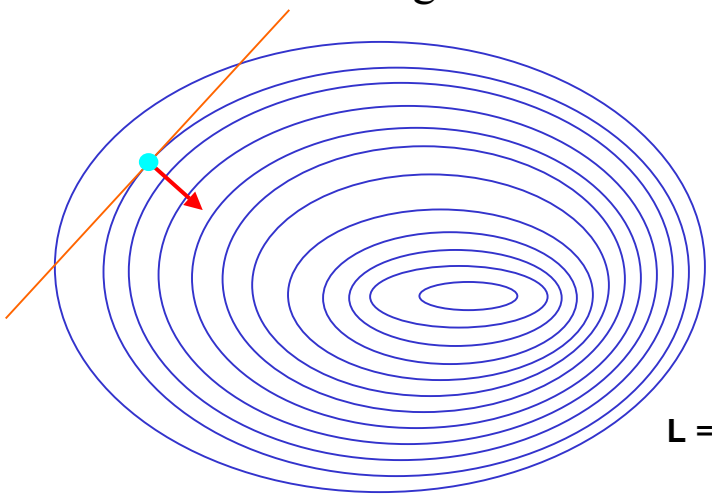
Problema: Minimo di una funzione



38/50

Metodo del gradiente

f(x , y)



S = Step

L = Learning Rate

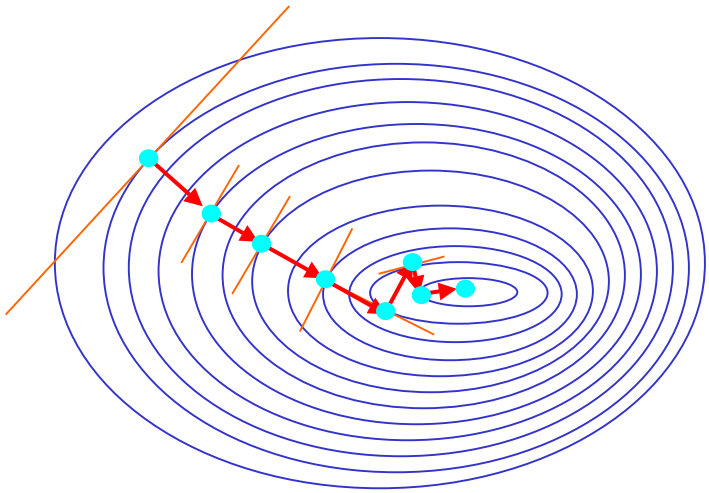
G(x , y) = ∇f(x , y)

S = L · |**G**(x , y)|

39/50

Metodo del gradiente

f(x , y)



G(x , y) = ∇f(x , y)

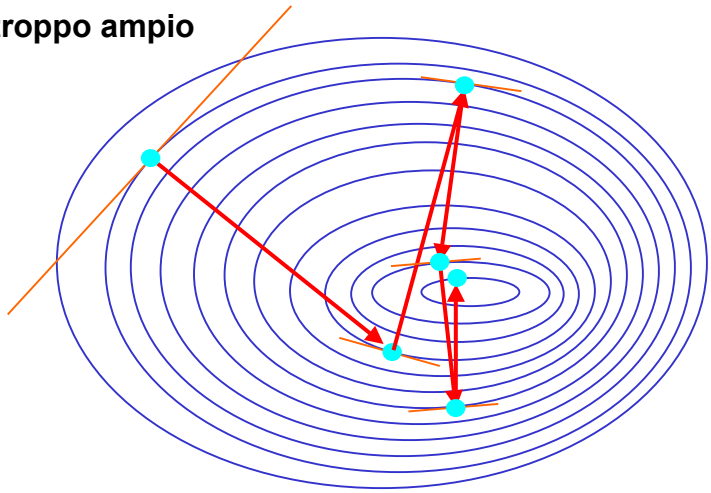
S = L · |**G**(x , y)|

40/50

Metodo del gradiente

L troppo ampio

f(x , y)



G(x , y) = ∇f(x , y)

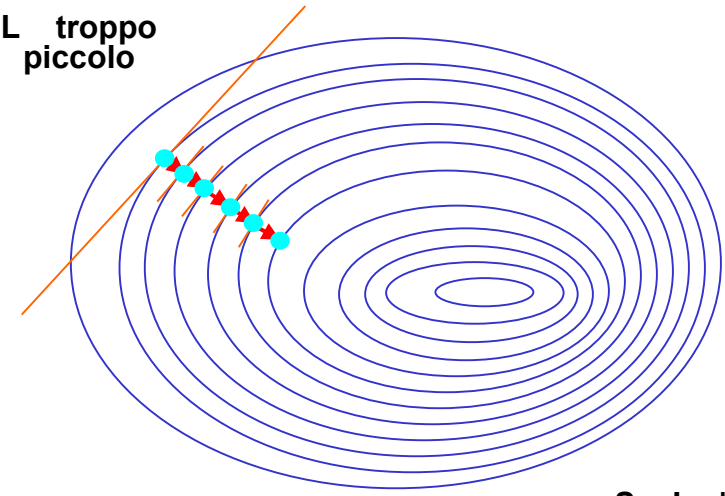
S = L · |G(x , y)|

41/50

Metodo del gradiente

L troppo piccolo

f(x , y)



G(x , y) = ∇f(x , y)

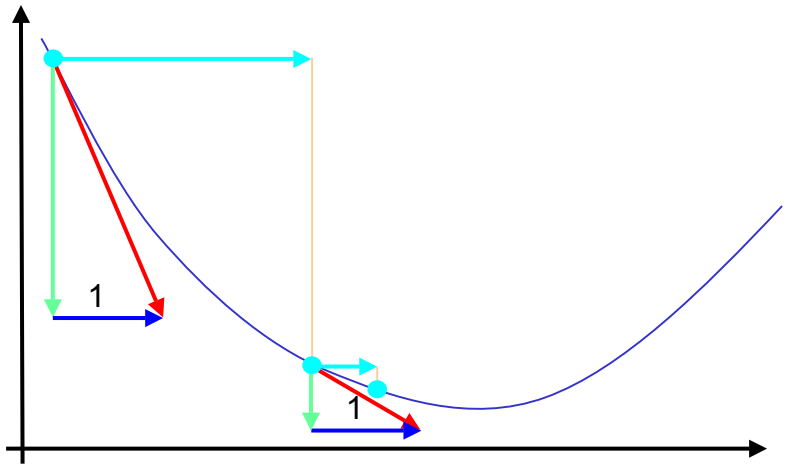
S = L · |G(x , y)|

42/50

Metodo del gradiente

S = L · |**G**(x)|

f(x)



43/50

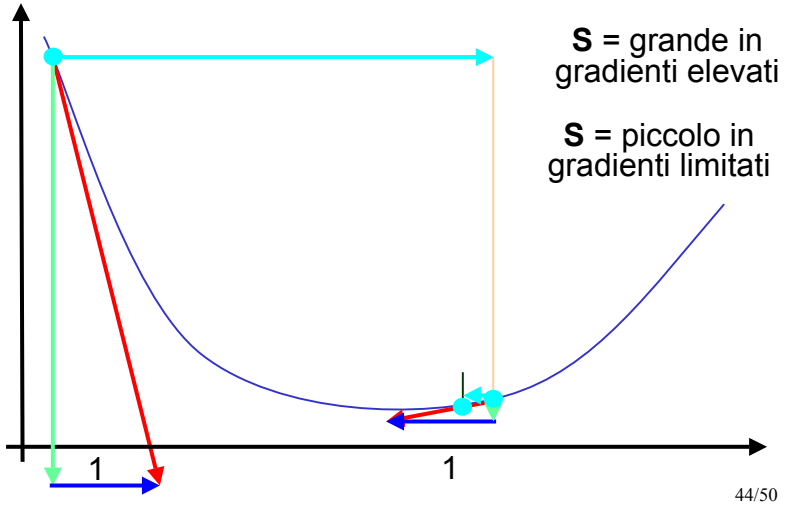
Metodo del gradiente

S = L · |**G**(x)|

f(x)

S = grande in gradienti elevati

S = piccolo in gradienti limitati

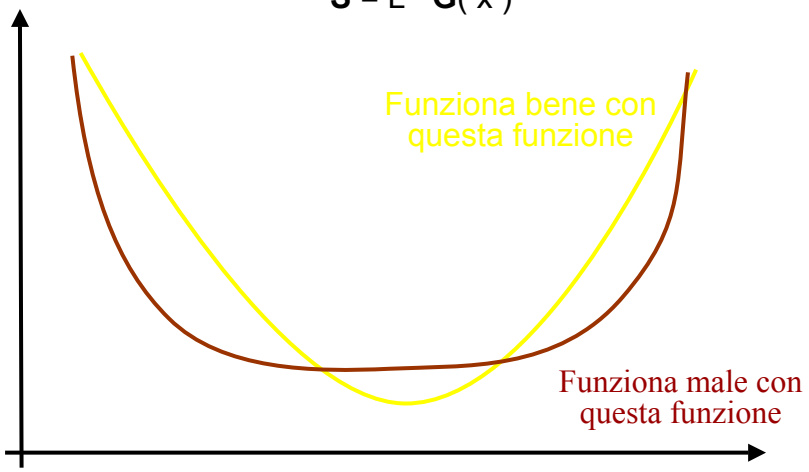


44/50

Metodo del gradiente

$\mathbf{S} = L \cdot \mathbf{G}(x)$

f(x)



45/50

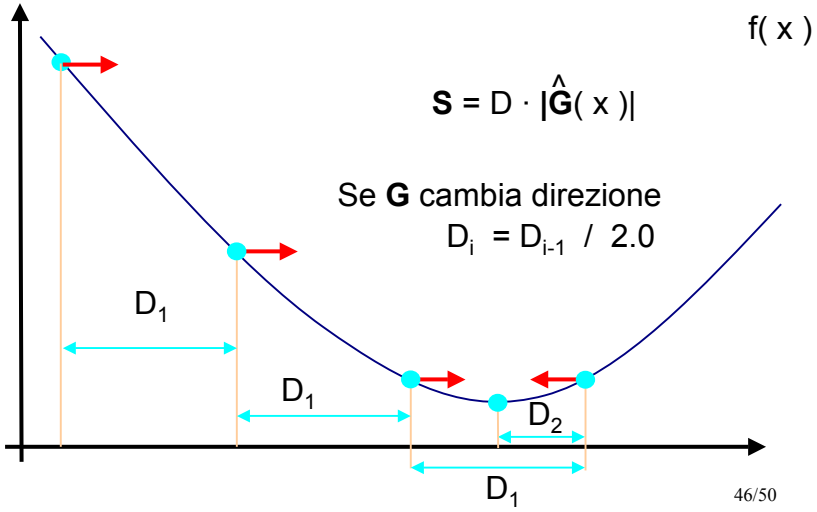
Metodo del gradiente a passo fisso

f(x)

$\mathbf{S} = D \cdot |\hat{\mathbf{G}}(x)|$

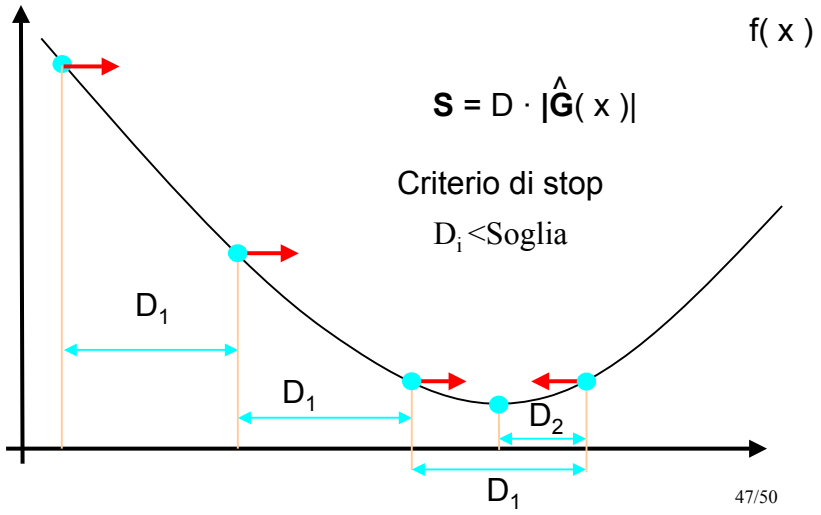
Se **G** cambia direzione

$D_i = D_{i-1} / 2.0$



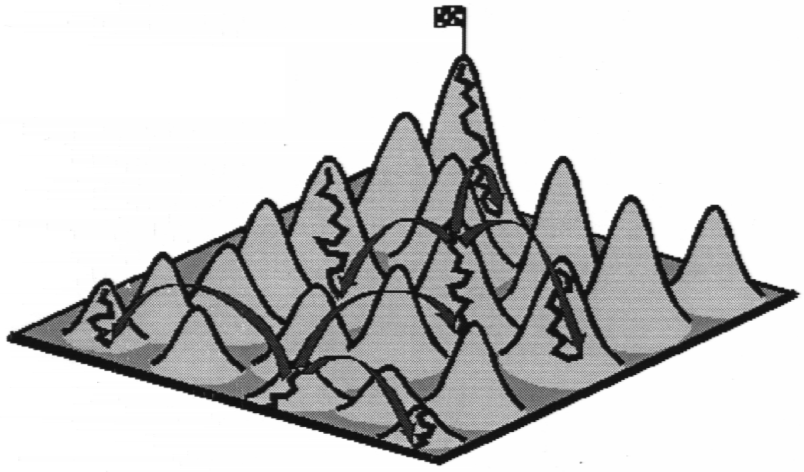
46/50

Metodo del gradiente a passo fisso



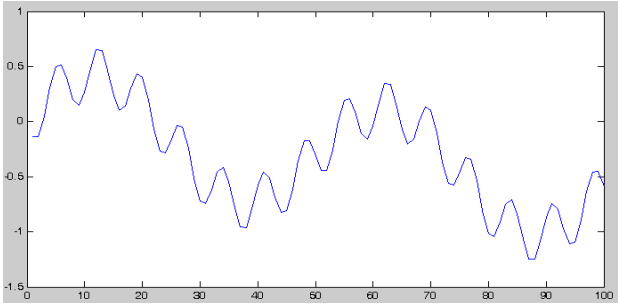
47/50

E questa funzione?



48/50

- In questo semplice esempio cerchiamo l'estremo di una funzione definita all'interno di un dominio di ricerca D in $\Re^1$
  - Spazio di ricerca: un intervallo dello spazio 1-dimensionale
  - Funzione costo: il valore della funzione che stiamo esplorando



Bibliografia

- Buche, D. Schraudolph, N.N. Koumoutsakos, P., Accelerating evolutionary algorithms with Gaussian process fitness function models, IEEE Transactions on Systems, Man and Cybernetics, Part C, 35(2), pp. 183- 194, 2005
- Cerveri P., Borghese N.A., Ferrigno G., Evolution strategies and epipolar geometry for dynamic calibration of a photogrammetric camera system, Medical & Biological Engineering & Computing, vol. 37(2), pp. 830-831, 1999.
- Cerveri P., Pedotti A., Borghese N.A., Combined evolution strategies for dynamic calibration of video based measurement systems IEEE Transaction on Evolutionary Computation, Volume: 5 Issue: 3, pp. 271 –282, June 2001.
- Cerveri P. Pedotti A., Ferrigno G., Evolutionary optimization for robust hierarchical computation of the rotation centres of kinematical chains from reduced ranges of motion: the lower spine case. Journal of Biomechanics, 37(12), pp. 1881-1890, 2004.
- Cerveri P., Lopomo N., Pedotti A., Ferrigno G., Derivation of centers and axes of rotation for wrist and fingers in a hand kinematic model: robust methods and reliability results, Annals of Biomedical Engineering, 33(3), pp. 402-12, 2005.
- Goldberg D.E. (1989), Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley Longman Inc.
- Hansen N. and Ostermeier A., "Completely derandomized self-adaptation in evolution strategies," *Evol. Comput.*, vol. 9, no. 2, pp. 159-195, 2001.
- Hansen N., Müller S. D., and Koumoutsakos P., "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evol. Comput.*, vol. 11, no. 1, pp. 1-18, 2003.
- Holland J.H. (1975), Adaptation in Natural and Artificial Systems, Cambridge MA, MIT Press.
- Schwefel, H.-P. (1981). Numerical optimization of computer models. Chichester: Wiley.