

Metodi e tecniche di ottimizzazione innovative per applicazioni elettromagnetiche

Algoritmi stocastici

Parte 1 Tecniche evoluzionistiche

M. Repetto

Politecnico di Torino, Dip. Ingegneria Elettrica Industriale



Contents

- tecniche evoluzionistiche
 - evolution strategy
 - genetic algorithm
- metodi collettivi
 - ant colony
 - particle swarm
- artificial immune systems



Algoritmi stocastici (1/2)

- Diversi algoritmi di ottimizzazione stocastica sono stati sviluppati implementando analogie con fenomeni naturali
- lo studio dell'*artificial life* ha fornito diversi spunti per il *problem solving*, tutti basati su popolazioni di individui interagenti la cui dinamica è regolata da leggi semplici:
 - evoluzione di una specie biologica che modifica nel tempo le sue caratteristiche per affrontare al meglio le interazioni con l'ambiente in cui vive
 - comportamento cooperativo di un gruppo che consente di affrontare l'interazione con l'ambiente meglio di quanto non possa fare il singolo
 - sistema di difesa di un organismo che deve difendersi da attacchi biologici in forma sempre diversa



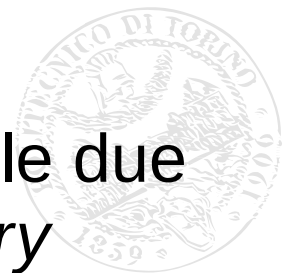
Algoritmi stocastici (2/2)

- Vengono presentati tre diverse classi di algoritmi
 - tecniche evoluzionistiche: basate sulla competizione tra gli individui della popolazione *Evolution Strategy ES, Genetic Algorithm GA*
 - tecniche collettive: basate sulla collaborazione e sullo scambio di informazioni tra individui *Particle Swarm Optimization PSO, Ant Colony ACO*
 - tecniche immuni: basate sulla massima diversificazione degli individui *Artificial Immune Systems AIS*



Tecniche Evoluzionistiche

- All'interno delle tecniche evoluzionistiche si evidenziano due metodi diversi per origine e struttura
 - Evolutionary Strategy: sviluppate in ambiente di lingua tedesca
1965 Rechenberg: Cybernetic solution path of an experimental problem
 - Genetic Algorithm: sviluppati in ambito americano
1966 Fogel Owens Walsh Artificial intelligence through simulated evolution
1975 Holland, Adaptation in natural and artificial systems
- nonostante facciano riferimento alla evoluzione Darwiniana basata sulla sopravvivenza del piu' adatto la loro implementazione e' piuttosto diversa
- quasi 30 anni dopo il loro comparire sulla scena, le due tecniche vengono spesso denominate *evolutionary computation*



Evoluzionistici: tratti comuni (1/3)

- **Popolazione:** gruppo di individui che rappresenta una generazione della specie, un campionamento temporale durante la sua dinamica
 - un individuo e' un punto nello spazio dei parametri X
 - eventualmente all'individuo possono essere associate altre caratteristiche
- **Fitness:** misura della capacita' dell'individuo di sopravvivere nell'ambiente (per un coniglio la velocita', per un falco la vista etc.), la fitness e' la funzione obiettivo
- **Corredo genetico o genetic outfit:** caratteristiche dell'individuo che vengono trasmesse alla sua discendenza



Evoluzionistici: tratti comuni (2/3)

- **Operatori genetici:** meccanismi attraverso i quali viene prodotta la successiva generazione a partire dalla attuale, devono garantire un grado di casualita' alla nuova generazione
 - condivisione del corredo genetico: meccanismo alla base di tutta la riproduzione sessuata (operatore binario o superiore)
 - mutazione del corredo genetico: evento casuale dovuto ad errori (entropizzazione) nella copia del materiale genetico (operatore unario)



Evoluzionistici: tratti comuni (3/3)

- **Riproduzione** : meccanismo attraverso cui un individuo si riproduce identico nella generazione successiva
- **Selezione** : operazione decisionale attraverso cui si determinano gli individui della generazione successiva, puo' essere:
 - deterministica: solo i migliori sopravvivono
 - probabilistica: i migliori hanno piu' elevate probabilita' di sopravvivenza



Evolutionary Strategy (1/3)

- **Rappresentazione degli individui:** gli individui vengono rappresentati come vettori floating point all'interno del dominio di problema

ogni individuo ha associato altre informazioni come ampiezza di mutazione etc.

- **Popolazione:** μ genitori si combinano tra di loro dando luogo a λ discendenti

- **Operatori genetici:** i genitori si riproducono attraverso

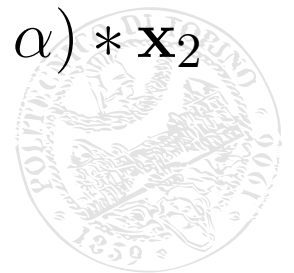
- X-over:

$$\mathbf{x}_A = (1 - \alpha) * \mathbf{x}_1 + \alpha * \mathbf{x}_2 \quad \mathbf{x}_B = \alpha * \mathbf{x}_1 + (1 - \alpha) * \mathbf{x}_2$$

- mutazione

$$\mathbf{x}_{new} = \mathbf{x}_{old} + r * \delta_{old}$$

- gli operatori vengono applicati in sequenza



Evolutionary Strategy (2/3)

- **Ampiezza di mutazione:** ogni individuo porta associata un'ampiezza di mutazione δ_{old}
 - l'ampiezza di mutazione viene ereditata dalle configurazioni discendenti
 - viene modificata dal tasso di successo delle configurazioni
 - tasso di successo alto $\Rightarrow$ aumento di δ_{old}
 - tasso di successo basso $\Rightarrow$ diminuzione di δ_{old}
- alcune implementazioni associano all'individuo anche una direzione di ricerca che si e' rivelata fruttuosa in passato



Evolutionary Strategy (3/3)

- **Selezione:** la selezione e' effettuata in modo strettamente deterministico facendo sopravvivere tra i nuovi individui solo i μ con fitness piu' elevata
esistono diverse tecniche per determinare la generazione successiva
 - tecnica (λ, μ) : la selezione viene effettuata solo tra i λ discendenti, ogni individuo vive solo una generazione
 - tecnica $(\lambda + \mu)$: la selezione viene effettuata considerando contemporaneamente genitori e figli, (fattore longevita')
- **Deterioramento della fitness:** con la tecnica (λ, μ) si accetta un peggioramento della fitness, mentre nella $(\lambda + \mu)$ il miglior valore ottenuto sopravvive sempre



Genetic Algorithm (1/6)

- **Rappresentazione individui:** gli individui vengono rappresentati mediante stringhe di numeri binari, eventualmente ottenute per conversione da valori reali

$$x_{1L} = 0, x_{1U} = 0.1, \text{Number of digit} = 8(2^8 = 128)$$

$$base_1 = (0.1 - 0) / 127 = 7.87 * 10^{-4},$$

$$x_1 = 0.05 \Rightarrow b_1 = \text{int} \frac{(0.05-0)}{7.87*10^{-4}} = 64 \Rightarrow 01000000$$

$$x_{2L} = 0.25, x_{2U} = 1, \text{Number of digit} = 10(2^{10} = 512)$$

$$base_2 = (1 - 0.25) / 512 = 1.46 * 10^{-3},$$

$$x_2 = 0.3 \Rightarrow b_2 = \text{int} \frac{(0.3-0.25)}{1.46*10^{-3}} = 34 \Rightarrow 0000001010$$

rappresentazione vettore $\mathbf{x} = \{x_1, x_2\}$

individuo stringa 18 bit $\rightarrow$ 0100000000000001010



Genetic Algorithm (2/6)

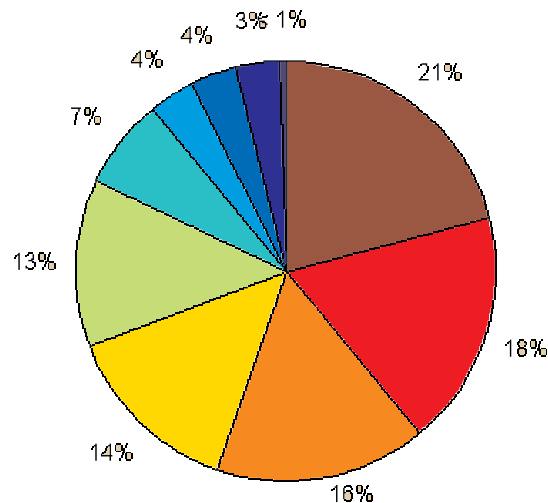
- **Popolazione:** la popolazione e' formata da un numero prefissato di individui
- **Mating pool:** popolazione intermedia di genitori che viene selezionata probabilisticamente in base al valore della fitness
- il procedimento di selezione della mating pool consente di prendere piu' volte lo stesso individuo



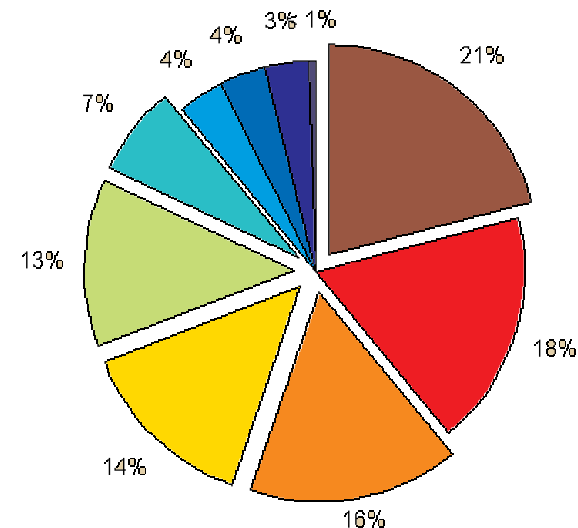
Genetic Algorithm (3/6)

- **Roulette wheel:** ad ogni configurazione e' assegnata una probabilita' di sopravvivenza data da:

$$p_i = \frac{f_i}{\sum_k f_k}$$



10 individui totali



5 individui selezionati

- **Deterioramento della fitness:** in maniera casuale la configurazione migliore potrebbe non essere scelta



Genetic Algorithm (4/6)

- **Operatori genetici:** una volta selezionata la popolazione di genitori vengono applicati gli operatori che producono i nuovi individui:
 - cross-over: due genitori mescolano il loro corredo genetico dando luogo a due individui figli
genitore 1 → 01010101 ↑ 100100100
genitore 2 → 01010000 ↑ 111100100
offspring 1 → 01010000 ↑ 111100100
offspring 2 → 01010000 ↑ 100100100
 - mutazione: un bit della configurazione viene cambiato in modo casuale:
genitore 1 → 01010101100100100
offspring 1 → 01010100100100100
- gli operatori vengono applicati in percentuale fissata a priori



Genetic Algorithm (5/6)

- **Selezione:** la selezione avviene nella scelta della mating pool quindi in maniera diversa da quanto fatto nella ES
- **Schema theorem:** la convergenza del metodo viene dimostrata attraverso il teorema che analizza l'evoluzione e la crescita nella popolazione delle configurazioni migliori
- **building block:** supponendo che le caratteristiche delle configurazioni migliori si concentrino in alcune regioni della stringa, la probabilità che queste passino alle generazioni successive è crescente durante il procedimento (selective pressure)

configurazione 1 → 01**11**100100100

configurazione 2 → 11**11**111000110



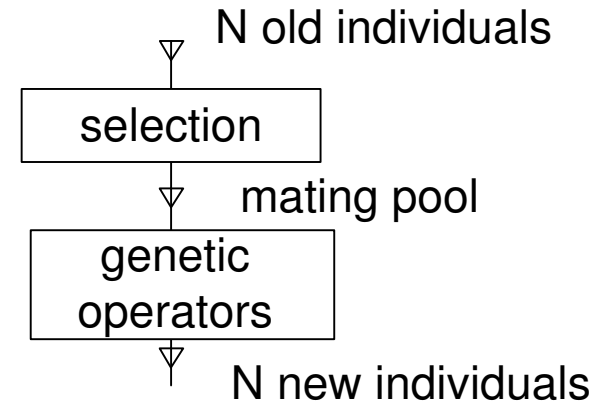
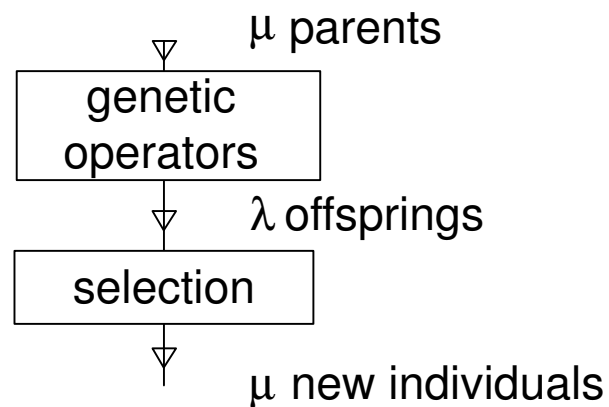
Genetic Algorithm (6/6)

- l'operatore x-over e' quello che meglio garantisce di non distruggere questi blocchi di informazione ed e' quindi il piu' utilizzato
- l'operatore mutazione serve a mantenere l'esplorazione dello spazio dei parametri generando nuove configurazioni
- coding: la codifica binaria da un'importanza gerarchica alla posizione dei bit cosi' che un flip in una posizione puo' portare ad una grande modifica della configurazione
- dato che in natura si vede che le mutazioni piu' utili sono quelle che creano piccole modifiche si preferisce una codifica in cui il flip di un bit porta ad una configurazione adiacente (Gray coding)



Confronto ES/GA (1/2)

ES		GA
floating point	representation	binary
deterministic	selection	probabilistic
changing	genetic operators	fixed



Selective pressure (1/3)

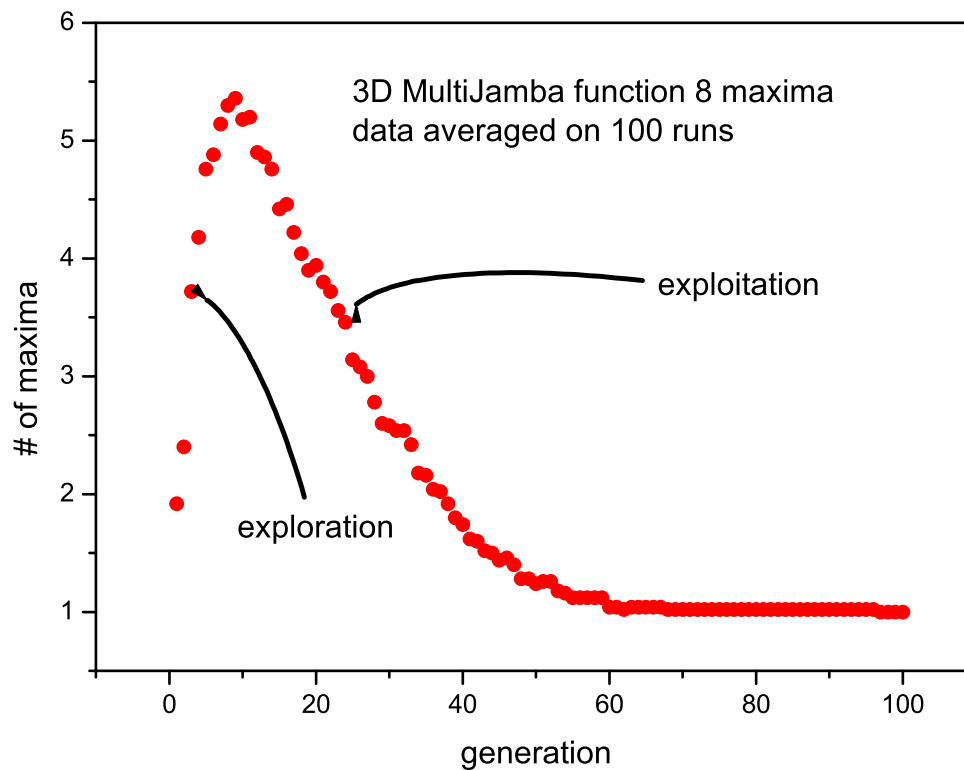
- le tecniche evoluzionistiche sono ottimizzatori globali e devono quindi ricercare la regione in cui questo si trova all'interno del dominio (**exploration**)
- la pressione selettiva imposta dall'algoritmo fa solitamente predominare pero' l'**exploitation** rispetto all'exploration
- la sopravvivenza dei migliori fa si che durante il processo si possa arrivare ad un *inbreeding* della popolazione, cioe' una riduzione del contenuto genetico della mating pool
- questo difetto dell'evoluzione in natura viene sorpassato da altri fenomeni interagenti con la specie biologica



Selective pressure (2/3)

- funzione multiJamba

$$f(\mathbf{x}) = -1 - \sum_{j=1}^{N_{dim}} \left(x_j^2 + \cos(2\pi \cdot x_j) \right)$$



Selective pressure (3/3)

- per contenere l'addensamento della popolazione si devono valutare gli individui non solo in base alla loro *fitness* bensì anche in base al loro corredo genetico (posizione nello spazio dei DoF)
- tra le modifiche proposte per ridurre la convergenza prematura dell'algoritmo si possono citare:
 - fitness sharing
 - clustering



Fitness sharing (1/2)

- l'applicazione delle tecniche evoluzionistiche comporta un addensamento della popolazione intorno a regioni caratterizzate da buoni valori di fitness
- in natura questo si verifica quando alcune circostanze naturali consentono ad una popolazione di espandersi grazie ad esempio ad un'abbondanza di risorse.
- In natura si ottiene un bilanciamento grazie all'equilibrio di due fenomeni:
 - abbondanza di risorse $\Rightarrow$ aumento densita' popolazione (azione positiva)
 - aumento densita' popolazione $\Rightarrow$ diminuzione risorse pro-capite (azione negativa)



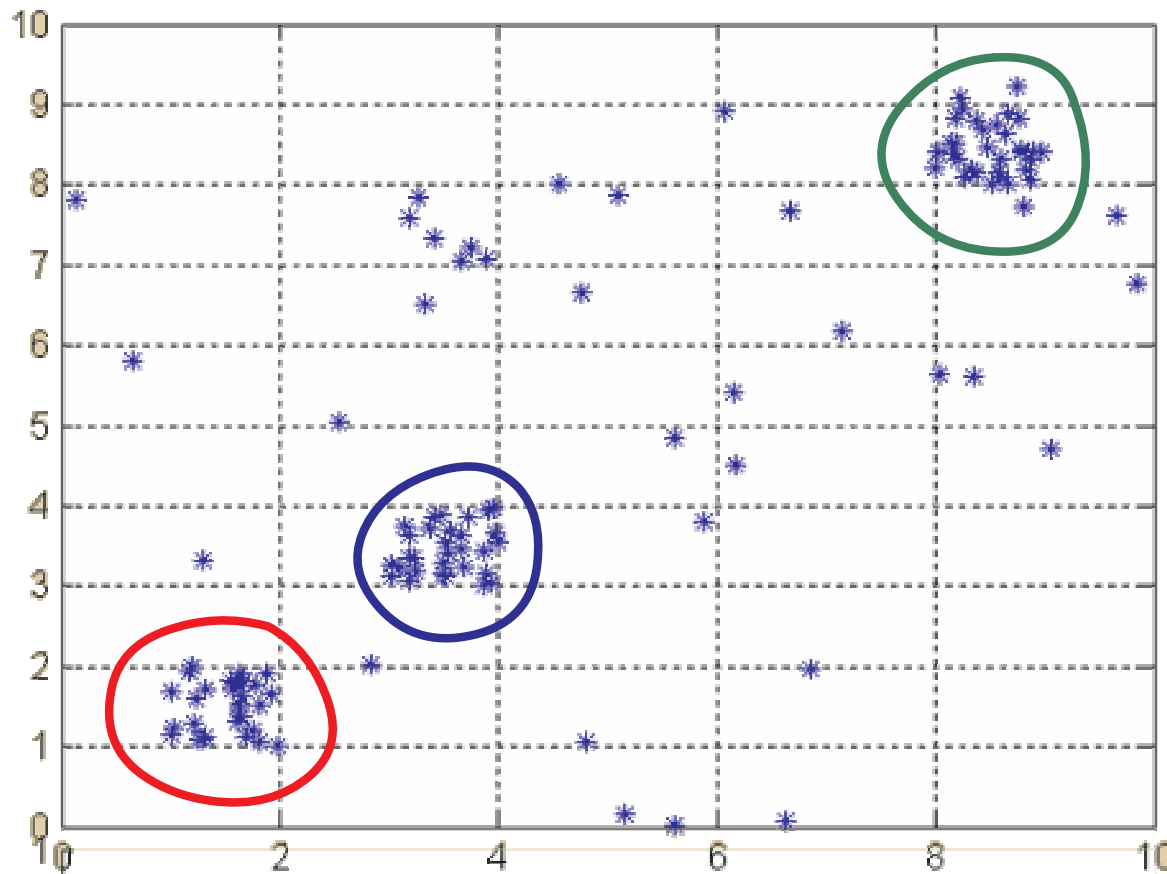
Fitness sharing (2/2)

- la tecnica del **fitness sharing** implementa il meccanismo di azione negativa all'affollamento attraverso una modifica della OF
 - individui molto vicini tra loro vedono ridotta la loro fitness
 - regioni molto affollate diventano così meno promettenti e l'algoritmo può fare emergere altre zone
- la modifica rende quindi meno conveniente l'exploitation a favore di una maggiore exploration dello spazio



Clustering (1/3)

- la tecnica del *clustering* si propone di mettere in evidenza l'addensamento degli individui in alcune regioni dello spazio



Clustering (2/3)

- il numero di individui per *cluster* puo' essere ridotto favorendo cosi' una parita' di trattamento a diverse regioni
- scegliendo ad esempio un solo individuo per ogni cluster la procedura di selezione della mating pool puo' prendere in considerazione individui la cui fitness e' non elevata ma che rappresentano potenzialmente una nuova specie emergente
- in entrambe le tecniche il prodotto del metodo non e' un singolo individuo, bensì una distribuzione di individui su piu' regioni



Clustering (3/3)

- funzione multiJamba

$$f(\mathbf{x}) = -1 - \sum_{j=1}^{N_{dim}} \left(x_j^2 + \cos(2\pi \cdot x_j) \right)$$

