

# Intelligenza Artificiale

## Calcolo evolutivo

Marco Piastra

# Evoluzione secondo Darwin

- **Individui e popolazione**

- Un **popolazione** consiste di un insieme di **individui** diversificati
- La **ricombinazione** (riproduttiva) di caratteristiche individuali che risultano più adatte all'ambiente tende a diventare prevalente nella popolazione

Gli individui sono le "unità di selezione"

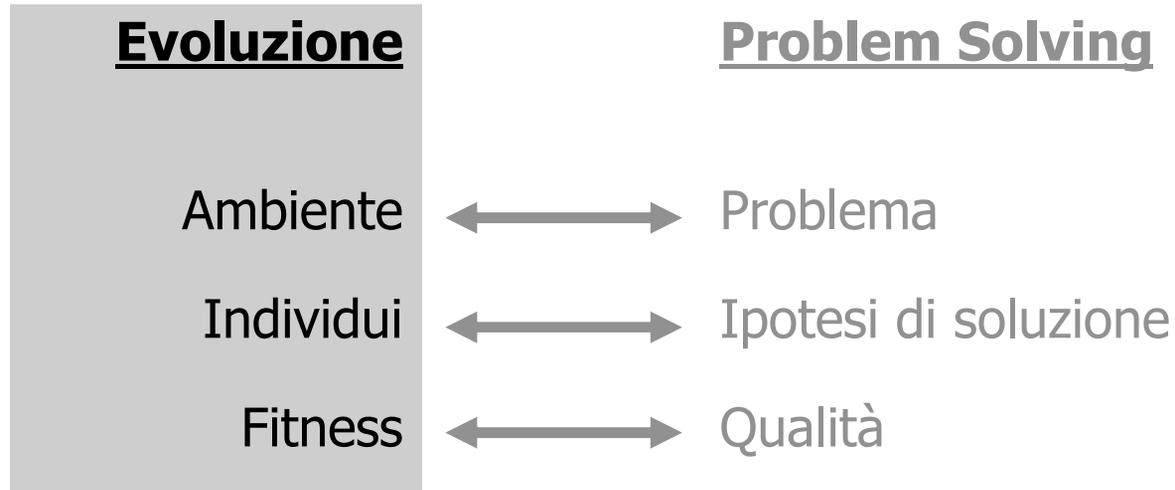
- **Ricombinazione e caso**

- Le variazioni casuali, dovute alle **mutazioni** spontanee, garantiscono una sorgente costante di diversità e preservano il potenziale evolutivo della popolazione

Le popolazioni sono le "unità di evoluzione"

- Si noti l'assenza di una *supervisione esterna*
  - l'evoluzione è una sorta di processo spontaneo

# Evoluzione biologica come algoritmo



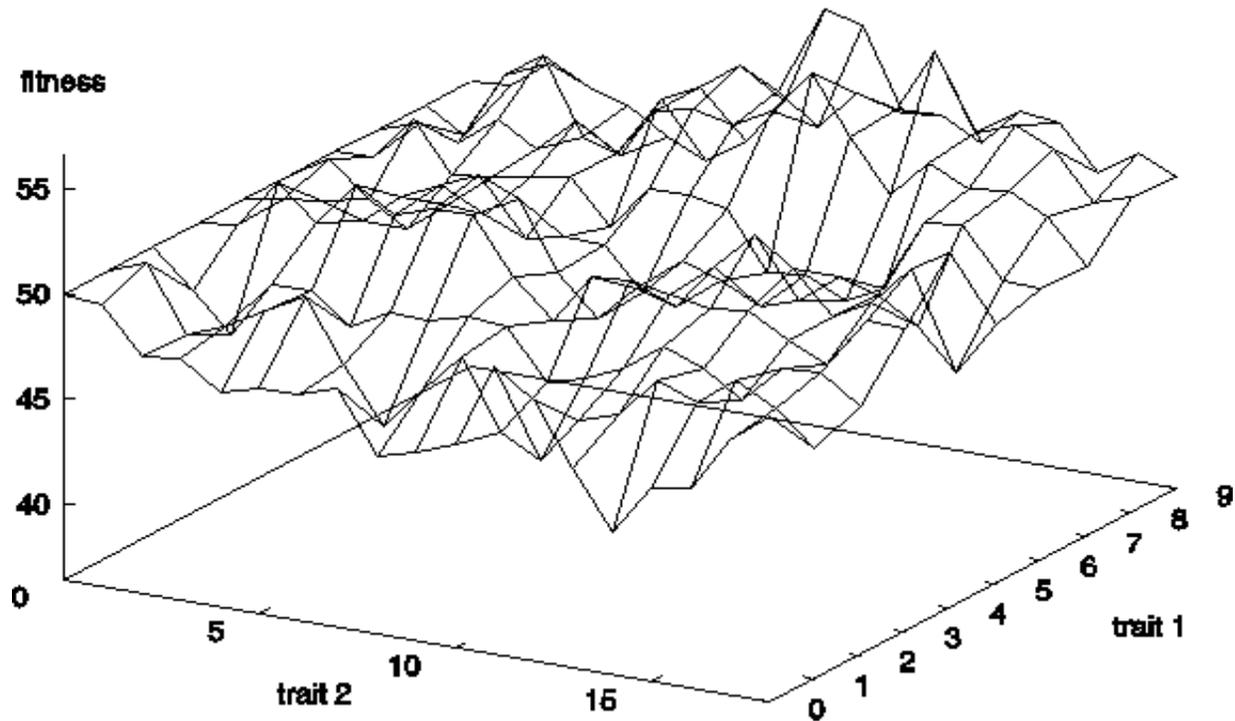
- **Analogie**

- **Fitness** → possibilità di sopravvivenza e riproduzione
- **Qualità** → bontà della soluzione

- **Differenze**

- **Popolazione** (di *individui*) → insieme di ipotesi di soluzione
- **Processo evolutivo** (della *popolazione*)  
→ migrazione progressiva della *popolazione* verso più alti livelli di **fitness**

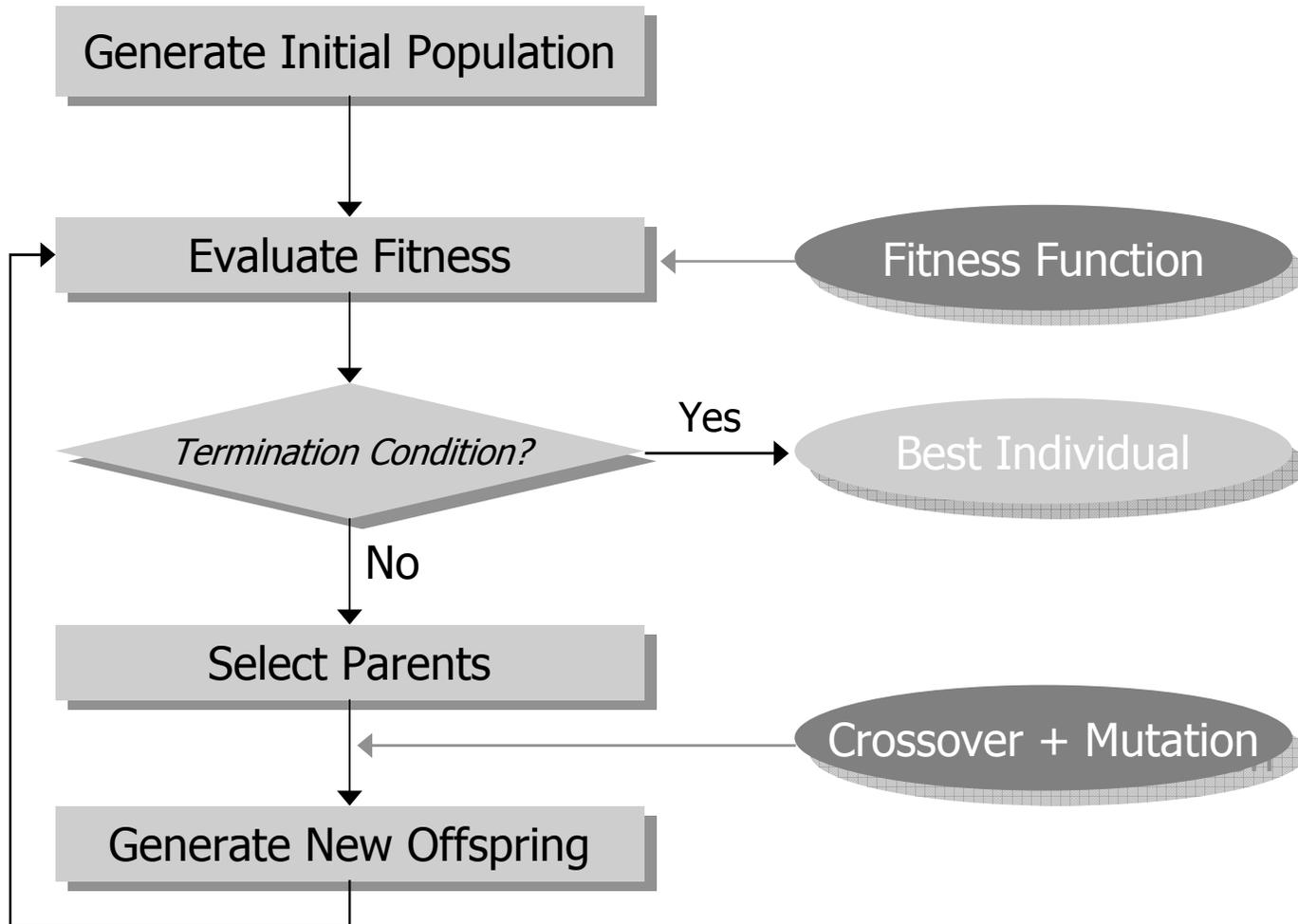
# EC – Fitness landscape



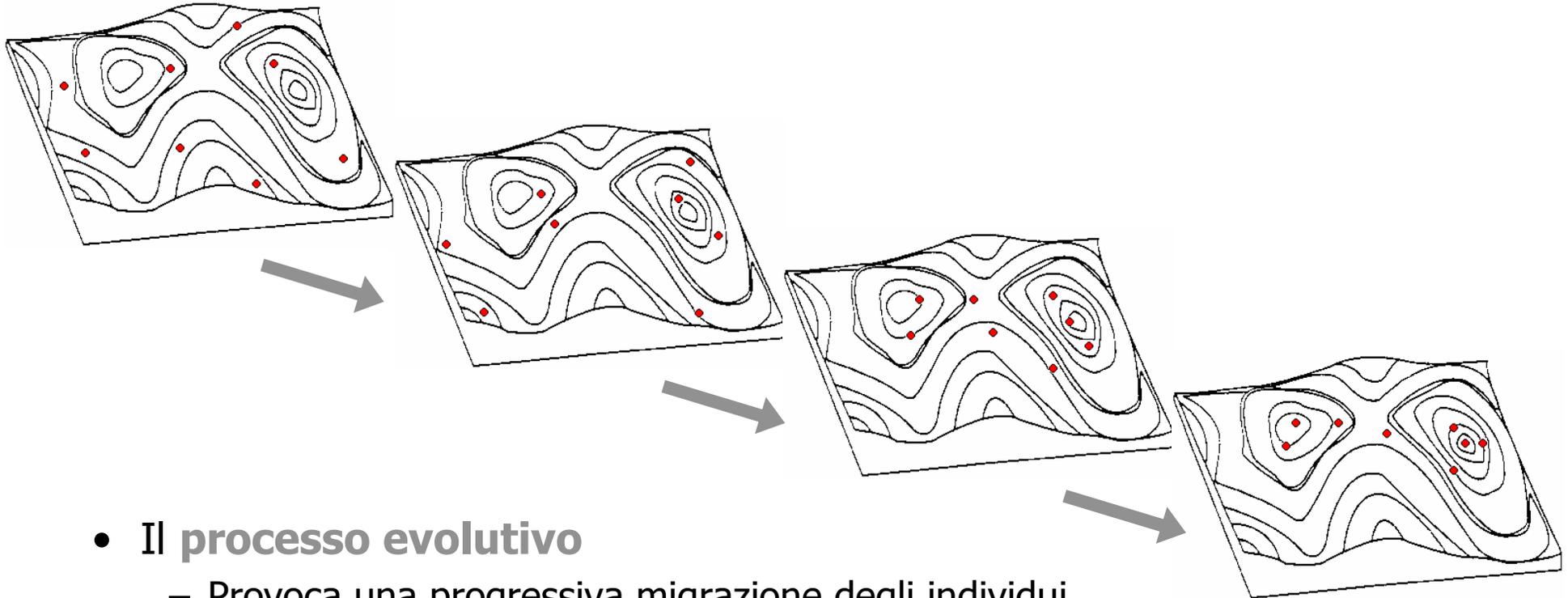
- **Ambiente come 'fitness landscape'**

- Descritto dai valori della funzione di fitness in funzione delle caratteristiche delle possibili soluzioni (individui)
- A ciascun individuo corrisponde un punto nel 'landscape'

# Struttura generale del processo



# Evoluzione come migrazione

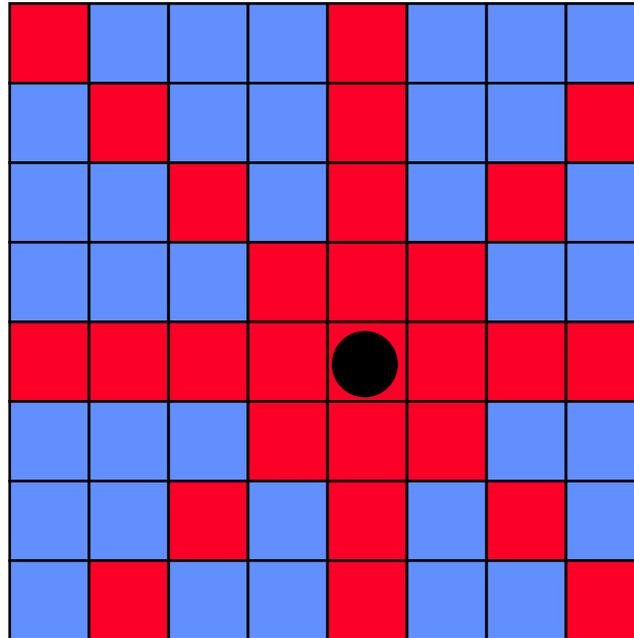


- **Il processo evolutivo**
  - Provoca una progressiva migrazione degli individui verso zone a fitness più elevata
- **Attenzione:** è la popolazione che migra, non gli individui
  - EC vs. Learning

# Cenni storici

- 1948, Turing:
  - propone la “*genetical or evolutionary search*”
- 1962, Bremermann
  - **ottimizzazione** tramite *evoluzione e ricombinazione*
- 1964, Rechenberg
  - introduce le **evolution strategies**
- 1965, L. Fogel, Owens and Walsh
  - introduce lo **evolutionary programming**
- 1975, Holland
  - introduce i **genetic algorithms**
- 1992, Koza
  - introduce il **genetic programming**

# Esempio: le 8 regine



- **Problema**

- Piazzare 8 regine su una scacchiera 8 x 8 in modo che nessuna possa attaccare l'altra

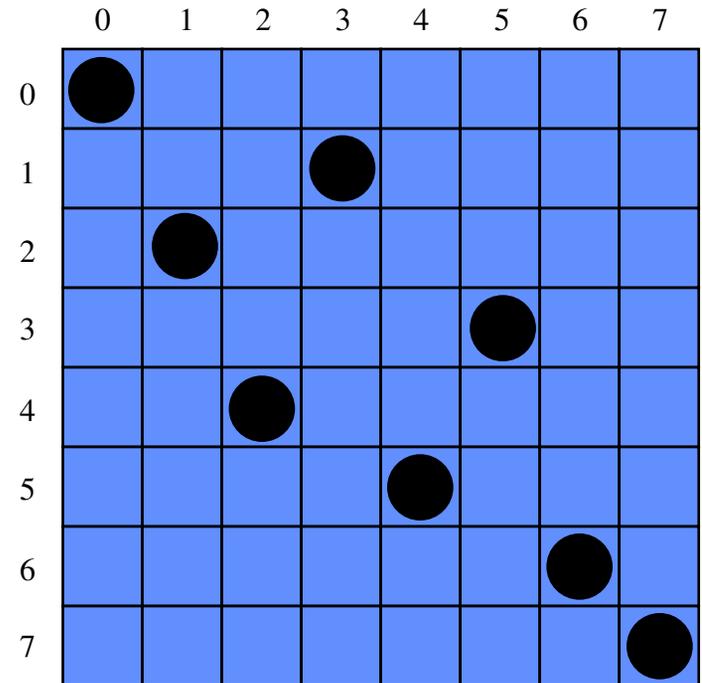
# Genotipo e fenotipo

- **Fenotipo**

- L'individuo completo che interagisce con l'ambiente
- Di cui si può valutare la fitness
  - In questo caso, una disposizione delle 8 regine sulla scacchiera

- **Genotipo**

- Il 'progetto genetico' dell'individuo
  - In questo caso, una sequenza di otto cifre, da 0 a 7
- Su cui agiscono gli operatori genetici



↑ **Espressione**

Genotipo

0 2 4 1 5 3 6 7

(Rappresentazione binaria)

000 010 100 001 101 011 110 111

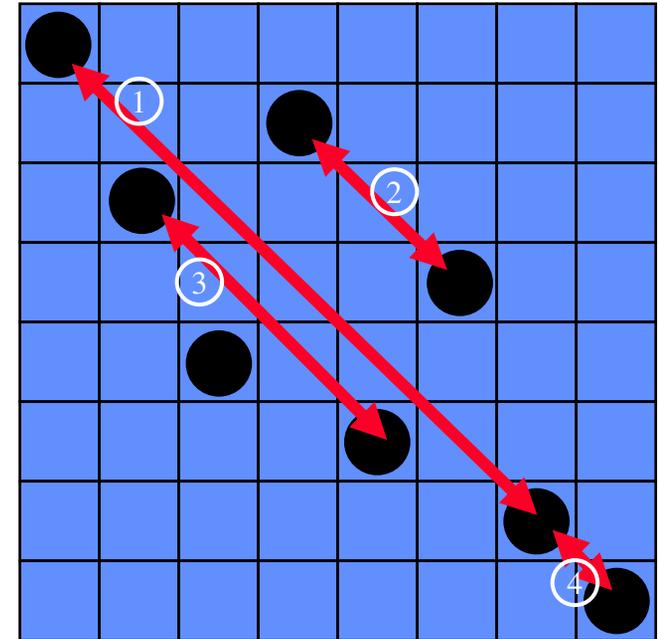
# Fitness

## • Problema

- Prevenire i possibili attacchi
- Penalità singola (di un'ipotesi di soluzione)
  - Un possibile attacco di due regine
- Penalità complessiva
  - Somma di tutte le penalità singole

## • Fitness

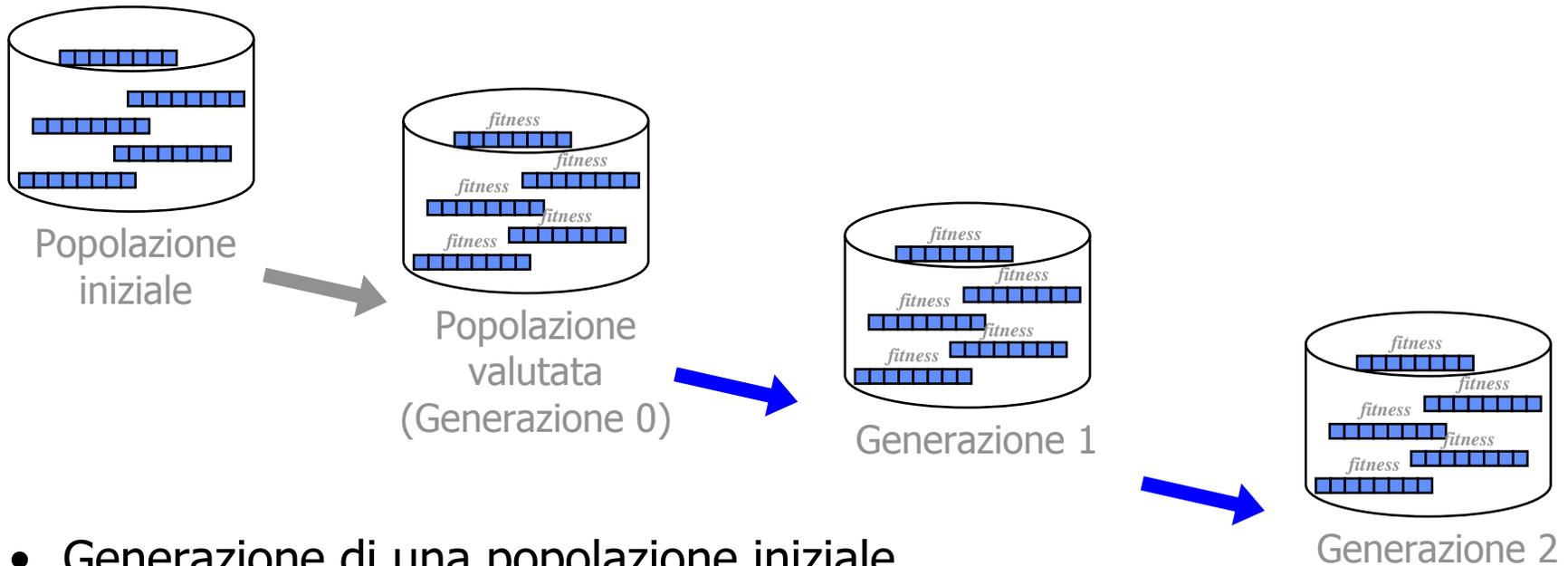
- Di una possibile soluzione (individuo):  
 $f = 8 - \text{la penalità complessiva}$ 
  - Nel caso in figura, la *fitness* del genotipo è 4



$$\text{fitness: } 8 - 4 = 4$$

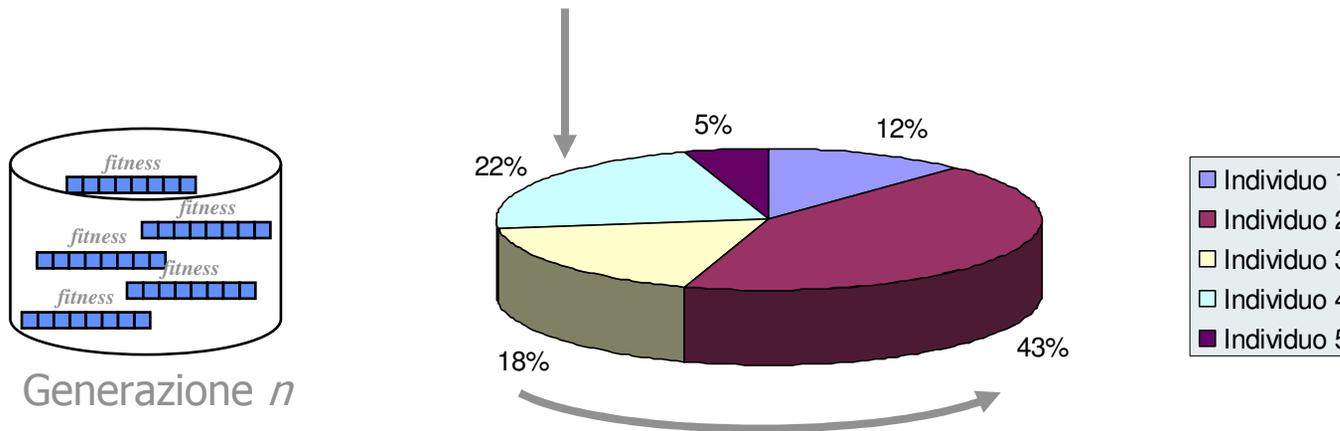
0	2	4	1	5	3	6	7
---	---	---	---	---	---	---	---

# Popolazioni e generazioni



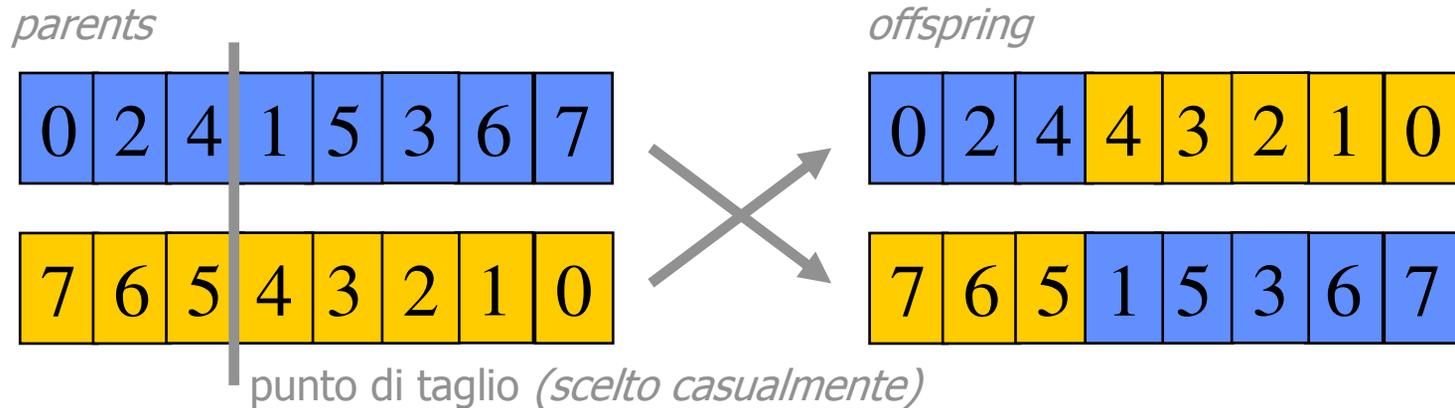
- Generazione di una popolazione iniziale
  - M permutazioni generate a caso (anche con duplicazioni)
- Valutazione della *fitness* di ciascun individuo
  - Si ottiene la generazione 0
- Attivazione del processo evolutivo
  - Produzione iterativa di ulteriori generazioni

# Selezione

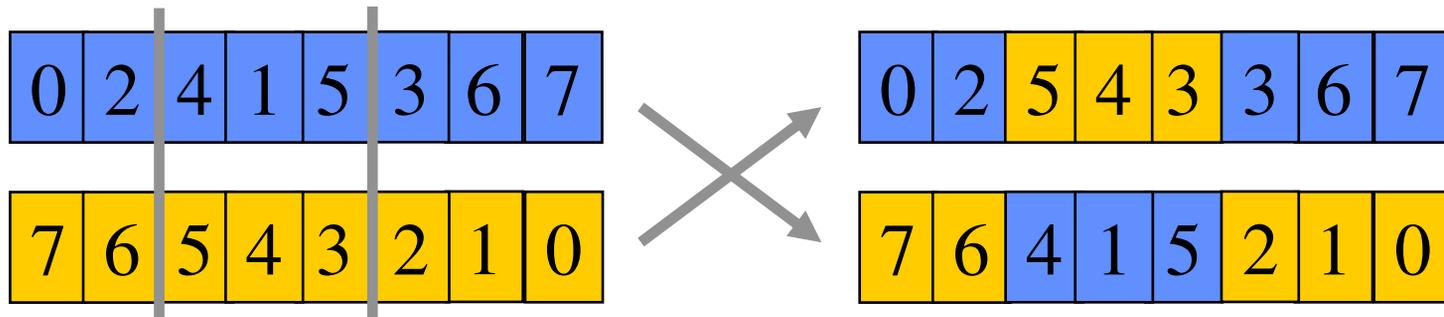


- Ruota della roulette (*Roulette wheel*)
  - A ciascun individuo si assegna un settore della roulette
  - L'ampiezza del settore è proporzionale alla *fitness*
    - Tipicamente, ampiezza  $\sim f / f_{\text{avg}}$
    - Migliore è la *fitness*, più largo il settore
  - La probabilità di selezione è quindi più alta quanto migliore è la *fitness*

# Ricombinazione (Crossover)



- Generazione di due nuovi individui a partire da due individui esistenti
  - Si seleziona un punto di *crossover*
  - Si incrociano le parti dei genotipi
  - Alternativa: due punti di taglio



# Mutazione

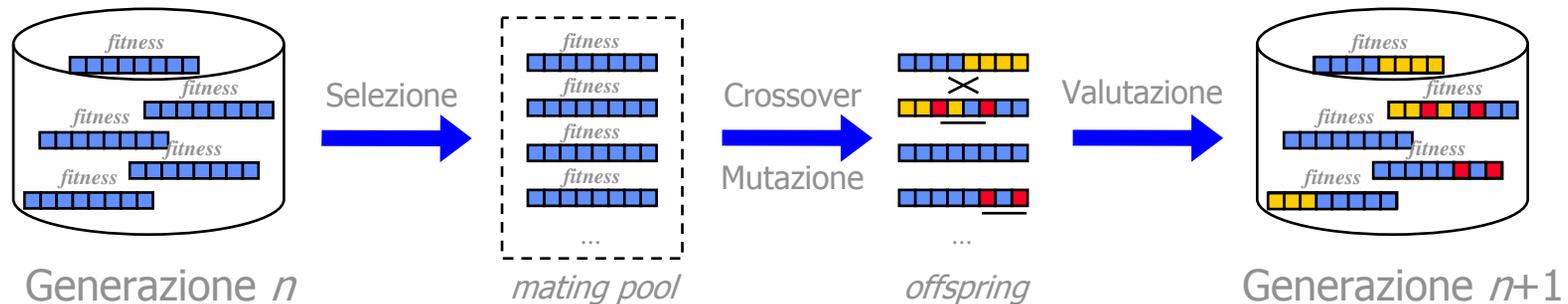


- Variazione casuale di un singolo individuo
  - alterazione di una cifra
  - Alternativa: inversione di due cifre



- Altra alternativa: ciascuna cifra viene mutata con probabilità  $p_m$

# Mating pool



- **Pseudo-algoritmo:**

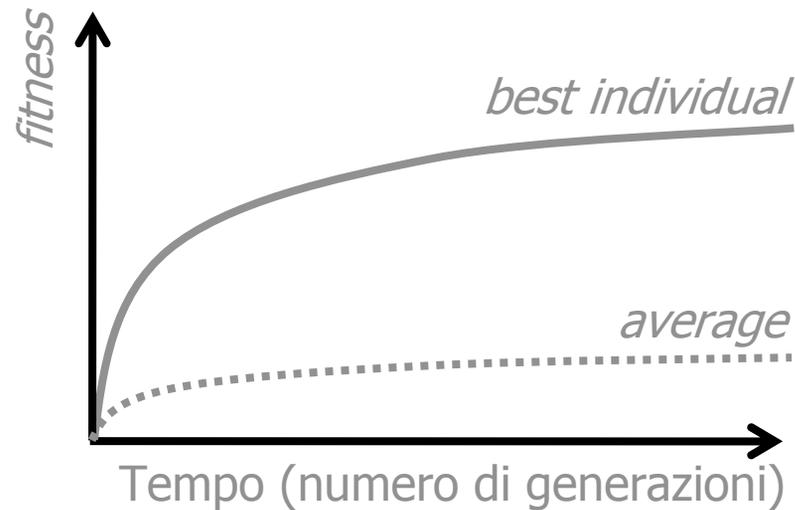
- 1) INPUT: Generazione  $n$  (di  $M$  individui)
- 2) Selezione di  $M$  individui (*roulette wheel*)
- 3) Per ciascuna coppia di individui (*parents*)
  - applicazione del *crossover* con probabilità  $p_C$
  - applicazione della *mutazione* (*all'offspring*) con probabilità  $p_M$
  - inserimento dei risultati nella Generazione  $n+1$
- 4) Valutazione della Generazione  $n+1$  (*fitness*)

# Processo evolutivo

- Un processo stocastico
  - Selezione casuale degli individui come *parents*
  - Identificazione casuale del punto di *crossover*
  - Applicazione casuale della mutazione all'*offspring*
- Cui si aggiunge un effetto 'deriva' causato dalla *fitness*
  - L'unico elemento condizionante è la **selezione**
    - Nella selezione si 'bilancia' il caso (*roulette*) ed il determinismo (*fitness*)
    - Si determina così una tendenza migratoria della popolazione verso i picchi del *fitness landscape*

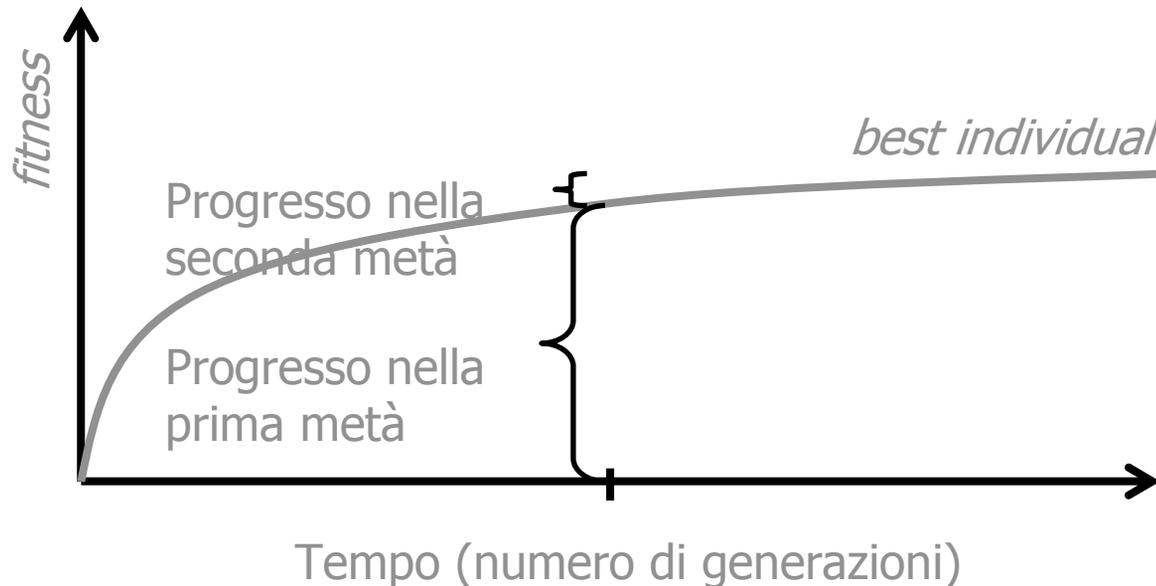


# Tipico andamento del processo evolutivo



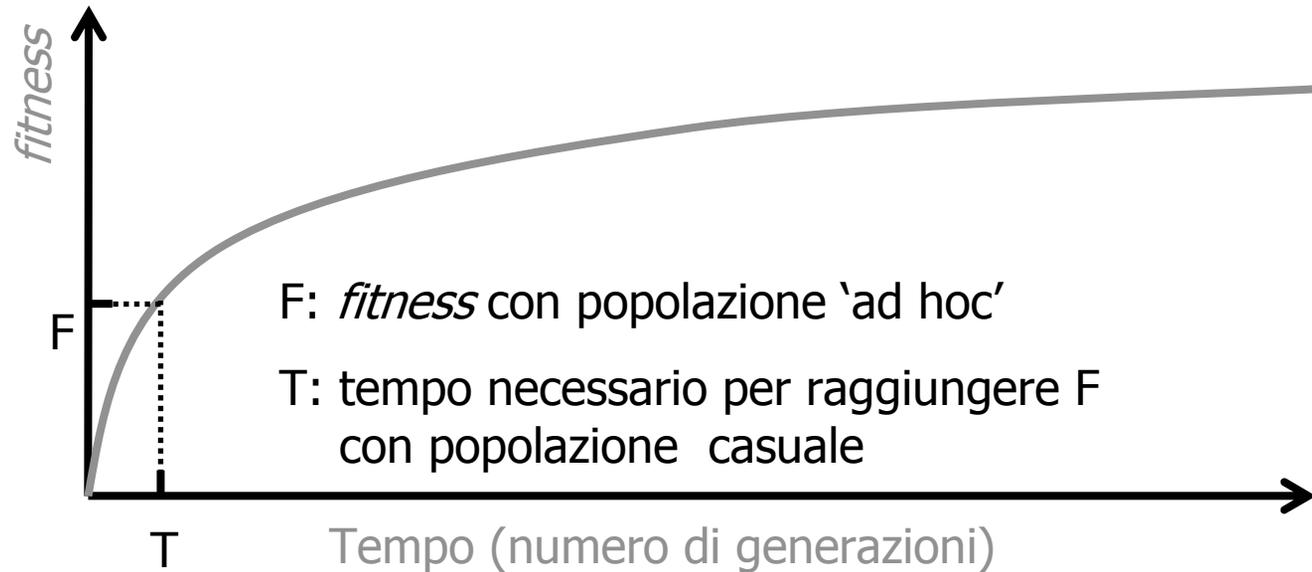
- L'effetto 'deriva' produce un miglioramento progressivo
  - *best fitness*, cioè del miglior individuo della generazione
    - migliora rapidamente
  - fitness media della generazione
    - migliora molto più lentamente
- Supervisione: fitness + criterio di terminazione
  - raramente, nella pratica, si ottiene un individuo ottimo

# Condizioni di terminazione



- Quando non si trova un individuo ottimo
  - Attendere ulteriormente (aumento del numero di generazioni) non sempre porta miglioramenti significativi dei risultati

# Popolazione iniziale



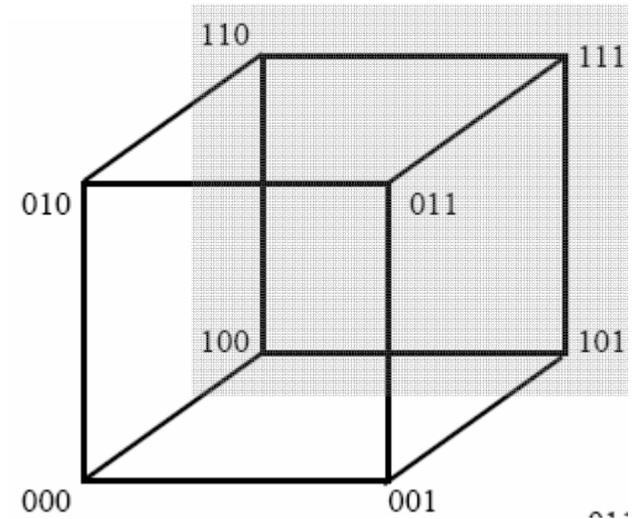
- In generale, la generazione *casuale* della popolazione iniziale è una buona scelta
  - L'iniziale miglioramento della fitness è rapido
  - Spesso lo sforzo di creare popolazioni iniziali più specifiche non è compensato da risultati apprezzabili

# Varianti del calcolo evolutivo

- Rappresentazione degli individui
  - vettori di bit, di interi di numeri reali
  - grafi
  - alberi
  - dimensione fissa, limitata o variabile
- Operatori genetici
  - mutazione, crossover
  - operatori speciali (dipendenti dalla rappresentazione)
- Metodi di selezione
  - Roulette (fitness proportionate)
  - Tournament
- Processo evolutivo
  - Generation-based
  - Steady state
- Fitness
  - scalare
  - multi-valore (multi-obiettivo)

# Come funziona: schemi e iperpiani

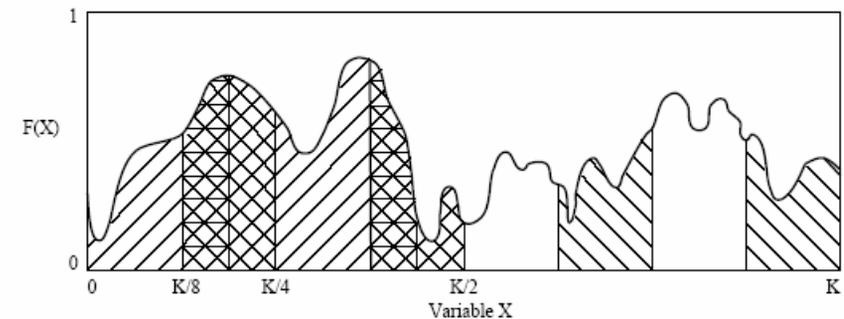
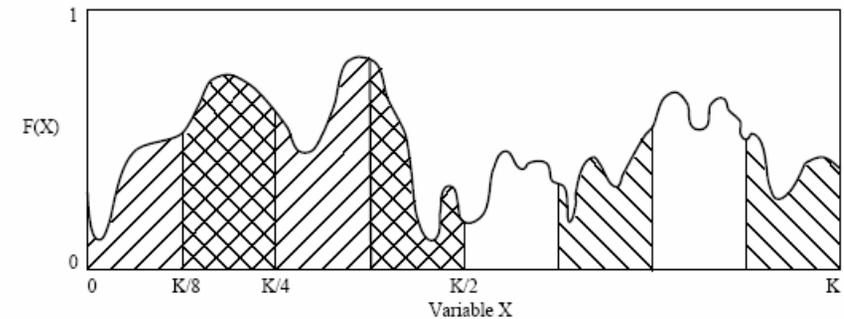
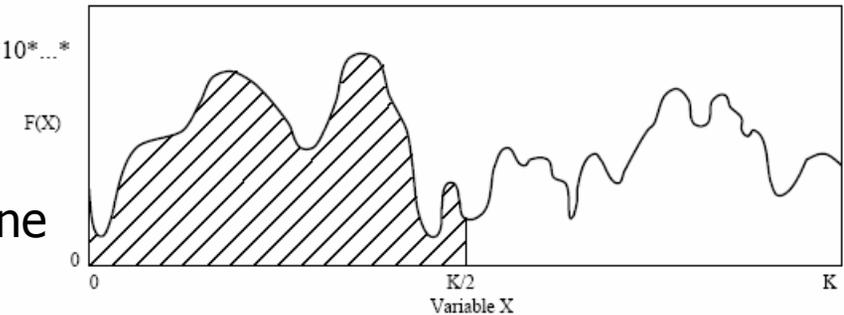
- Spazi di ricerca
  - Si consideri una popolazione di stringhe binarie di lunghezza tre
  - Lo spazio di ricerca (tutti i possibili individui) può essere rappresentato come un **cubo**
- **Schemi**
  - uno schema come "1\*\*" descrive tutti gli individui aventi un 1 al primo posto
  - vale a dire, un piano del cubo
  - il piano complementare è "0\*\*"
- Ordine degli schemi e generalizzazione
  - "1\*\*" descrive un piano di ordine 1, mentre p.es. "\*01" descrive un piano di ordine 2 (**ordine** = numero di bit definiti)
  - generalizzando, stringhe binarie di lunghezza qualsiasi corrispondono ad un **ipercubo** di pari dimensioni
  - ogni schema descrive un **iperpiano**



# Schemi, campionamento e fitness



- Schemi e campionamento
  - Una popolazione contiene un campione di ciascuno schema
  - Si immagina che la ricerca avvenga su più iperpiani in parallelo
- Fitness di uno schema
  - E' il valor medio delle fitness del campione (i.e. tutti gli individui appartenenti all'iperpiano)
- Selezione
  - La strategia di selezione premia (con una maggiore presenza nel mating pool) i campioni a fitness più elevata



# Campioni e selezione

- Esempio

Schemata and Fitness Values									
Schema	Mean	Count	Expect	Obs	Schema	Mean	Count	Expect	Obs
101*...*	1.70	2	3.4	3	*0**...*	0.991	11	10.9	9
111*...*	1.70	2	3.4	4	00**...*	0.967	6	5.8	4
1*1*...*	1.70	4	6.8	7	0***...*	0.933	12	11.2	10
*01*...*	1.38	5	6.9	6	011*...*	0.900	3	2.7	4
**1*...*	1.30	10	13.0	14	010*...*	0.900	3	2.7	2
*11*...*	1.22	5	6.1	8	01**...*	0.900	6	5.4	6
11**...*	1.175	4	4.7	6	0*0*...*	0.833	6	5.0	3
001*...*	1.166	3	3.5	3	*10*...*	0.800	5	4.0	4
1***...*	1.089	9	9.8	11	000*...*	0.767	3	2.3	1
0*1*...*	1.033	6	6.2	7	**0*...*	0.727	11	8.0	7
10**...*	1.020	5	5.1	5	*00*...*	0.667	6	4.0	3
*1**...*	1.010	10	10.1	12	110*...*	0.650	2	1.3	2
****...*	1.000	21	21.0	21	1*0*...*	0.600	5	3.0	4
					100*...*	0.566	3	1.70	2

- In generale

$$M(H, t + \textit{intermediate}) = M(H, t) \frac{f(H, t)}{\bar{f}}$$

$M(H, t)$  è la dimensione del campione dell'iperpiano  $H$  alla generazione  $t$

$t + \textit{intermediate}$  è la selezione (mating pool)

$f(H, t)$  è la fitness media del campione di  $H$  alla generazione  $t$

$\bar{f}$  è la fitness media del campione

# Crossover, mutazione e schemi

- Crossover e vulnerabilità degli schemi
  - Un crossover che ‘taglia a metà’ uno schema lo sconvolge
  - Schemi più lunghi sono quindi più vulnerabili
  - La **defining length**  $\Delta(H)$  di uno schema è la distanza massima tra i bit definiti
    - $\Delta(H)$  di “\*\*10\*\*\*\*” è 2
    - $\Delta(H)$  di “\*\*10\*0\*\*” è 4
    - $\Delta(H)$  di “1\*10\*\*0\*” è 7
  - La probabilità che il crossover ‘tagli’ uno schema è quindi  $p_c \frac{\Delta(H)}{L-1}$ 
    - dove  $L$  è la lunghezza degli individui e  $p_c$  la probabilità di occorrenza del crossover
- Mutazione e schemi
  - Detto  $o(H)$  l’ordine dello schema  $H$  (il numero di bit predefiniti)
  - La probabilità che lo schema sopravviva intatto è  $(1 - p_m)^{o(H)}$ 
    - dove  $p_m$  è la probabilità che uno dei bit definiti lo schema venga mutato (per comodità di definizione)

# Schema Theorem (Holland, 1975)

- Probabilità e generazioni
  - Definita come  $P(H,t)$  la frequenza relativa dei campioni di  $H$  nella generazione  $t$  (i.e. una probabilità)
  - Il teorema stabilisce una relazione tra  $P(H,t)$  e  $P(H,t + 1)$

| Frequenza relativa alla generazione  $t + 1$

$$P(H, t + 1) \geq P(H, t) \frac{f(H, t)}{\bar{f}} \left[ 1 - p_c \frac{\Delta(H)}{L - 1} \left( 1 - P(H, t) \frac{f(H, t)}{\bar{f}} \right) \right] (1 - p_m)^{o(H)}$$

| Probabilità di sopravvivenza a crossover e mutazione

| Proporzione attesa per la selezione

| Frequenza relativa alla generazione  $t$

# Caratteristiche del calcolo evolutivo

- Processo evolutivo
  - Convergente (nel senso di crescita degli schemi a fitness più alta)
    - data una ragionevole scelta di  $p_c$  e  $p_m$
  - Non c'è garanzia di convergenza verso un ottimo globale
    - Esistono risultati più precisi dello Schema Theorem
      - Ma nessuno comporta la convergenza ad una soluzione
  - Metodo di ottimizzazione
    - Approssimato: manca la garanzia di convergenza
    - *Anytime*: in qualsiasi momento si può ottenere una soluzione approssimata
    - Robusto: poco influenzato dai massimi locali
- Operatori genetici ed individui
  - Gli schemi **corti** e **compatti** 'sopravvivono' meglio
    - Minore probabilità di 'taglio' da parte del crossover
    - Minore probabilità di sconvolgimento da parte della mutazione
  - Gli schemi devono avere una lunghezza sufficiente
    - Che permetta loro di esprimere un carattere significativo