

Alessandro Bollini
bollini@vision.unipv.it

Dipartimento di Informatica e Sistemistica
Università di Pavia
Via Ferrata, 1
27100 Pavia

Programmazione Genetica



Problem-solving e programmi

- Le attività di problem-solving possono essere formalizzate come ricerca di un programma.

Classificazione	Verifica qualità
Controllo	Controllo di impianti
Identificazione	Analisi di serie temporali
Pianificazione	Controllo di un robot

L'idea

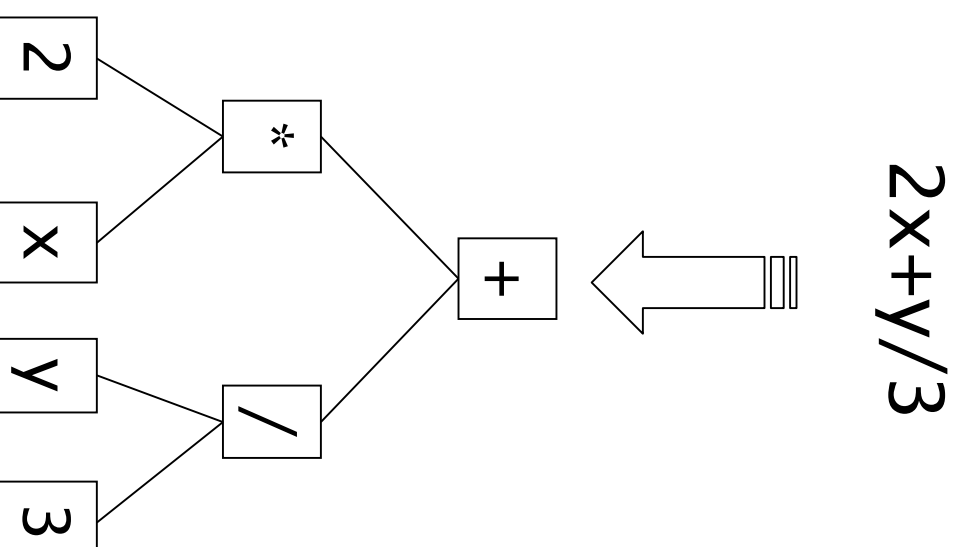
- Problem-solving come ricerca in uno spazio di programmi.
- Algoritmi genetici come ottimizzazione in uno spazio di soluzioni.
- Affrontare un problema come identificazione di un programma effettuata tramite tecniche evolutive.
- Cramer (1985, assembler), Koza (1998, LISP).

Programmazione genetica

- Algoritmo di base mutuato dagli algoritmi genetici.
- Gli individui sono programmi rappresentati come S-Expression, che hanno contemporaneamente il ruolo di genotipo e di fenotipo.
- Lo spazio di ricerca è molto diverso.
- Come si creano, valutano, modificano?

S-Expression

- **Funzioni.**
- **Terminali:**
 - Valori.
 - Variabili.
- **Binding:**
 - Contesto.
 - Sensori e attuatori.
- **Valore.**



Inizializzazione

- Definizione function/terminal set.
- Non esiste una lunghezza fissa:
 - Limite di profondità.
 - Costruzione full e grow.
- Diversità della popolazione iniziale:
 - Inizializzazione ramped half-and-half.

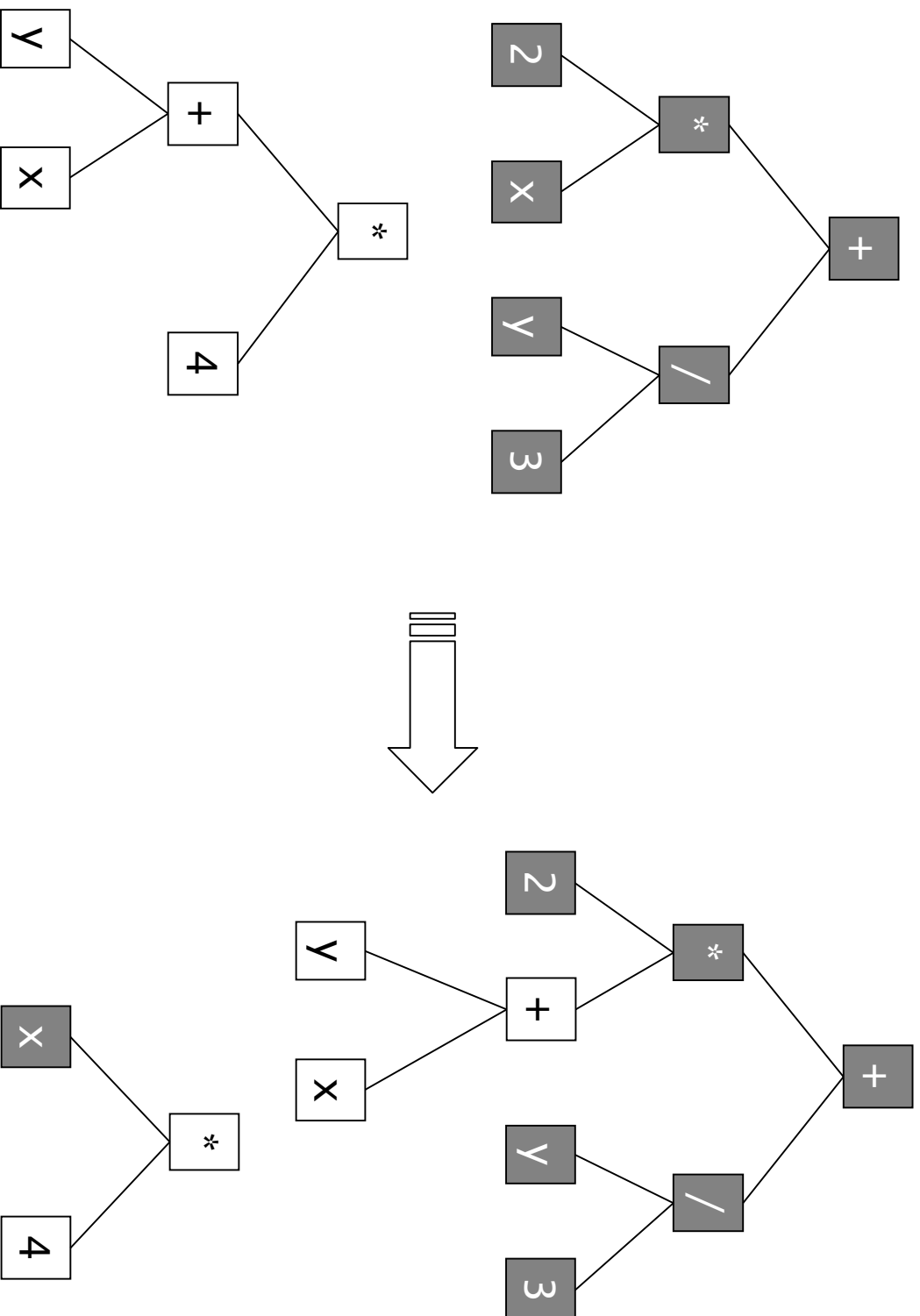
Valutazione

- Non esiste conversione genotipo-fenotipo (almeno nella versione di base...).
- Esecuzione, se necessario in un contesto (binding, sensori e attuatori...).
- Lettura dei risultati (valori o stati), wrapping.
- Uso dei risultati.

Operatori

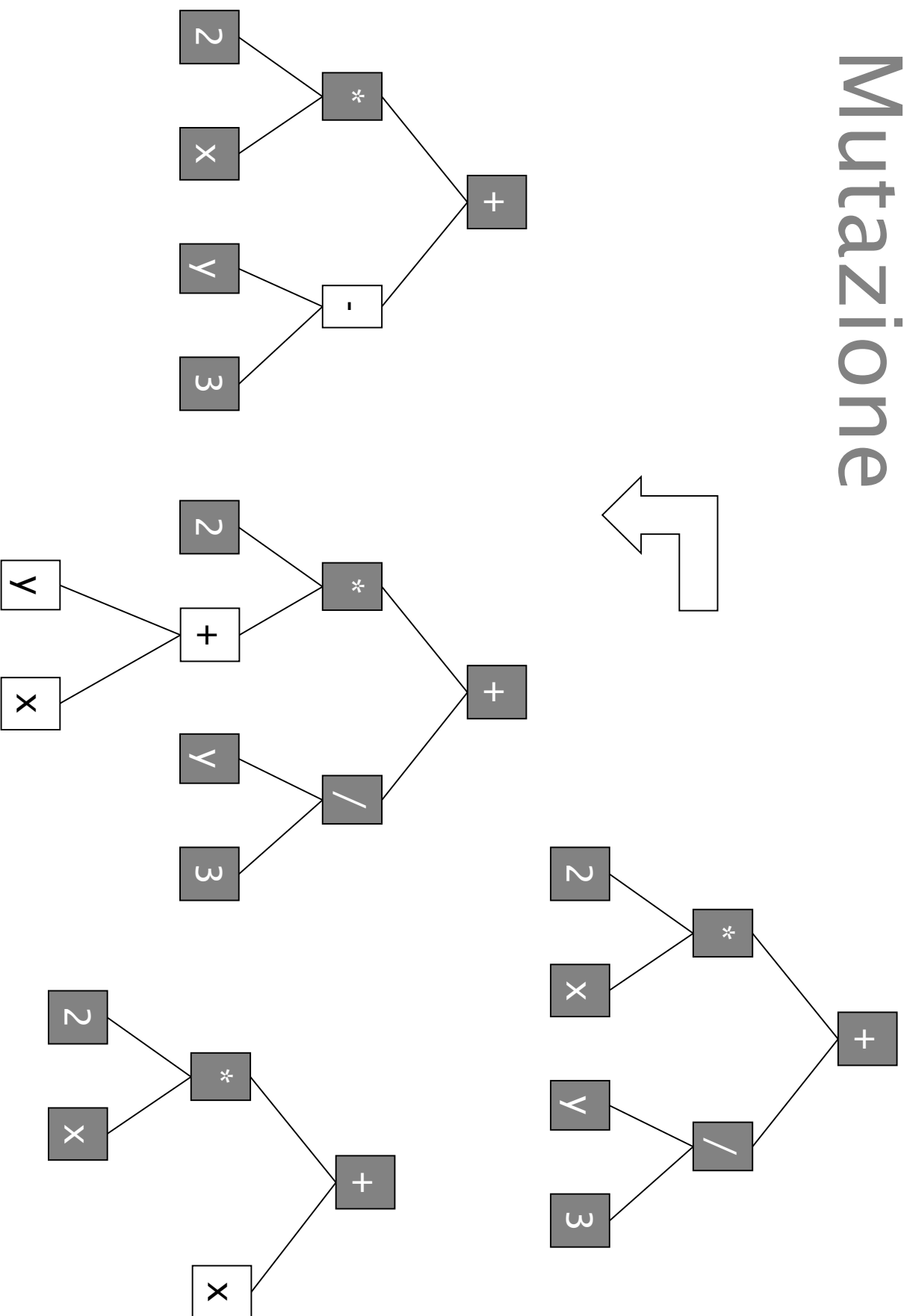
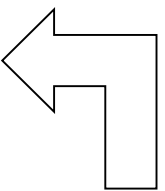
- Riproduzione.
- Ricombinazione.
- Mutazione.
- Editing.

Ricombinazione



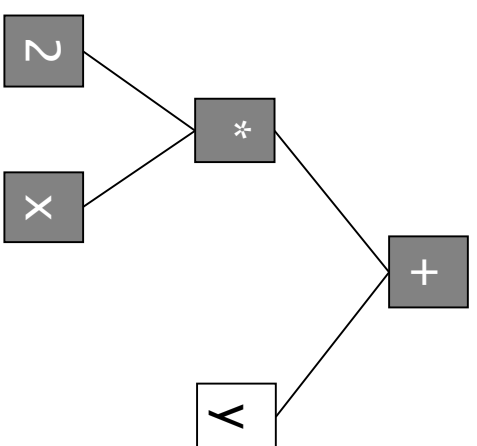
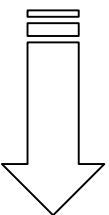
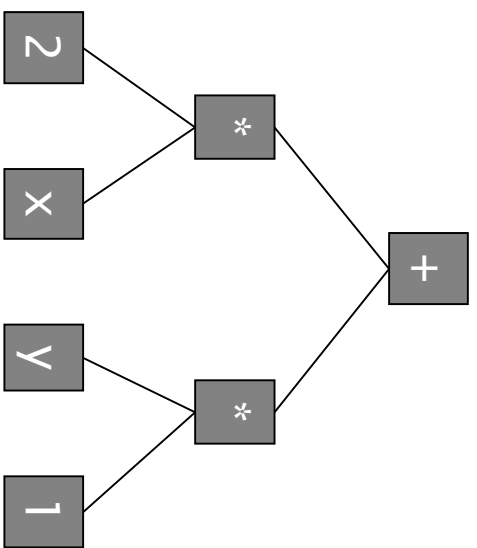
Programmazione Genetica

Mutazione



Programmazione Genetica

Editing



Programmazione Genetica

Altri elementi

- **Integrità:**
 - Operazioni protette, eccezioni.
 - Chiusura funzionale.
 - Programmazione strongly-typed.
 - Grammatiche.
- **Costanti (ERC, ER Data).**
- **Programmazione strutturata.**
- **Memoria.**

Impostazione

- Obiettivo.
- Function/terminal set.
- Contesto esecutivo: dati in ingresso/uscita, effetti collaterali, wrapping.
- Funzione di valutazione delle prestazioni.
- Criteri di terminazione.
- Configurazione del processo: dimensione della popolazione, probabilità di applicazione degli operatori, ...
- Ingegnerizzazione.

Vantaggi

- Adattamento dinamico della struttura alla forma generale ed alla complessità del problema.
- Forma simbolica:
 - Analisi delle soluzioni per capire la strategia adottata.
 - Identificazione dei fattori rilevanti.
- Possibilità di ingegnerizzazione.

Limiti

- Stocasticità, non provabilità dell'ottimo.
- Dinamica mal compresa, specialmente negli aspetti specifici delle strutture variabili.
- Integrazione con la programmazione in grande.
- Forme funzionali limitate e gestione di spazi di ricerca relativamente piccoli.

Conclusioni

- Adattando le idee di base degli algoritmi genetici all'evoluzione di strutture dati variabili è possibile esplorare lo spazio delle soluzioni di class di problemi apparentemente diversi.
- Soluzione valida per applicazioni di nicchia.