

## Genetic Programming (2) Regressione Simbolica

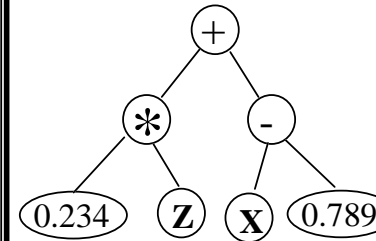
**Problema:** Dato un insieme di dati  $\{(x_1, y_1) \dots (x_n, y_n)\}$  trovare la funzione che meglio approssima i dati.

**Terminali:**  $\{X\}$

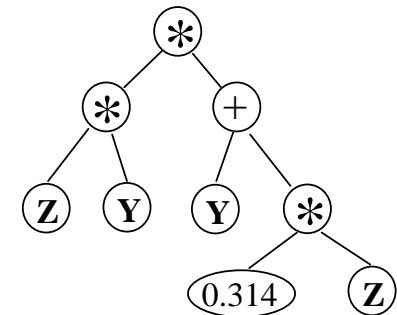
**Funzioni Primitive:**  $\{+, -, *, /, \sin, \cos, \exp, \log\}$

**Funzione di Fitness:** Somma dei valori assoluti degli errori fra gli  $y_i$  e gli  $S(x_i)$ .

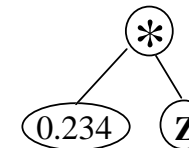
## Genetic Programming (3) - Crossover



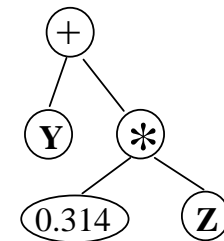
$$0.234 \cdot Z + X - 0.789$$



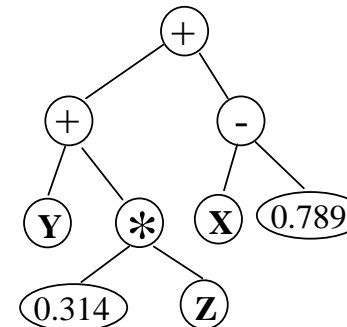
$$Z \cdot Y \cdot (Y + 0.314 \cdot Z)$$



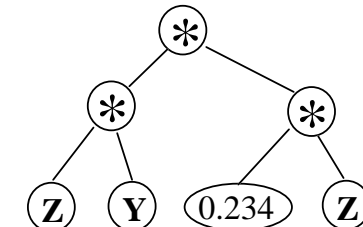
$$0.234 \cdot Z$$



$$Y + 0.314 \cdot Z$$



$$Y + 0.314 \cdot Z + X - 0.789$$



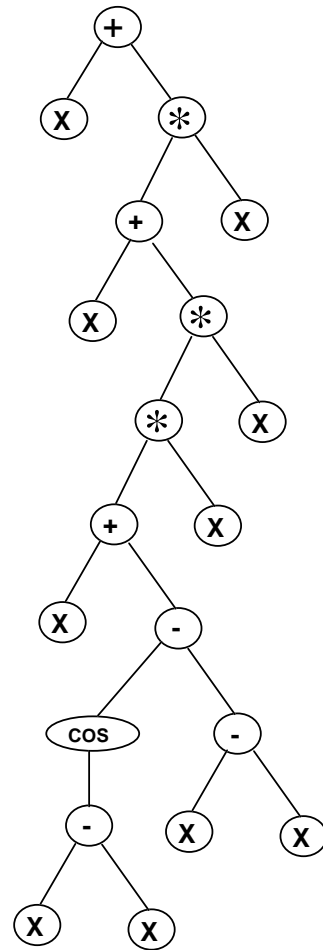
$$Z \cdot Y \cdot 0.234 \cdot Z$$

## Genetic Programming (4) Esempio di Risultato

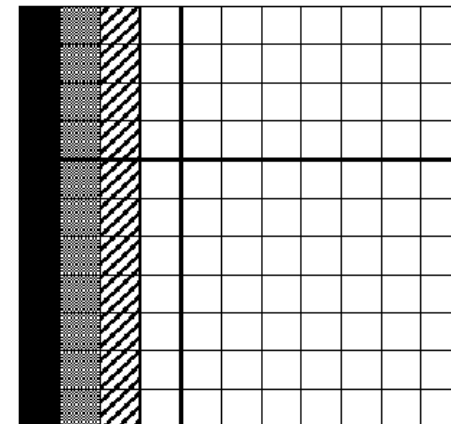
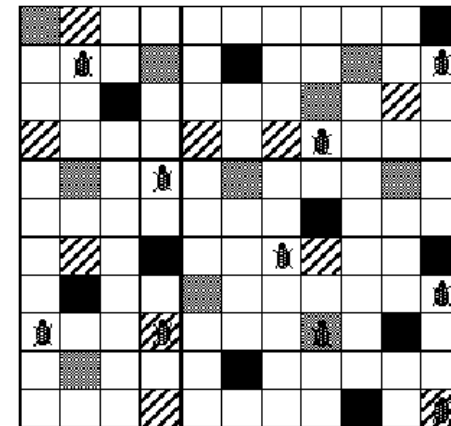
Dati generati dalla funzione:  $x^4 + x^3 + x^2 + x$

Applichiamo l'algoritmo per 34 cicli e otteniamo come migliore soluzione la S-espressione:

$(+ X (* (+ X (* (* (+ X (- (\cos (- X X)) (- X X))) X) X)) X))$



## Esempio: Painted Desert Problem (1)



## Esempio: Painted Desert Problem (2)

### Terminali:

**X** = posizione orizzontale della formica (0 .. 9)

**Y** = posizione verticale della formica (0 .. 9)

**CARRY** = colore del grano di sabbia portato (0 .. 3)

**COLOR** = colore del grano di sabbia nella  
posizione corrente della formica (0 .. 3)

### Funzioni Primitive

**GO-N, GO-E, GO-S, GO-W**

**MOVE-RANDOM, PICK-UP**

**IFLTE(x, y, a, b)**

**IF  $x < y$  THEN a ELSE b**

**IFLTZ(x, a, b)**

**IF  $x < 0$  THEN a ELSE b**

**IF-DROP(a, b)**

**IF la formica trasporta un grano di sabbia AND  
non c'è sabbia nella posizione corrente**

**THEN lascialo AND a**

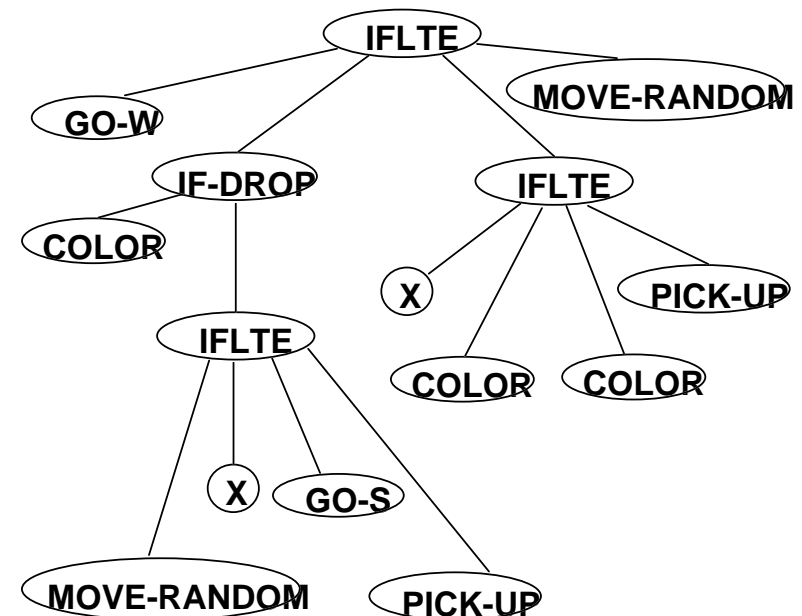
**ELSE b**

## Esempio: Painted Desert Problem (3)

### Fitness

**Somma della distanza dei grani di sabbia  
dalla loro posizione finale desiderata.**

### Una Possibile Soluzione



## Algoritmi Genetici Esempio: Hamburger Restaurant (1)

Come vendere la pizza in 4 ristoranti?

**Goal:** Massimizzare il profitto

Trovare i migliori valori possibili per le variabili:

Prezzo (Price) -> high/low (0/1)

Bevanda (Drink) -> wine/cola (0/1)

Servizio (Service)->leisurely&slow/fast&basic (0/1)

**Rappresentazione:** Un individuo rappresenta una possibile configurazione di un ristorante utilizzando tre parametri binari.

## Esempio: Hamburger Restaurant (2)

### Popolazione Iniziale

Restaurant number	Price	Drink	Speed	Binary representation
1	High	Cola	Fast	011
2	High	Wine	Fast	001
3	Low	Cola	Leisurely	110
4	High	Cola	Leisurely	010

### Valori della Funzione di Fitness Osservati

Generation 0		
i	String $X_i$	Fitness $f(X_i)$
1	011	3
2	001	1
3	110	6
4	010	2
Total		12
Worst		1
Average		3.00
Best		6

## Esempio: Hamburger Restaurant (3)

Riproduzione: Un risultato possibile.

Generation 0				Mating pool created after reproduction	
i	String $X_i$	Fitness $f(X_i)$	$\frac{f(X_i)}{\sum f(X_i)}$	Mating pool	$f(X_i)$
1	011	3	.25	011	3
2	001	1	.08	110	6
3	110	6	.50	110	6
4	010	2	.17	010	2
Total		12			17
Worst		1			2
Average		3.00			4.25
Best		6			6

Nella nuova popolazione ci sono:

- Due copie della stringa 3;
- Una copia delle stringhe 1 e 4;
- Sparisce la stringa 2.

## Esempio: Hamburger Restaurant (4)

Crossover

Un possibile risultato dell'applicazione con una probabilità pari a  $P_c = 0.5$

Generation 0				Mating pool created after reproduction		After crossover (generation 1)		
i	String $X_i$	Fitness $f(X_i)$	$\frac{f(X_i)}{\sum f(X_i)}$	Mating pool	$f(X_i)$	Crossover point	$X_i$	$f(X_i)$
1	011	3	.25	011	3	2	111	7
2	001	1	.08	110	6	2	010	2
3	110	6	.50	110	6	-	110	6
4	010	2	.17	010	2	-	010	2
Total		12			17			17
Worst		1			2			2
Average		3.00			4.25			4.25
Best		6			6			7