

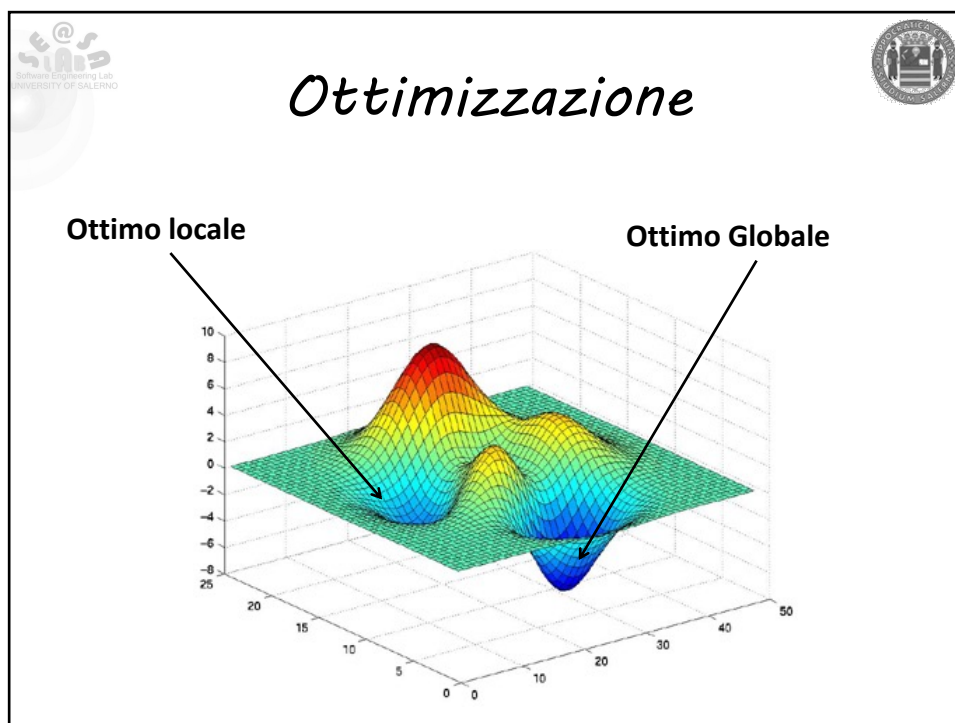


Search Based Software Engineering (SBSE)


...e algoritmi evolutivi




1



2



Background



Problemi di ottimizzazione con un solo obiettivo (funzione) da minimizzare (o massimizzare). Formalmente, abbiamo:

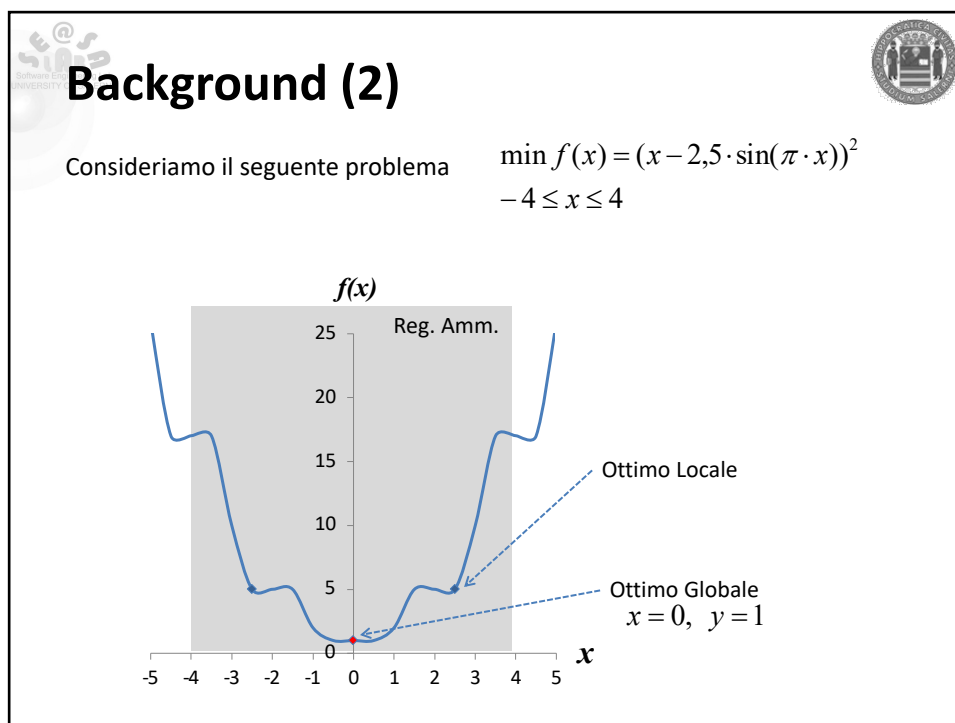
- Una funzione da ottimizzare $\min f : \mathbb{R}^n \rightarrow \mathbb{R}^1$
- Un vettore di variabili indipendenti $x = (x_1 \dots x_n)$
- Un sistema di vincoli da rispettare $\begin{cases} A \cdot x \leq b \\ x_{\min} \leq x \leq x_{\max} \end{cases}$ **Regione Ammissibile**

Soluzione

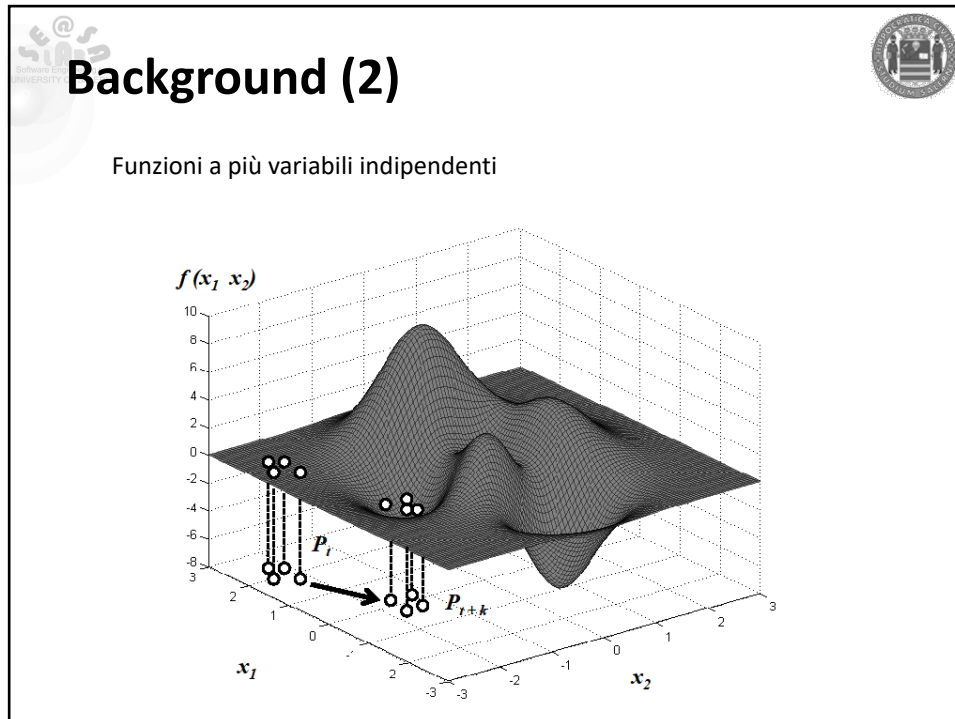
Diremo che x^* è un punto di minimo (globale) di f in x se e solo se

$$\forall y \in R.A.: f(x^*) \leq f(y)$$

3



4



5



Ottimizzazione e S.E.

Search Based Software Engineering (SBSE): *Software Engineering questions are often phrased in a language that simply cries out for an optimization-based solution.*

Tipici problemi di SBSE:

- Qual è il **più piccolo sottoinsieme** delle test suite che copre tutte le istruzioni del codice?
- Qual è la **migliore allocazione** delle risorse per un particolare progetto software?
- Qual è la migliore decomposizione del sistema che **massimizza la coesione** delle classi?
-

6

Ottimizzazione e S.E.



Search Based Software Engineering (SBSE): Software Engineering questions are often phrased in a language that is not directly amenable to an optimization-based solution

Tipici problemi

- Qual è la migliore istanza di un problema?
- Quali sono le migliori soluzioni software per un dato problema?
- Quali sono le migliori configurazioni del sistema che **massimizza la coesione** e **minimizza la complessità**?
-

Problemi sono NP-completi
Non c'è formulazione algebrica
Spazio di ricerca troppo vasto

7

Ottimizzazione e S.E.



Search Based Software Engineering (SBSE): Software Engineering questions are often phrased in a language that is not directly amenable to an optimization-based solution

Tipici problemi

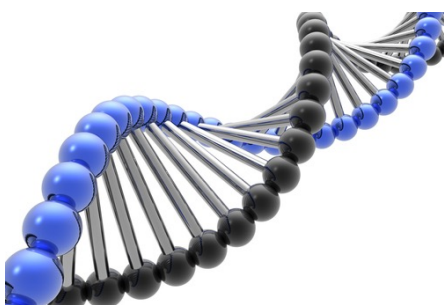
- 1) Scrivere una funzione di fitness (da ottimizzare)
- 2) Usare algoritmi di intelligenza artificiale per trovare l'ottimo.

- Quali sono le migliori soluzioni software per un dato problema?
- Quali sono le migliori configurazioni del sistema che **massimizza la coesione** e **minimizza la complessità**?
-

8



Algoritmi genetici (GA)



“Genetic Algorithms are good at taking large, potentially huge search spaces and navigating them, looking for optimal combinations of things, solutions you might not otherwise find in a lifetime.”

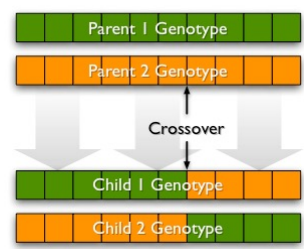
- Holland, *Adaptation in Natural and Artificial Systems* , May 1975

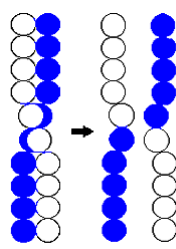
9



Background

Evoluzione: in natura gli organismi (animali o piante) **producono** un numero di figli (offspring) che sono abbastanza (non completamente) simili ai propri genitori (Estinzione e Adattamento)






10






Evoluzione

Il principio dell'evoluzione è “**survival of the fittest**”: The good offspring survive, while bad ones die”.



11

Concetti di GA

Dato un problema di ottimizzazione

$$\begin{aligned} &\text{minimize} \quad f(x) = f(x_1, x_2, \dots, x_n) \\ &\text{subject to} \quad x_i^{\min} \leq x_i \leq x_i^{\max} \quad i = 1, 2, \dots, n \end{aligned}$$

Ogni soluzione \mathcal{X} al problema, può essere visto come un individuo

$f(x)$

 $= f($



x_1, x_2, \dots, x_n

 $)$

Probabilità di
sopravvivenza

Cromosoma

12






Esempio

Funzione da ottimizzare

$$\min y = \sum_{i=1}^4 (x_i)^2$$

13

Esempio



Funzione da ottimizzare

$$\min y = \sum_{i=1}^4 (x_i)^2$$

1) Soluzioni (individui) casuali

x1	x2	x3	x4	Y(X)
1	-1	2	5	31
3	-2	2	8	81
-10	6	5	2	165
-2	2	2	0	12

14

Esempio

Funzione da ottimizzare

$$\min y = \sum_{i=1}^4 (x_i)^2$$



1) Soluzioni (individui) casuali 2) Selezione delle migliori soluzioni

x1	x2	x3	x4	Y(X)
1	-1	2	5	31
3	-2	2	8	81
-10	6	5	2	165
-2	2	2	0	12

1-125

-2220

15

Esempio

Funzione da ottimizzare

$$\min y = \sum_{i=1}^4 (x_i)^2$$

3) Ricombinazione



1-125

-2220

1-120

-2225

16

Esempio



Funzione da ottimizzare

$$\min y = \sum_{i=1}^4 (x_i)^2$$

3) Ricombinazione

4) Mutazione

17

Esempio

Funzione da ottimizzare

$$\min y = \sum_{i=1}^4 (x_i)^2$$



3) Ricombinazione

4) Mutazione

Nuova Generazione

x1	x2	x3	x4	Y(X)
1	-1	2	5	31
-2	2	2	-1	13
1	-1	2	0	6
-2	2	2	0	12

18

Esempio

Funzione da ottimizzare

min $y =$

3) R

1

-2

2

2

0

Processo Evolutivo
=
Algoritmo iterativo

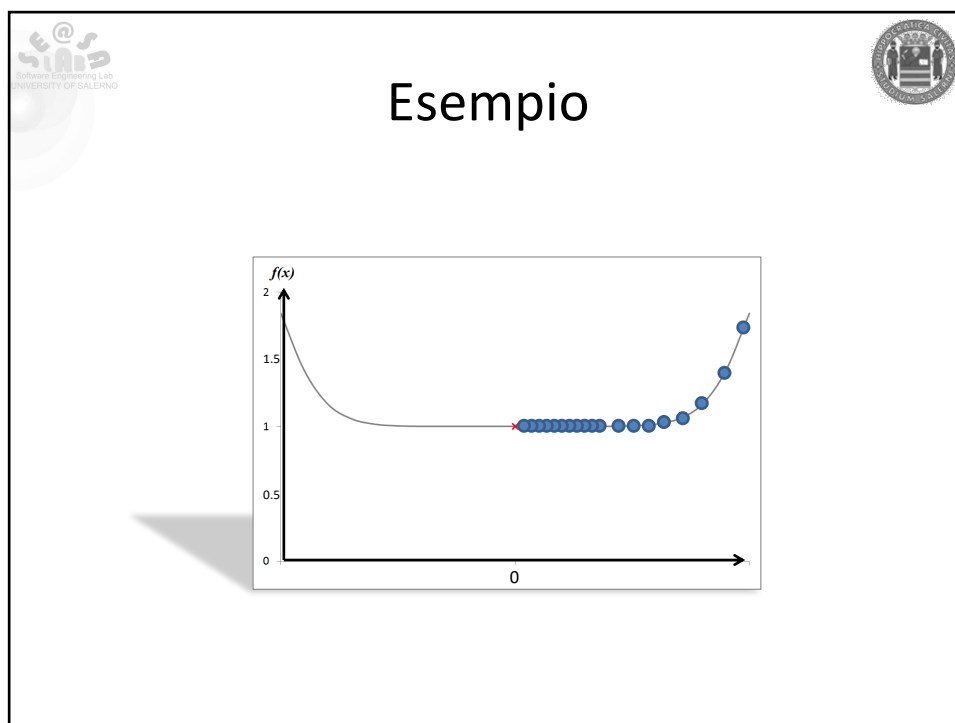
31

13

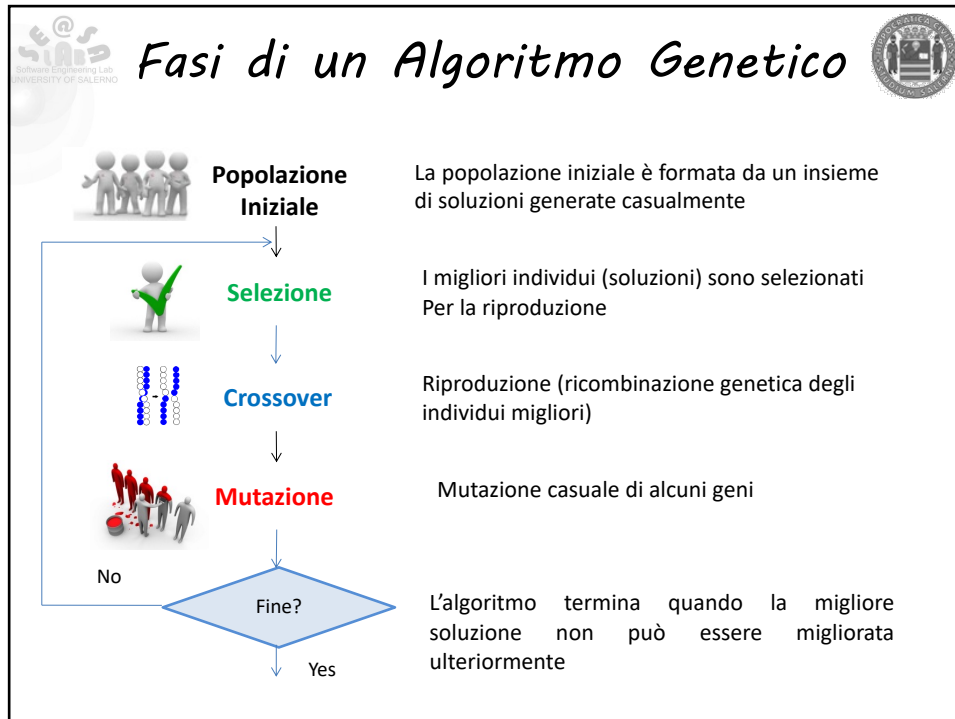
6

12

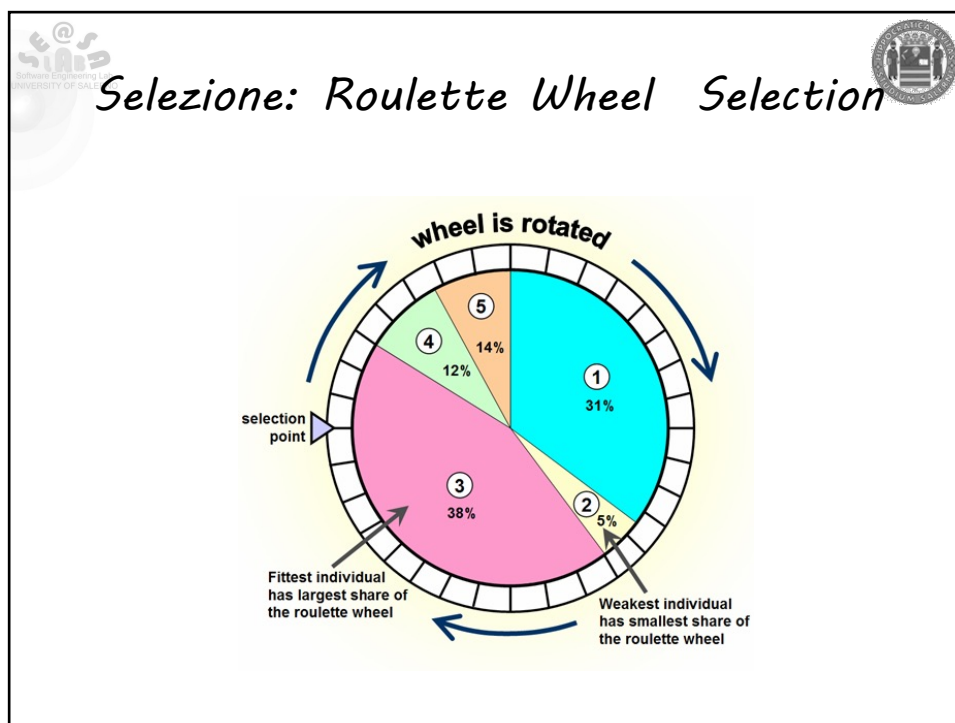
19



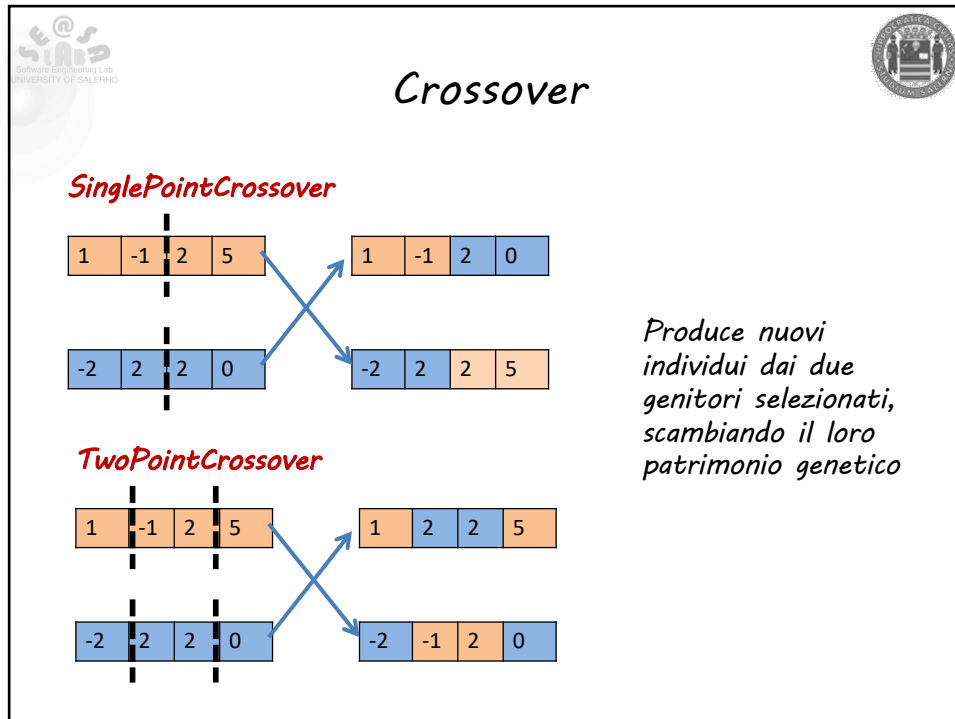
20



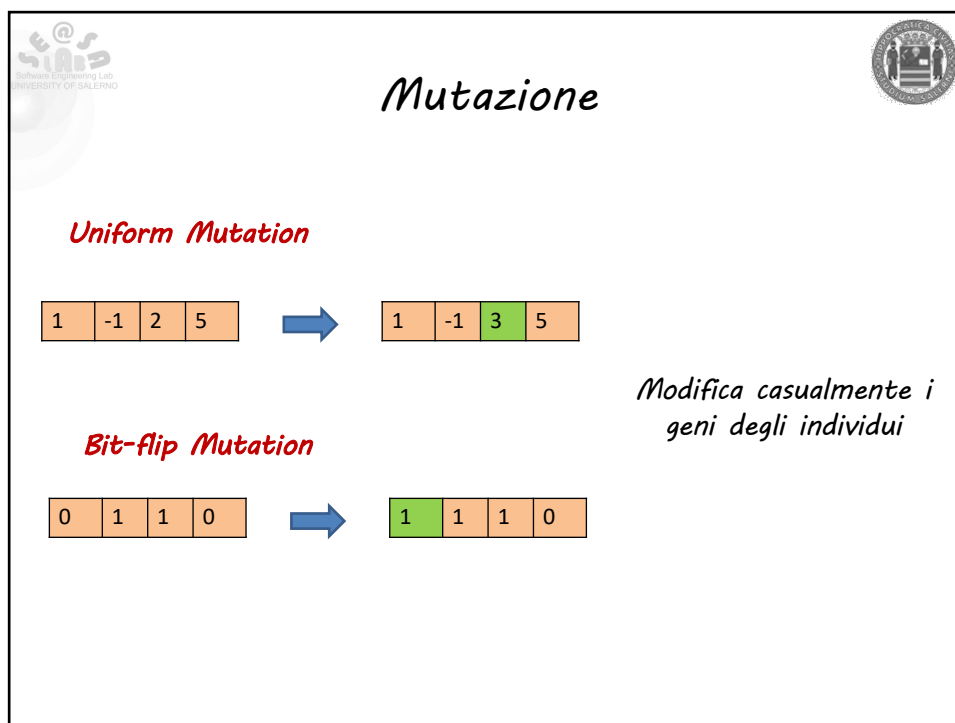
21





22



23



24

Problemi Multi-obiettivo (MOP)

Problemi nel quale bisogna ottimizzare più funzioni contemporaneamente.

$$\min \{f_1(x) \dots f_m(x)\}$$

Più funzioni da ottimizzare contemporaneamente



$$A \cdot x \geq b$$

$$x_{\min} \leq x \leq x_{\max}$$

Vincoli che determinano la regione ammissibile

Con $x = (x_1 \dots x_n)$

25

Sfide dei MOP

- In molti problemi della vita reale, le funzioni da ottimizzare sono in conflitto tra loro.
- Ottimizzando x rispetto ad una unica funzione si traduce spesso in risultati inaccettabili rispetto alle restanti funzioni.
- Una perfetta soluzione multi-obiettivo che ottimizza allo stesso tempo ogni funzione obiettivo è spesso impossibile.
- Problemi che risultano essere semplice nella versione mono-obiettivo, spesso diventano complessi (NP-hard) nella versione multiobiettivo.

26

MOP - Esempio

Test Case Selection:

Selezionare un sottoinsieme della test suite che (i) massimizzi la copertura del codice e (ii) minimizzi il costo complessivo di esecuzione.

Una soluzione è un vettore binario $X = \{x_1, x_2, \dots, x_n\}$

Cromosoma

$$x_i = \begin{cases} 1 & \text{se } t_i \text{ è selez.} \\ 0 & \text{altrimenti} \end{cases}$$

Funzioni da ottimizzare

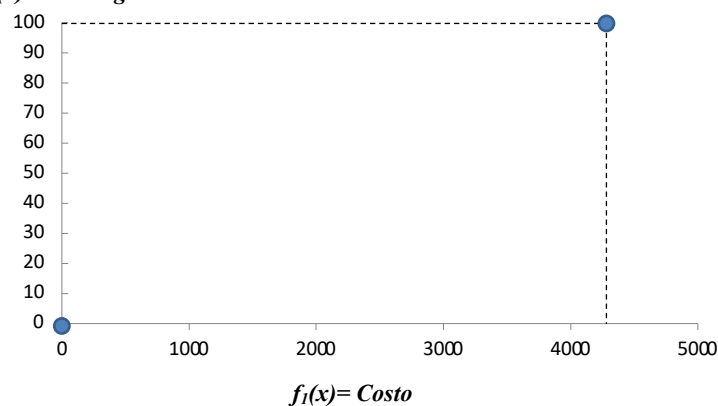
$$\begin{aligned} \min \text{ cost}(X) &= \sum_{i=1}^n (x_i \cdot c_i) \\ \max \text{ cov}(X) &= M \cdot X^T \end{aligned}$$

27

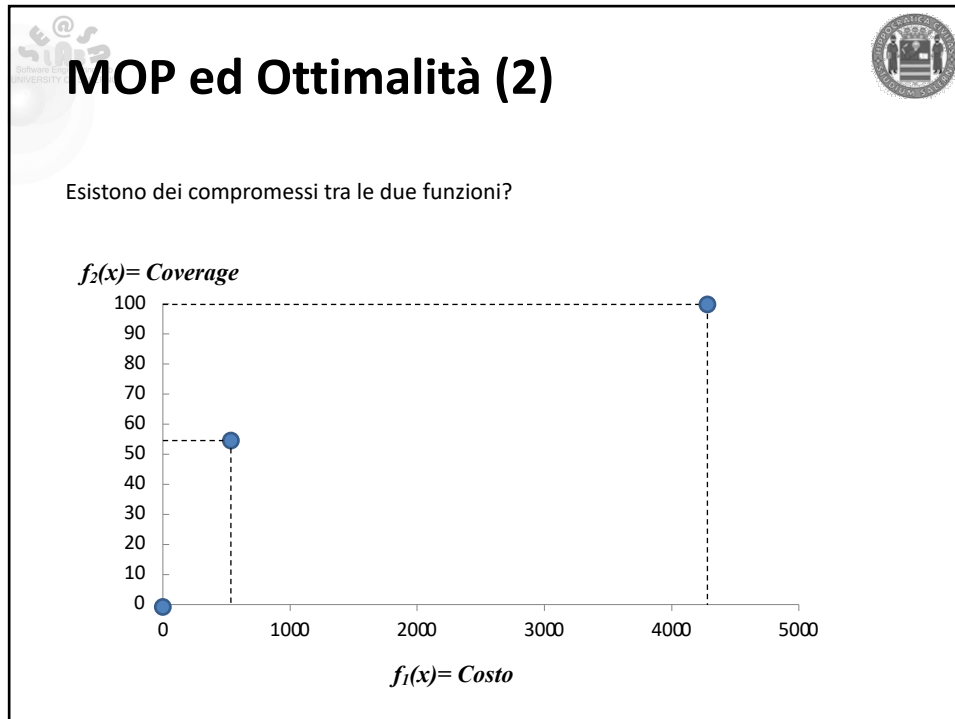
MOP ed Ottimalità (2)

Esistono dei compromessi tra le due funzioni?

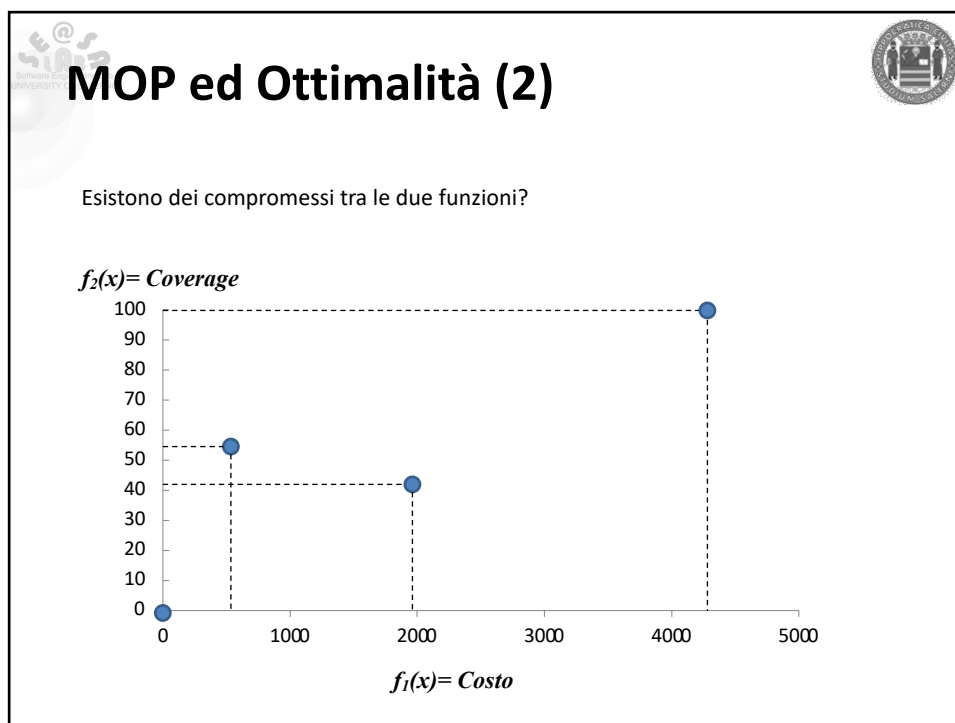
$f_2(x) = \text{Coverage}$



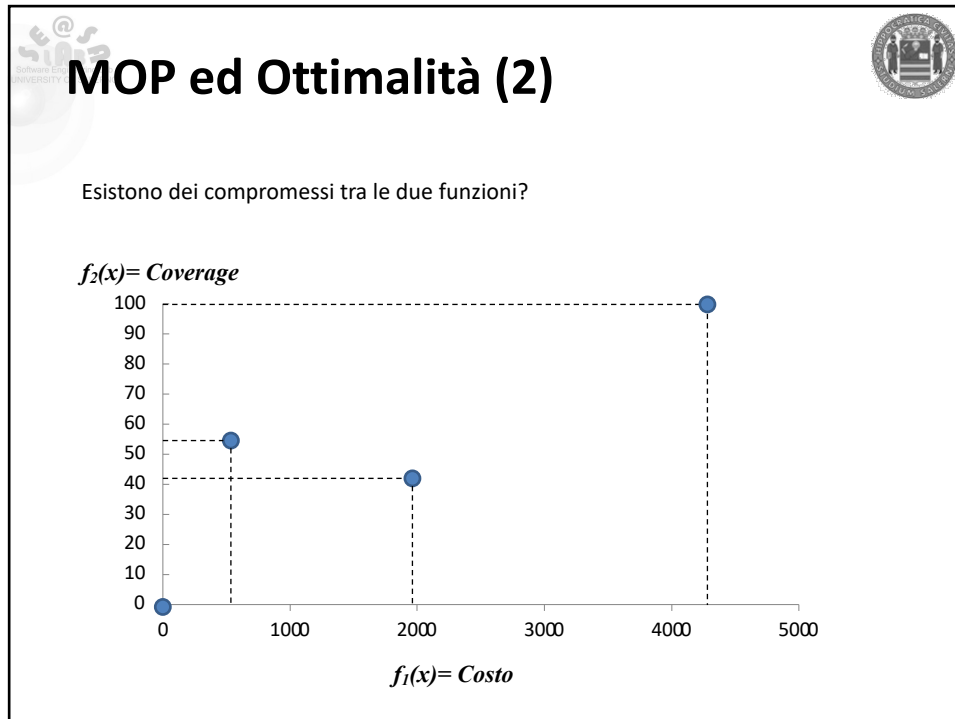
28



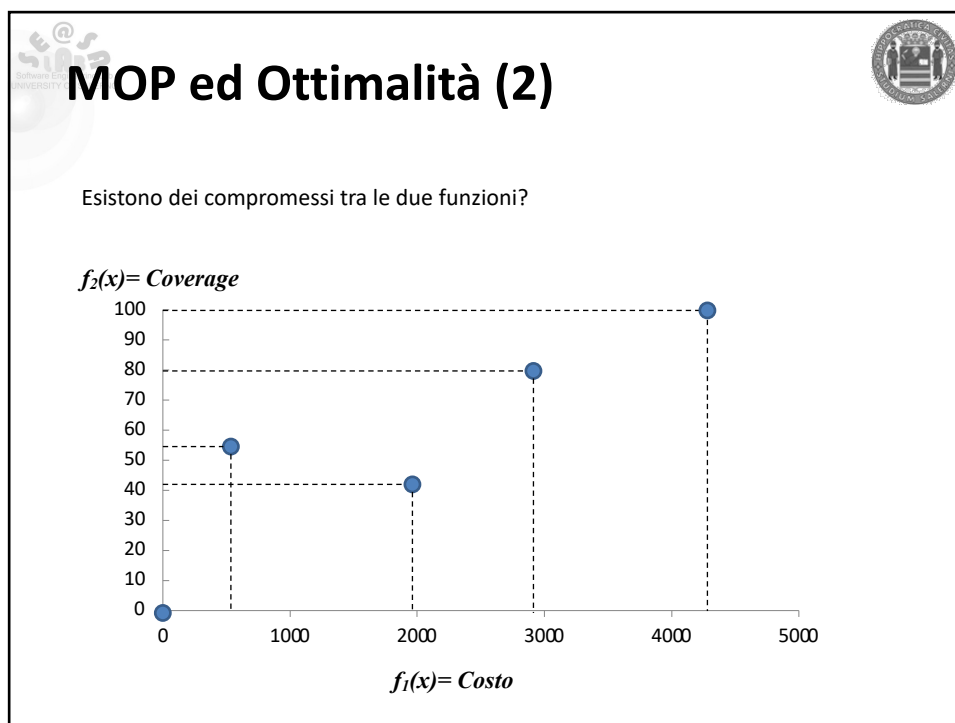
29



30



31



32

Ottimo di Pareto

E' necessario definire con chiarezza cosa si intende per soluzione ottima di un problema di ottimizzazione multiobiettivo.

Definizione 1. Dati due vettori di soluzioni x e $y \in R^n$, diremo che x domina y secondo Pareto ($x \leq_p y$) se e solo se

- 1) $f_i(x) \leq f_i(y)$ per ogni $i=1,2,\dots,m$
- 2) $f_j(x) < f_j(y)$ per almeno un indice $j \in \{1, \dots, m\}$

Definizione 2. Un vettore di soluzioni $x^* \in \text{Reg. Amm.}$ è un ottimo di Pareto per le funzioni $\{f_1, \dots, f_m\}$ se non esiste un altro vettore $x \in \text{Reg. Amm.}$ tale che

$$f_i(x) \leq_p f_i(x^*)$$

33

GA vs. Multi-Objective GA

GA

Problema

$$\begin{aligned} \min \quad & f(x) \\ A \cdot x & \geq b \\ x_{\min} & \leq x \leq x_{\max} \end{aligned}$$

$$f(x) = f(x_1, x_2, \dots, x_n)$$

Probabilità
Di sopravvivere

Cromosoma

Multi-objective GA

Problema


$$\begin{aligned} \min \quad & \{f_1(x) \quad \dots \quad f_m(x)\} \\ A \cdot x & \geq b \\ x_{\min} & \leq x \leq x_{\max} \end{aligned}$$

$$\begin{aligned} f_1(x) &= f_1(x_1, x_2, \dots, x_n) \\ \vdots &= \vdots \\ f_m(x) &= f_m(x_1, x_2, \dots, x_n) \end{aligned}$$


Probabilità
Di sopravvivere

Cromosoma

34




Multi-Objective GA




- **Quali sono gli individui migliori?**
 - Per i problemi mono-obiettivo gli individui “migliori” sono quelli che hanno valore minimo (o massimo) della funzione obiettivo.
 - Per i problemi multi-obiettivo gli individui “migliori” devono ottimizzare più funzioni obiettivo contemporaneamente .
- **Come calcoliamo la probabilità di sopravvivenza degli individui?**
 - Per i problemi mono-obiettivo gli individui “migliori” hanno maggiore probabilità di sopravvivere.
 - Determinare la probabilità di sopravvivenza degli individui in problemi multi-obiettivo è tutt'altro che scontata.

35



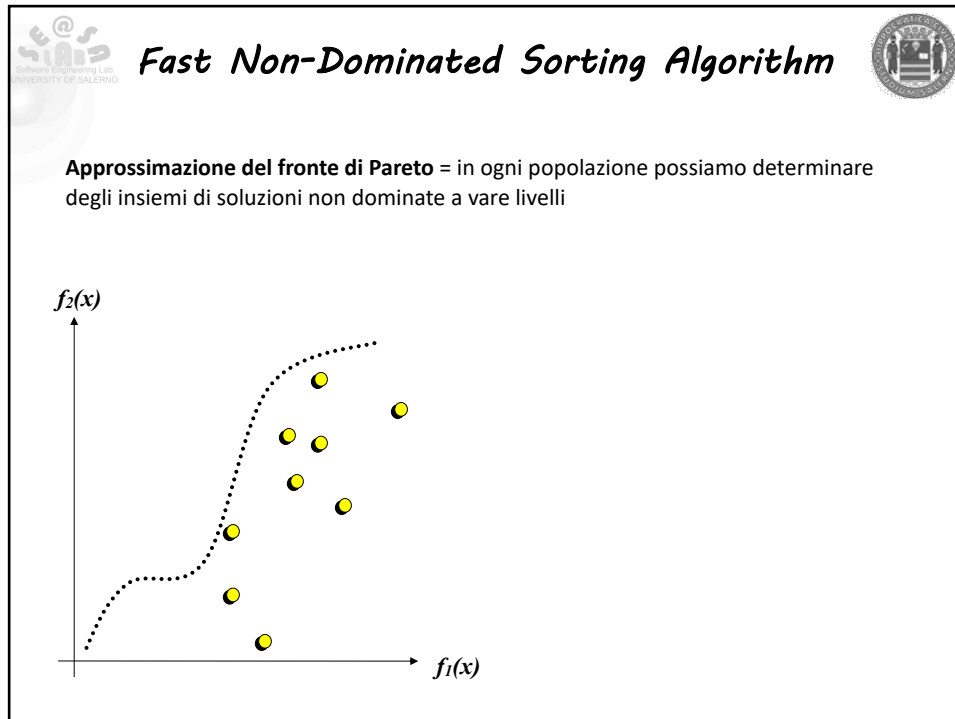
Multi-Objective GA



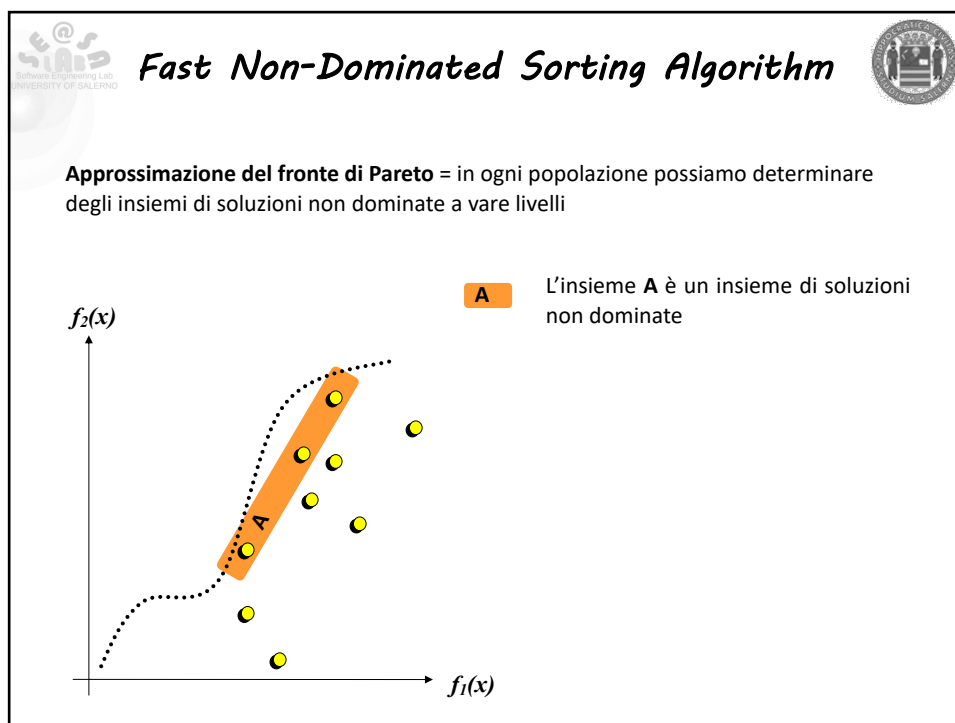
- **Quali sono gli individui migliori?**
 - Per i problemi mono-obiettivo gli individui “migliori” sono quelli che hanno valore minimo (o massimo) della funzione obiettivo.
 - Per i problemi multi-obiettivo gli individui “migliori” devono ottimizzare più funzioni obiettivo contemporaneamente .
- **Come calcoliamo la probabilità di sopravvivenza degli individui?**
 - Per i problemi mono-obiettivo gli individui “migliori” hanno maggiore probabilità di sopravvivere.
 - Determinare la probabilità di sopravvivenza degli individui in problemi multi-obiettivo è tutt'altro che scontata.

Non-Dominated Sorting Algorithm

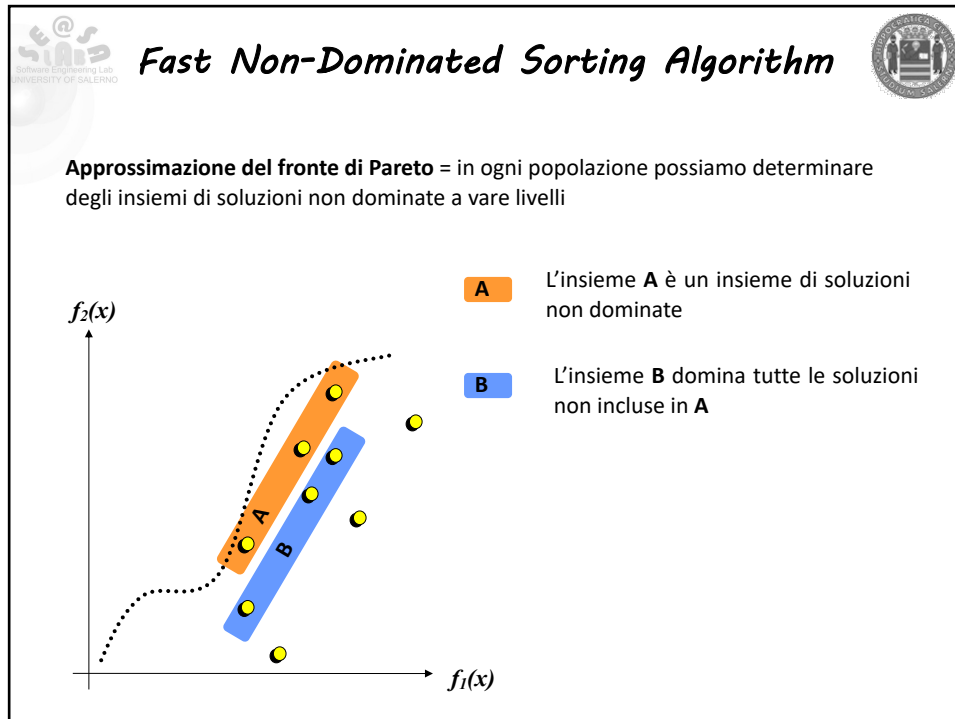
36



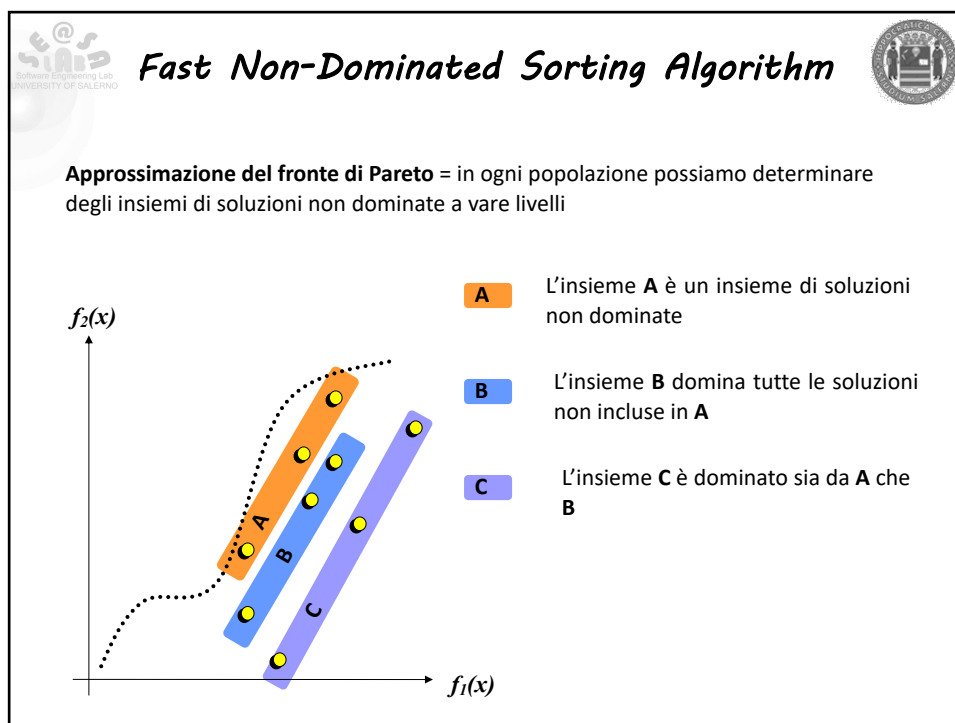
37



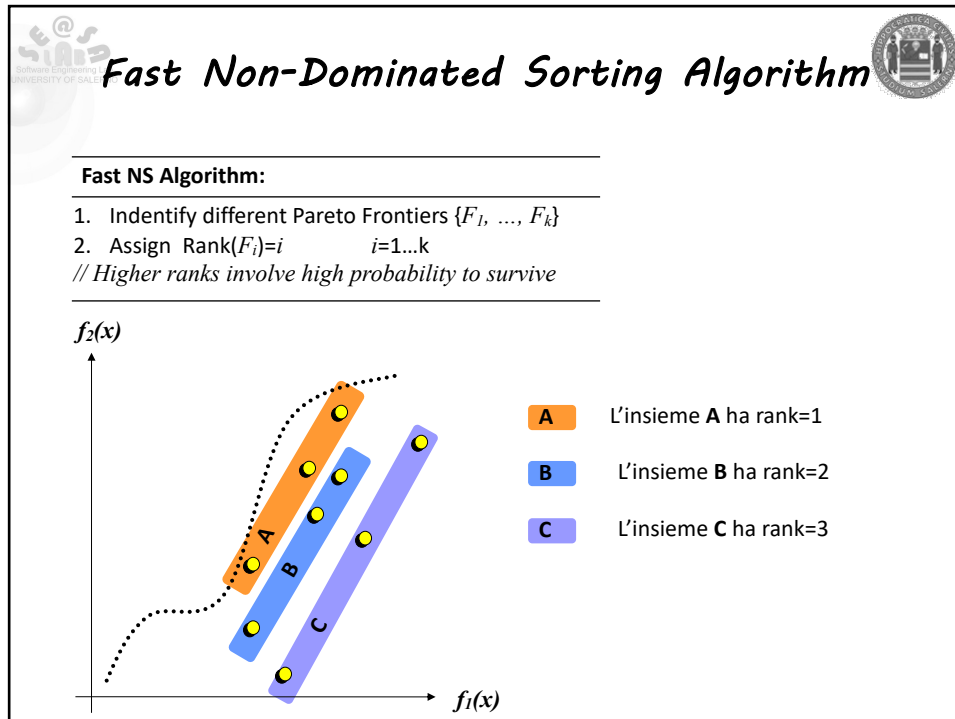
38



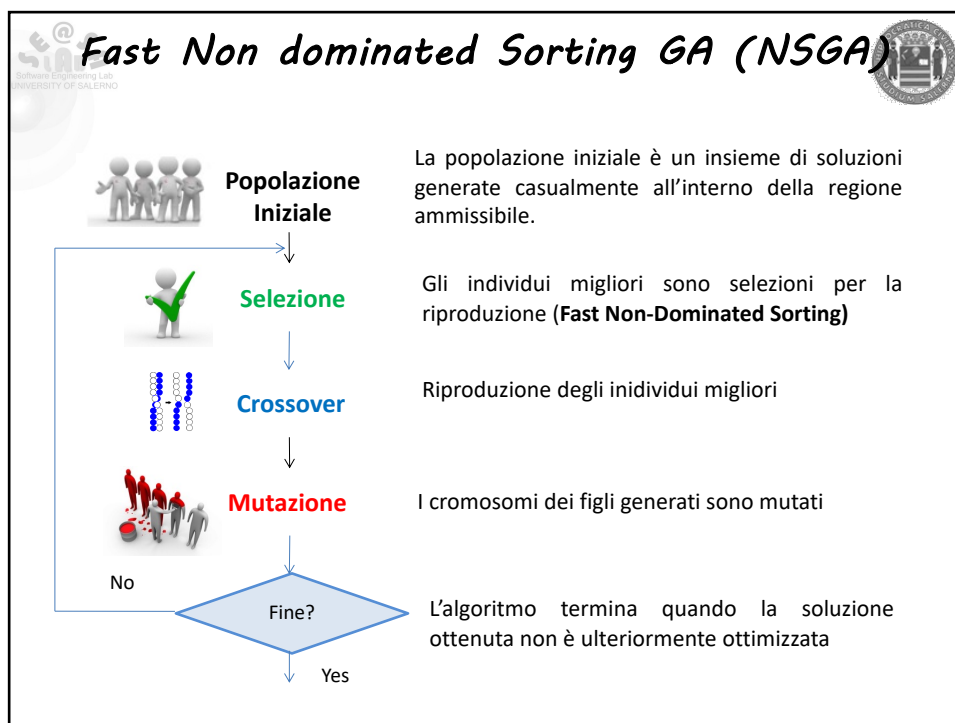
39



40



41



42



GA per l'Ottimizzazione del Testing di Regressione



Rappresentazione - Test Suite Minimization and Test Case Selection

1	0	0	0	1	0	1	1	1
---	---	---	---	---	---	---	---	---

1	1	0	1	1	0	0	1	0
---	---	---	---	---	---	---	---	---

Rappresentazione - Test Case Prioritization

3	5	1	4	7	2	8	6	9
---	---	---	---	---	---	---	---	---

8	5	1	9	6	2	3	7	4
---	---	---	---	---	---	---	---	---