UNIVERSITA' DEGLI STUDI DI
NAPOLI FEDERICO II
Scuola Politecnica e delle Scienze di Base
Corso di Laurea Magistrale in Ingegneria Informatica

tesi di laurea magistrale

Un algoritmo genetico per il testing di applicazioni web basate su Javascript

relatore
Ch.mo Prof Porfirio Tramontana

candidato
Michele Palumbo
Matr. M63/215

MOTIVAZIONI

- *Applicazioni web moderne* sempre più interattive → complesse
- *Javascript* è il linguaggio che garantisce l'interattività
- La dinamicità complica l'attività di testing
- *Testing manuale* diventa troppo costoso

OBIETTIVO

- Realizzazione di una tecnica di testing completamente automatica che, a partire da una test-suite ottenuta con l'ausilio di un tool esterno, sia in grado di modificarne i singoli test case in modo da massimizzare l'efficienza della test-suite finale.

TECNICHE DI TESTING

- ***SISTEMATICO*:** esplorano l'applicazione sulla base di un modello.
- ***RANDOM*:** effettuano un'esplorazione casuale dell'applicazione.
- ***TECNICHE SEARCH-BASED***
  - **Ottimizzazioni classiche**

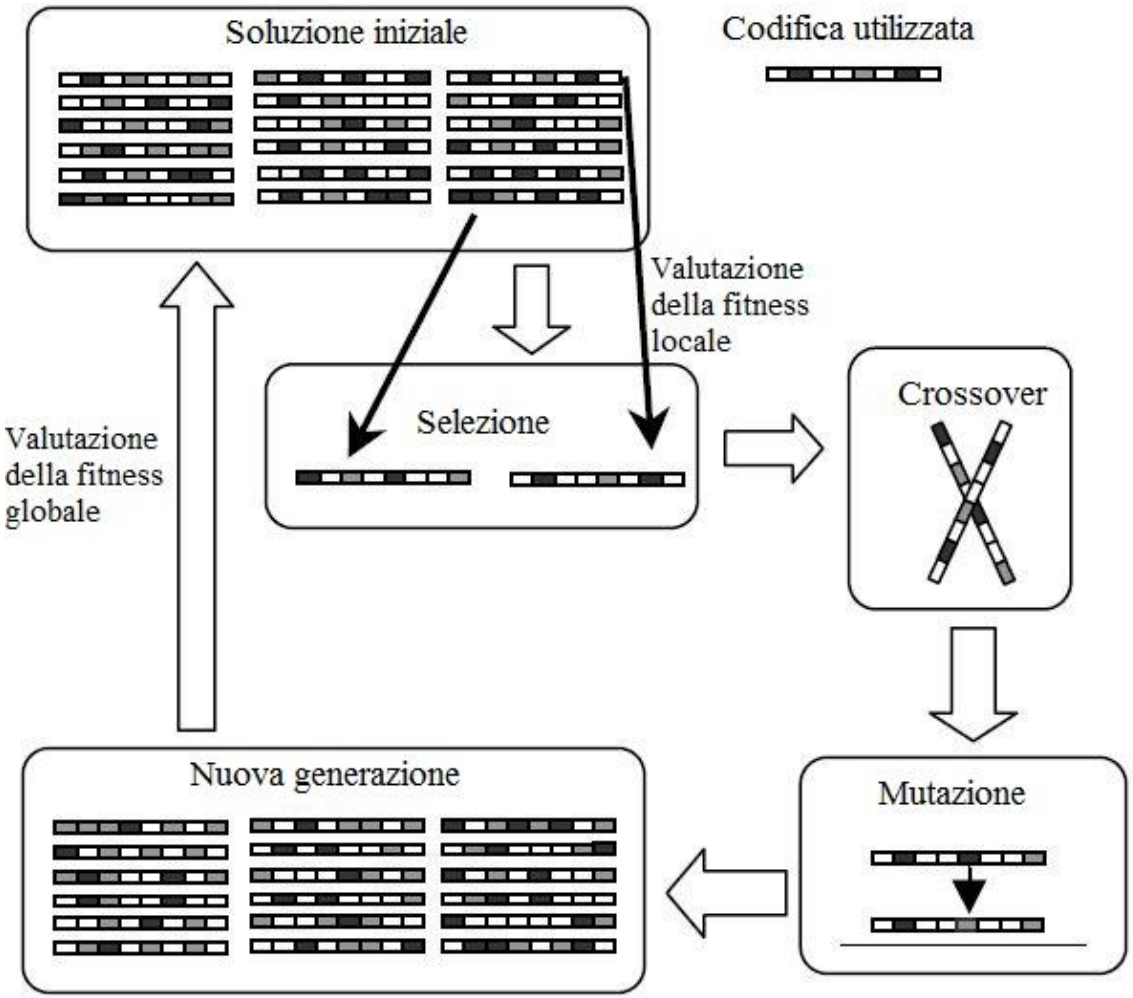
    Programmazione lineare
  - **Ricerca metaeuristiche:**

    Hill Climbing

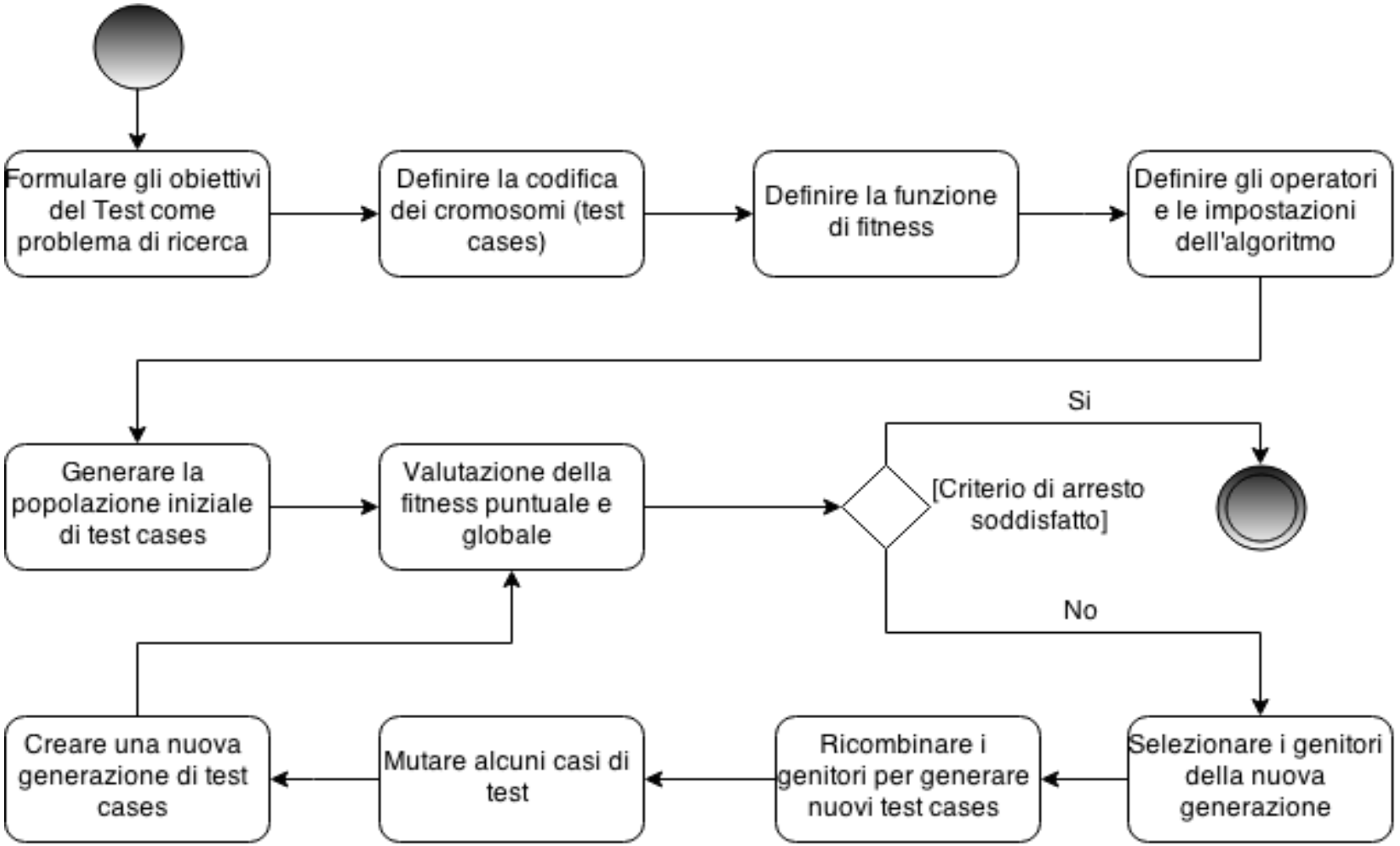
    Simulated Annealing

    Algoritmi genetici

UN GENERICO PROCESSO GENETICO



TESTING AUTOMATIZZATO TRAMITE ALGORITMI GENETICI



L'ALGORITMO GENETICO PROPOSTO – LE COMPONENTI PRINCIPALI (1/2)

Soluzione inziale

- **Grafo degli stati,** lista eventi ed input generati da Crawljax
- Prima soluzione ammissibile: **test-suite inizale**

Codifica utilizzata

- Definizione di **gene:** $G = \{S, E, I, F\}$ dove:

  Definizione di **cromosoma:** $C = \{G_1, \dots, G_n\}$

S : *stato iniziale*
E : evento scatenabile su S
I : valori di input associati ad E
F : stato finale

Funzioni di fitness

- Metriche puntuali: *copertura locale*, *diversità* e *rank*
- Metrica globale: efficacia in termini di LOC coperte

$Rank = \sum_{1 \le j \le n} l_j w_j$ dove:

- *lj* che vale 1 se la i-esima linea è coperta, 0 viceversa
- $0 \le w_j \le 1/2 \ \ con \ \ w_j = 1/\sum_{k=1}^{T} \boldsymbol{m}_k$ con T dimensione della test-suite e $\boldsymbol{m}_k$ che vale 1 se la i-esima linea è coperta dal k-esimo test di T e 0 viceversa.

Tecnica di selezione

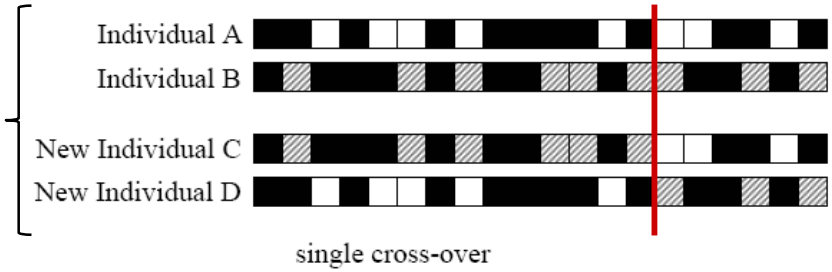
- Selezione dei cromosomi da sostituire
- Selezione test da incrociare: *Steady-State reproduction*
- Selezione cromosomi da mutare

Matrice delle diversità «D»
tale che:

- *nella posizione D[i][j] è memorizzato il numero di linee coperte dal test i-esimo ma non dal test j-esimo*.

L'ALGORITMO GENETICO PROPOSTO – LE COMPONENTI PRINCIPALI (2/2)

- **Tecnica di Crossover**
  - Condizioni di compatibilità tra cromosomi
  - Tipologia: *Single-Point Cut*



- **Tecnica di Mutazione**
  - Scelta dei nuovi valori
  - Tipologia: *random* o *sistematica*

- **Impulso di turnover - Cataclysmic Mutation**
  - Soluzione al problema del massimo locale: quando la *fitness globale* rimane stabile per un determinato numero di iterazioni, si sostituiscono tutti i test la cui copertura è in almeno un altro test con nuovi «individui» generati tramite operatori di crossover e mutazioni

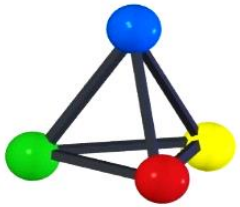
- **Condizioni di terminazione**
  - Numero di iterazioni massime raggiunto
  - *Coverage* desiderata raggiunta

STRUMENTI UTILIZZATI

Selenium

- Framework open source per testare automaticamente le applicazioni web
- Componenti principali: *IDE, Cliente Api, Remote Control, Web Driver, Grid*

Crawljax



- Software open source per l'esplorazione automatica di applicazioni web di tipo AJAX basate su JavaScript
- Motore *event-driven*
- Generazione automatica del *grafo degli stati* relativo ai DOM dinamici
- Architettura basata su Plugin

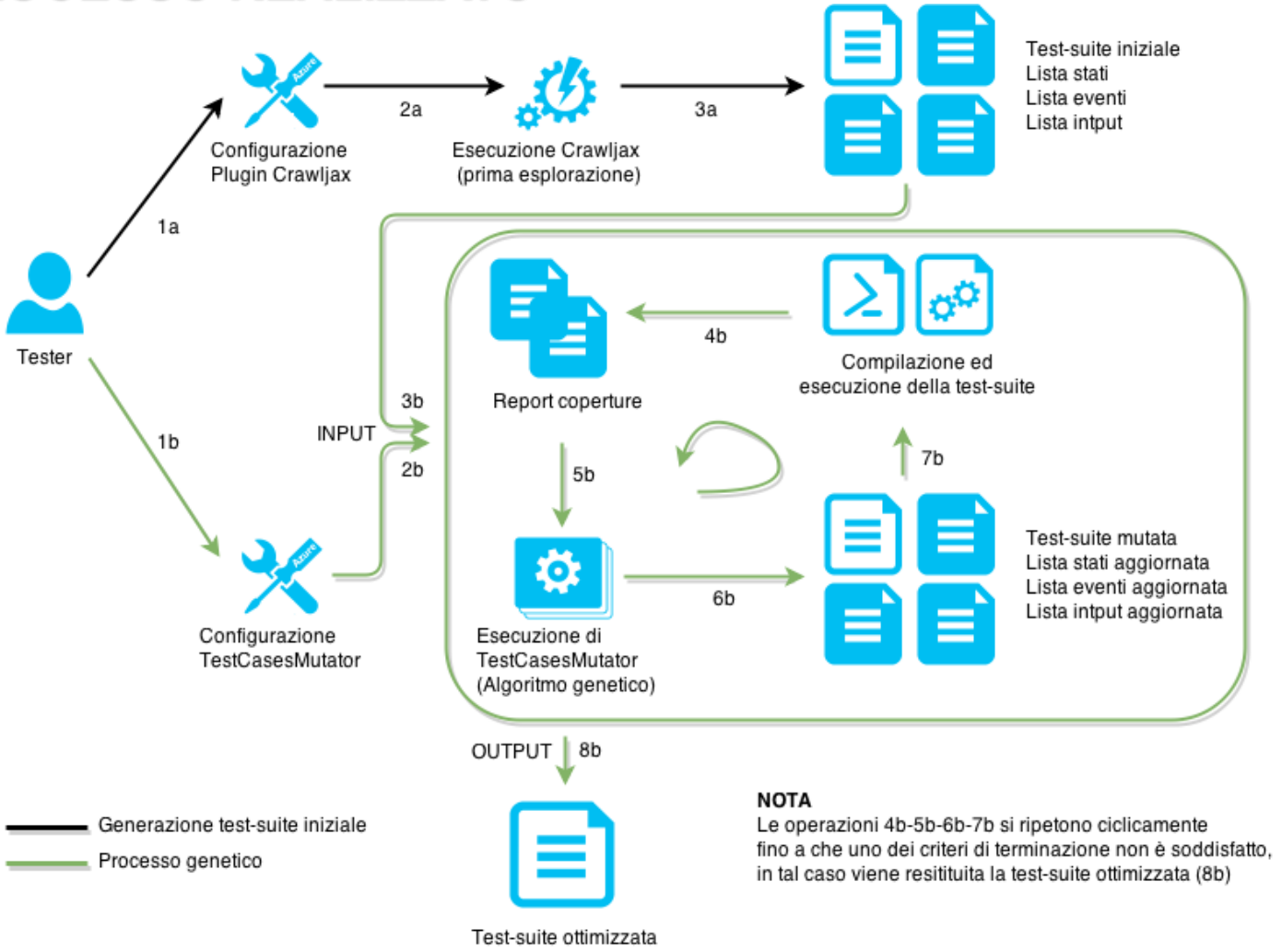
Junit

- Framework open source per scrittura ed esecuzione di casi di test in JAVA
- Misurazione progressi, rilevazione difetti, esecuzione automatica, meccanismo delle annotazioni (***@Before, @After, @BeforeClass etc..***)

JSCover

- Software open source per la misura della copertura di codice Javascript
- Instrumentazione del codice JS in diverse modalità (*Server, FS, Proxy*)
- Generazione dei files di report in differenti formati (*LCOV, XML Summary*…)

PROCESSO REALIZZATO



VARIABILI DELLO STRUMENTO REALIZZATO

- **Variabili indipententi**
  - Percentuale crossover, percentuale mutazioni, percentuale turnover
  - Numero iterazioni massime
  - Frequenza di impulso
  - Dimensione test-suite
  - Intervallo di tempo tra la generazione di eventi consecutivi

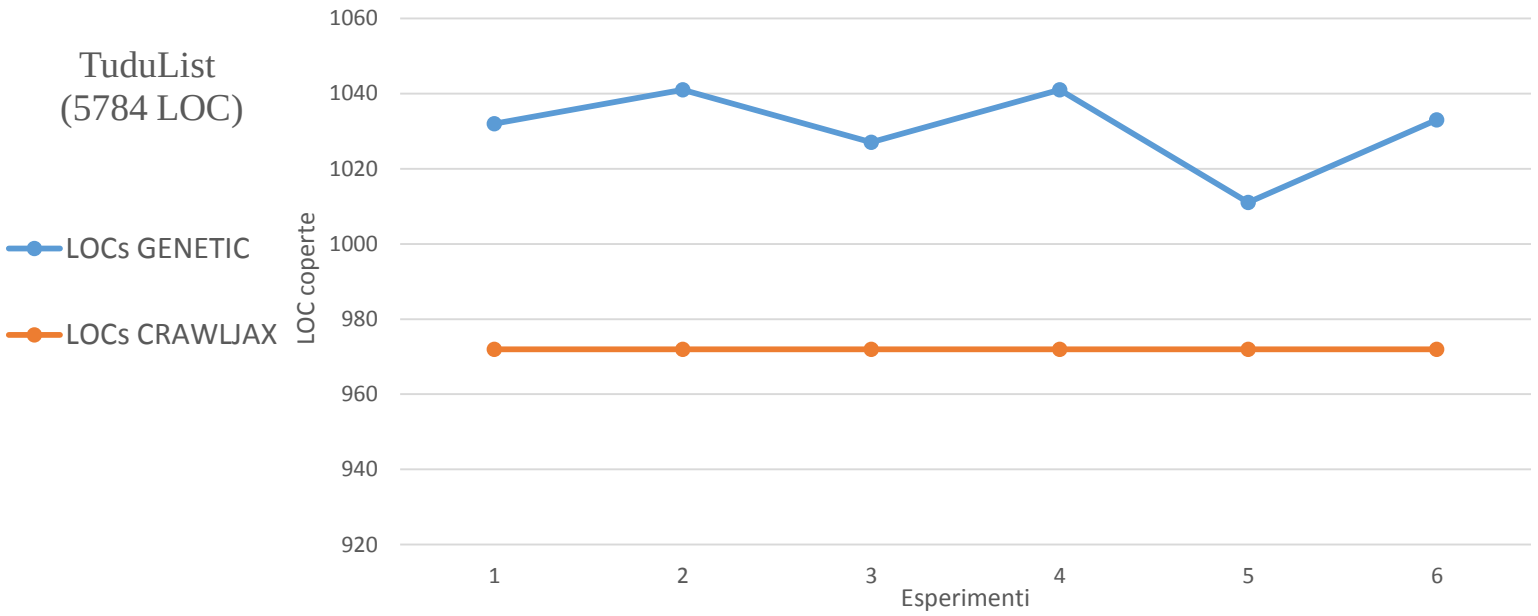
- **Variabili dipendenti**
  - Efficacia: $\eta(G) = \frac{LOC\ coperte\ da\ G}{Numero\ totale\ di\ LOC} * 100$

- **Variabili controllate**
  - Numero di terazioni

- **Variabili randomizzate**
  - Selezione pseudo-casuale dei test da incrociare
  - Selezione pseudo-casual dei test da mutare e della mutazione

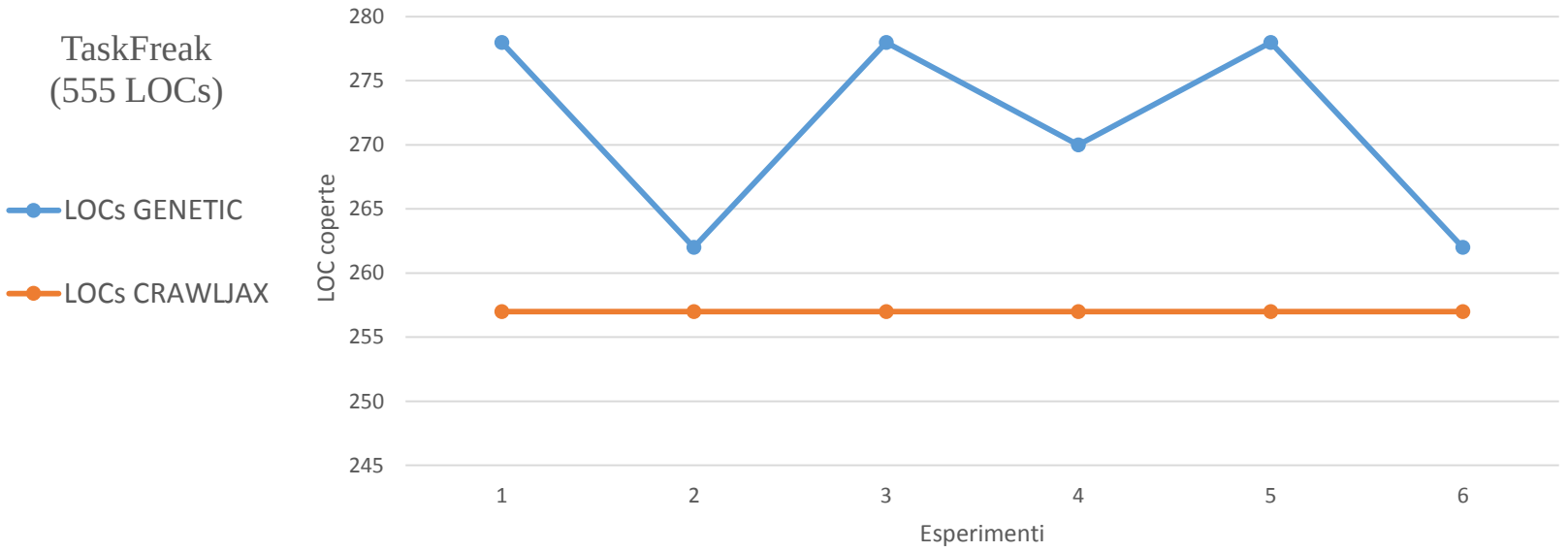
APPLICAZIONI TESTATE E RISULTATI OTTENUTI (1/2)

WebApp	LOCs Totali	LOCs Crawljax	LOCs Genetic Esperimento migliore	Iterazioni	Numero Massimo Impulsi	Dimensione Test Suite
Tudulist	5784	972 (15% totali) (84% manual)	1027 (18%totali) (90%manual)	30	5	50



APPLICAZIONI TESTATE E RISULTATI OTTENUTI (2/2)

WebApp	LOCs Totali	LOCs Crawljax	LOCs Genetic Esperimento migliore	Iterazioni	Numero Massimo Impulsi	Dimensione Test Suite
Taskfreak	555	258 (46% totali) (88% manual)	279 (48%totali) (96%manual)	30	4	50



CONCLUSIONI

- I risultati ottenuti dimostrano che lo strumento realizzato, agendo puntualmente sui singoli casi di test, ha effettivamente permesso di migliorare la test-suite iniziale, portando un **aumento di efficacia** fino all'8%.
- Inoltre anche l'**efficienza** ha subito un miglioramento dato che la dimensione della test-suite è rimasta costante ed i test ridondanti o interamente inclusi in altri sono stati individuati e sostituiti.

SVILUPPI FUTURI

- Avvio di una nuova esplorazione alla scoperta di un nuovo stato individuato dall'algoritmo genetico tramite l'utilizzo di Crawljax (tecnica *Hill Climbing*)
- Migliorare i tempi di esecuzione dello strumento tramite l'esecuzione su macchine che lavorano in parallelo
- Ricerca della configurazione ottima dei parametri di input per l'algoritmo genetico proposto