

Tesi di Laurea

# **Tecniche per la ricerca di test suite minimali per RIA**

Anno Accademico 2008/2009

## **Relatore**

Ch.mo prof. Porfirio Tramontana

## **Correlatore**

Ch.mo Ing. Domenico Amalfitano

## **Candidato**

Angelo Di Maria

Matr. 885/257



## **Obiettivo:**

- Realizzazione di nuove metodologie per la ricerca di Test Suite minimali a supporto del processo di testing delle Rich Internet Application (RIA) basate su AJAX.

## **Problematiche affrontate:**

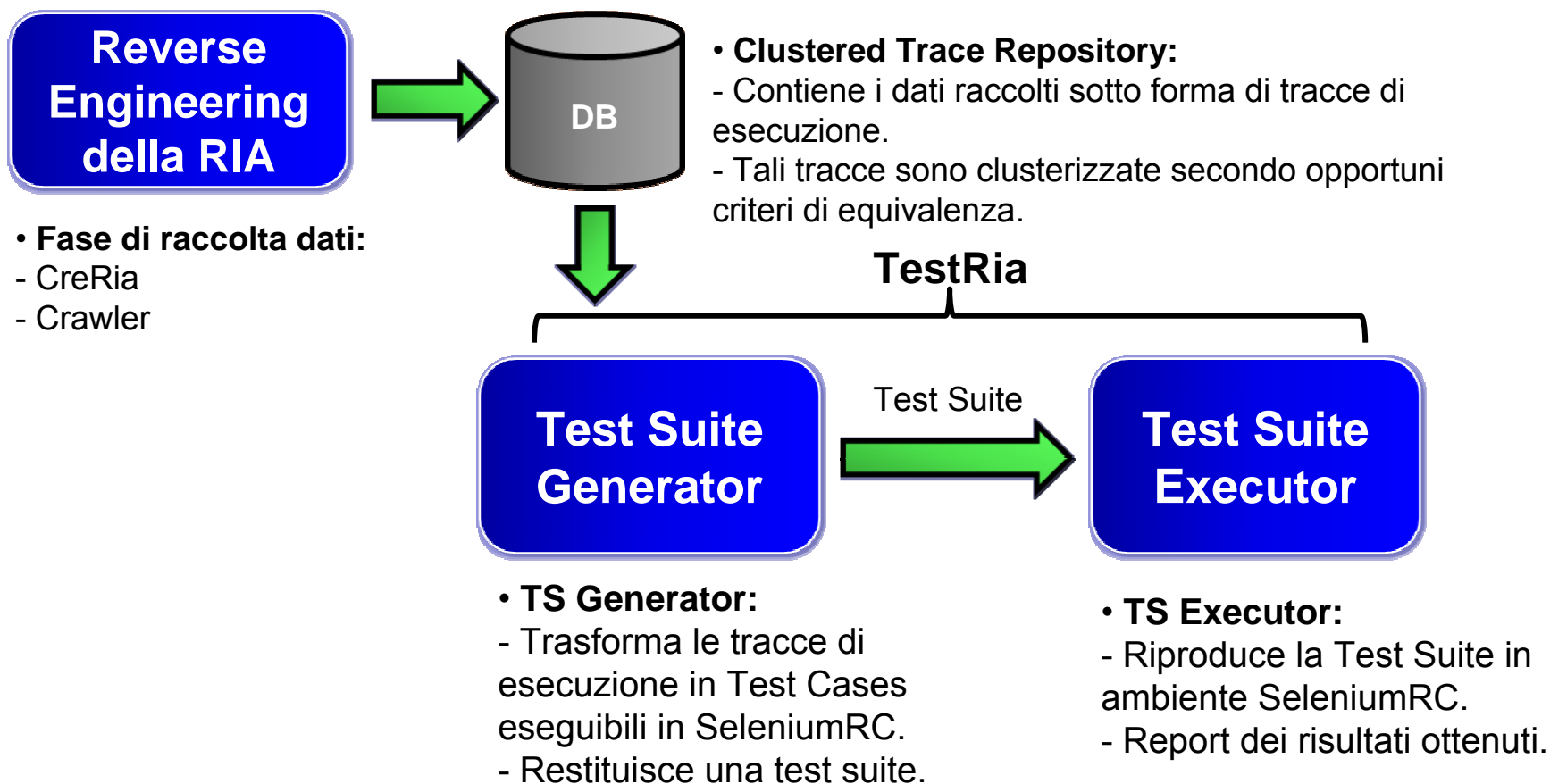
- Il processo di Reverse Engineering di una RIA produce Test Suite di dimensioni elevate rendendo dunque il tempo di esecuzione una variabile critica .

## **Soluzione proposta:**

- Sviluppo di nuove tecniche di minimizzazione delle Test Suite registrate in fase di RE di una RIA; tali tecniche:

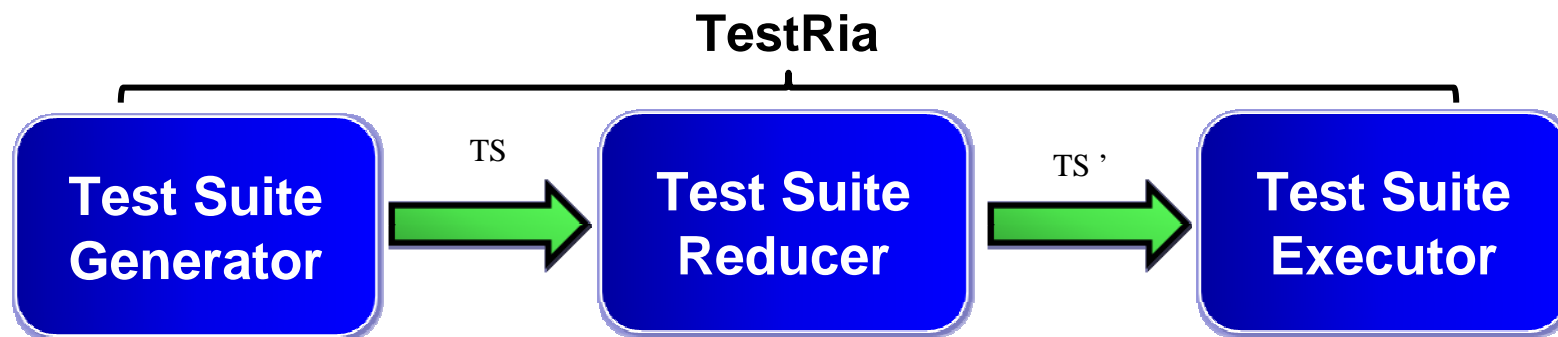
- usano una matrice di copertura MdC  $m \times n$
- usano l'algoritmo di riduzione di una matrice basato sul metodo delle righe dominate e colonne dominanti
- assicurano la stessa copertura delle Test Suite complete

## Processo di testing proposto per una RIA



## Il contributo apportato

- Il contributo apportato consiste nell'estendere il tool TestRia, di supporto alla fase di testing, implementando un ulteriore blocco funzionale chiamato **TestSuite Reducer**.

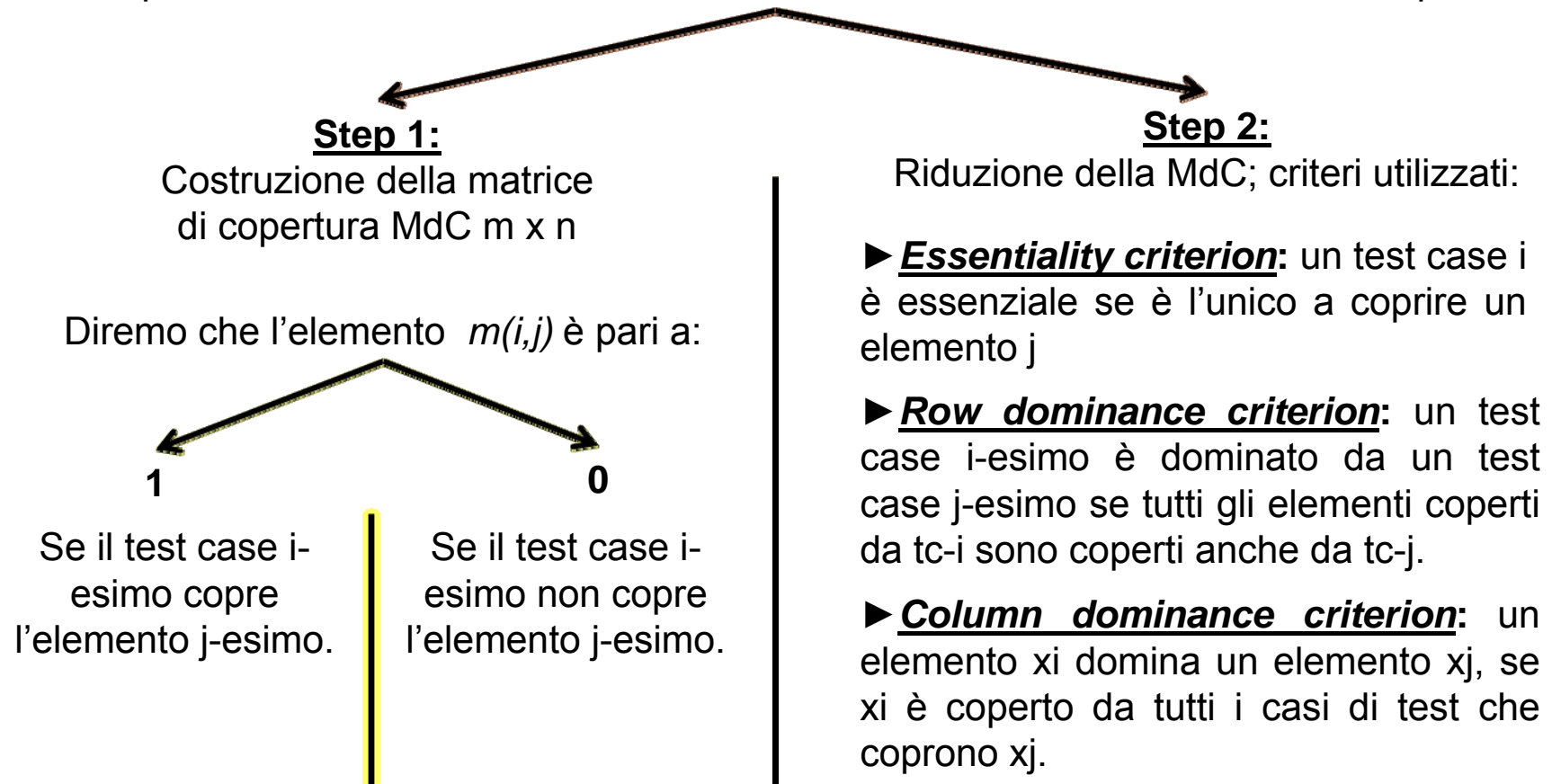


• **TS Reducer:**

A partire da una Test Suite TS data in ingresso restituisce una ulteriore Test Suite TS' in uscita di dimensione inferiore o al più uguale a quella di partenza e con la stessa percentuale di copertura.

## Test Suite Reducer (1/3)

Il processo di riduzione di una Test Suite è essenzialmente suddiviso in 2 step:



## Test Suite Reducer (2/3) : Step 1

Il processo di costruzione della matrice di copertura MdC  $m \times n$  prevede:

- di porre sulle righe i Test Case che compongono la test suite completa
- di porre sulle colonne gli elementi da coprire:
  - Stati
  - Transizioni
  - JavaScript eseguite
  - ...

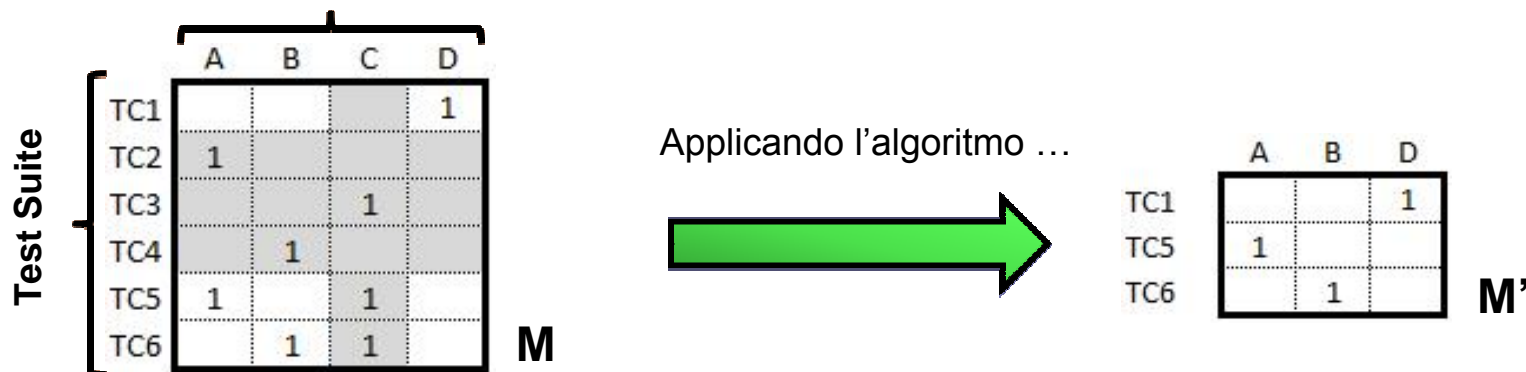
- 
- La cardinalità delle informazioni poste in colonna dipende sia dalla dimensione della Test Suite in esame che dai criteri di clusterizzazione usati.
  - Le informazioni poste in colonna rappresentano seppur in maniera parziale la nostra conoscenza sulla RIA.

	E1	E2	E3	...	EN
TC1				...	1
TC2	1			...	
TC3			1	...	
TC4		1		...	
...	...	...	...	...	...
TCM		1	1	...	

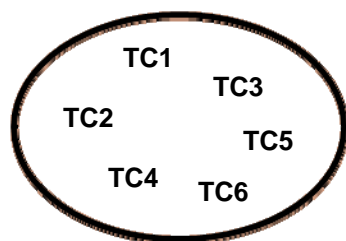
## Test Suite Reducer (3/3) : Step 2

**Teorema:** Eliminando in maniera iterativa le righe dominate e le colonne dominanti da una matrice  $M$ , si ottiene una matrice  $M'$  equivalente che rappresenta il medesimo problema di copertura.

Stati, Transizioni o Javascript

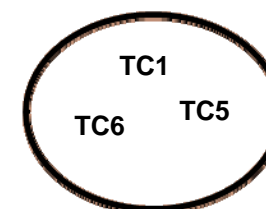


Test Suite Completa



Test Suite Reducer

Test Suite Ridotta



## Criteri di equivalenza utilizzati

- **Criteri di equivalenza tra interfacce:** due interfacce I1 e I2 sono equivalenti se i widget di I1 sono inclusi in I2, e i widget di I2 sono inclusi in I1. Caratteristiche dei widget considerate:

- **C1:** - Tipo di event listener
  - Nome degli event handler
  - IndexedPath
- **C2:** - Visibilità e abilitazione;
  - Tipo di event listener
  - Nome degli event handler
  - IndexedPath
- **C3:** - Visibilità e abilitazione;
  - UnindexedPath
- **C4:** - Visibilità e abilitazione;:
  - Pathid

- **Criteri di equivalenza per le transizioni:** due transizioni sono considerate equivalenti se per il criterio:

- **T0** si ha:
  - Stesso tipo di evento che ha causato la transizione
  - Stesso Pathid dell'oggetto DOM su cui è stato scatenato l'evento
- **T1** si ha :
  - Stessa interfaccia di partenza
  - Stessa interfaccia di arrivo
  - Tipo di evento che ha causato la transizione
  - Pathid dell'oggetto DOM su cui è stato scatenato l'evento
- **T2** si ha:
  - Stessa interfaccia di arrivo
  - Stesso tipo di evento che ha causato la transizione
  - Stesso Pathid dell'oggetto DOM su cui è stato scatenato l'evento

- Solo i criteri di equivalenza T1 e T2 dipendono dai criteri di equivalenza per interfacce.



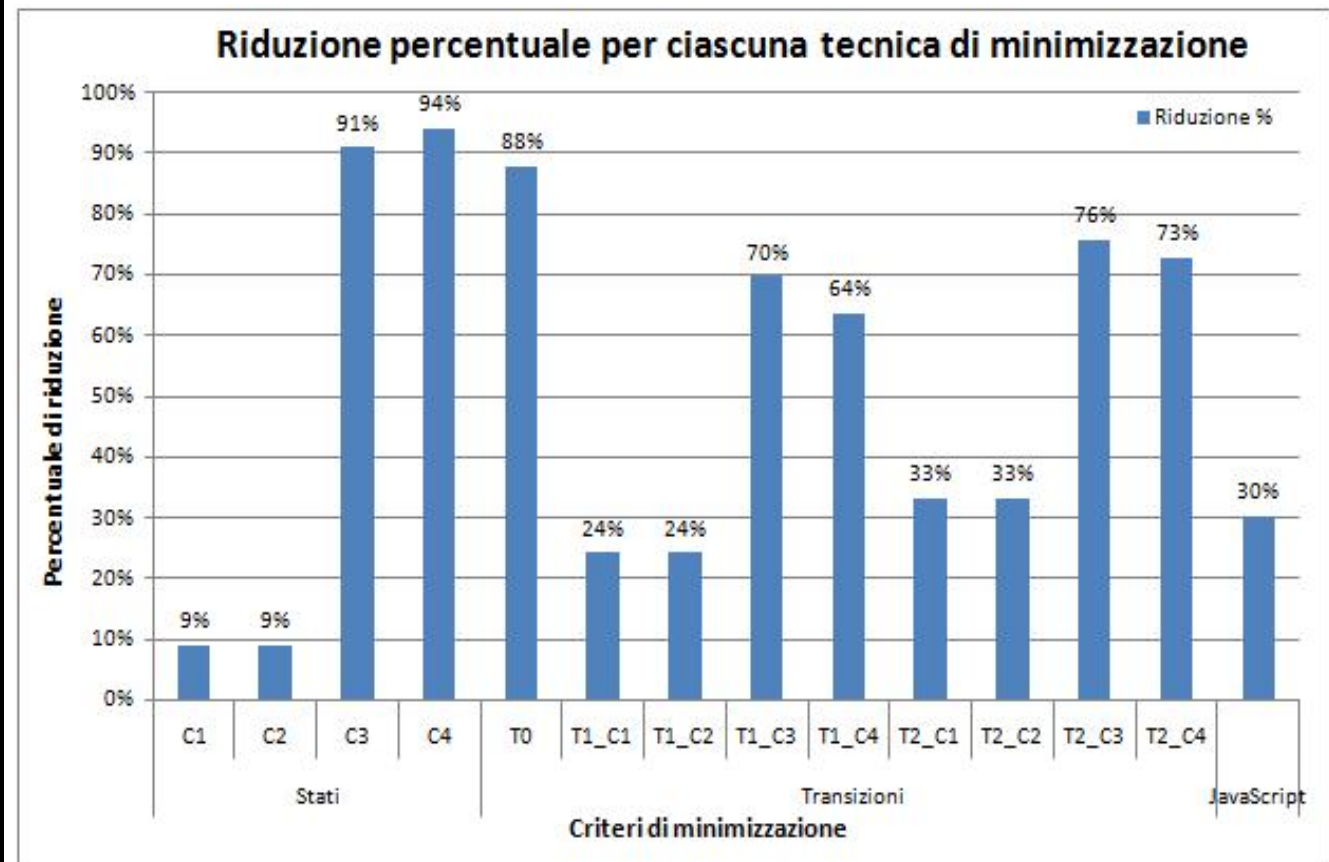
## Sperimentazione

- RIA target: **Tudu**
- Test Suite considerata **B1**:
  - ottenuta da un alpha tester
  - dimensione pari a 33 test case

- 14 Tecniche di riduzione suddivise in tre macro gruppi:
  - **Copertura di Stati**
  - **Copertura delle Transizioni**
  - **Copertura delle funzioni JavaScript**

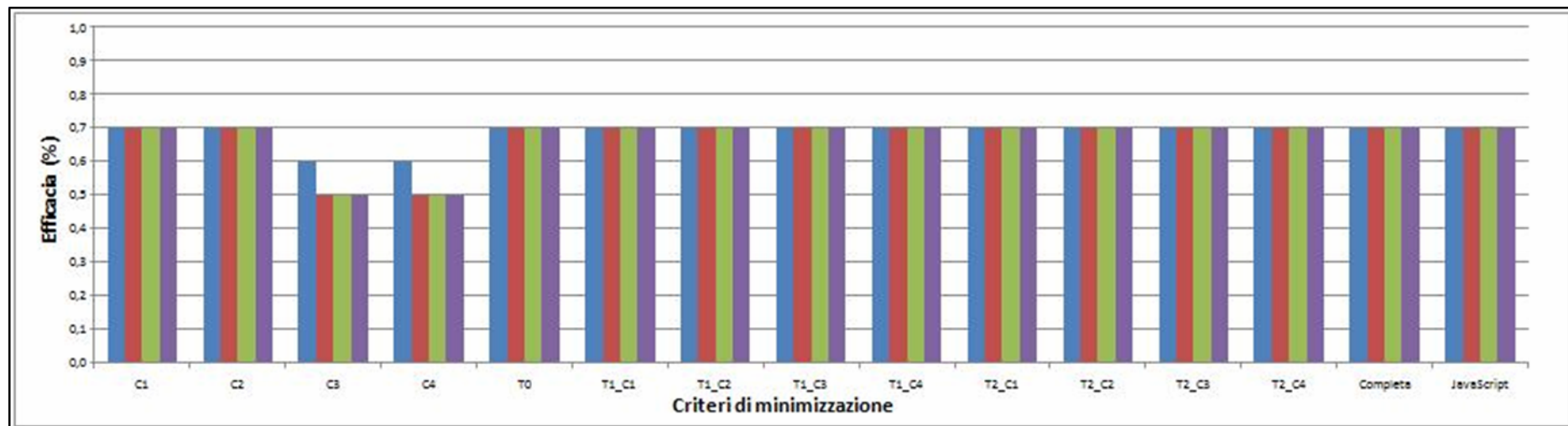
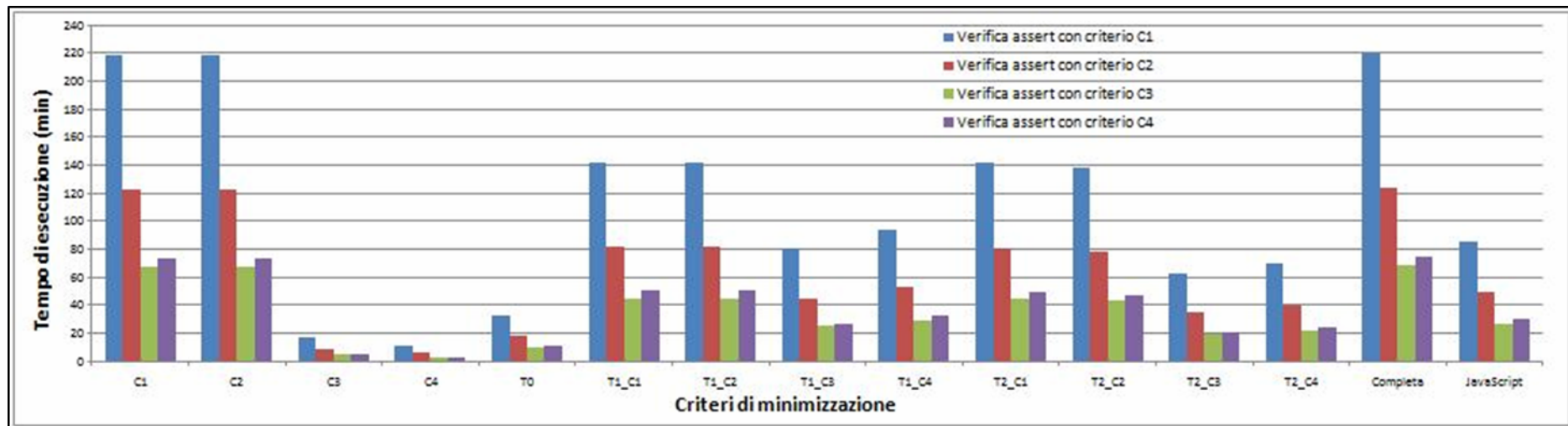
- **Obiettivo** : effettuare una comparazione delle tecniche di minimizzazione proposte.

- I criteri di minimizzazione C3, C4 e T0 effettuano una riduzione percentuale oltre l'80%.



## Sperimentazione

### Andamento dei tempi di esecuzione e dell'efficacia





## Conclusioni

- Abbiamo introdotto delle tecniche di minimizzazione che permettono di ricavare, a partire da Test Suite complete, Test Suite minime.
- Si è visto come l'efficacia e l'efficienza delle Test Suite minime dipendano dal criterio di minimizzazione scelto.
- Limitazioni:
  - Utilizzo di un'unica RIA target: Tudu.
  - Test Suite raccolte da un alpha tester.
  - Stesse condizioni iniziali per ciascun test case.
  - Numero e tipologia dei difetti iniettati

## Sviluppi futuri

- Sviluppi futuri posso riguardare:
  - ulteriori studi dei criteri di minimizzazione proposti che coinvolgono altre RIA reali, Test suite raccolte da un crawler e un numero più elevato di difetti iniettati.
  - l'introduzione di nuove tecniche di riduzione basate su una copertura dei cammini linearmente indipendenti sull'FSM o delle righe di codice eseguite.