MAC vs Forward Checking in Sudoku Resolution

Girolamo Macaluso

January 2021

1 Introduction

1.1 Sudoku as CSP

Il sudoku è diventato popolare nei magazine verso la fine degli anni 80, il gioco consiste nel riempire una tabella 9x9 con i numeri da 1 a 9, i vincoli sono che per ogni riga, colonna e regione, cioè sottoinsieme di celle 3x3, ogni numero deve comparire una sola volta. Questo puzzle è facilmente esprimibile come un csp in cui le variabili sono le celle non assegnate e vincoli sono di allDifferent.

1.2 Backtracking

Il backtraking è un ottimo algorimo per approcciare la risoluzione dei sudoku, poichè riesce sempre, se esiste, a trovare una soluzione. Il backtraking *"Brute Force"*, cioè senza inferenza, analizza tutti i possibili sotto problemi, questa soluzione risulta efficace ma poco efficente; gli algoritmi di inferenza presentati in seguito, MAC e forward checking, migliorano di molto le prestazioni.

2 Overview

2.1 Sudoku Generation

Il primo passo per costruire un test sulle varianti del backtracking è trovare i sudoku da testare, intuitivamente il miglior modo per generarli è crearne di completi ed eliminare causalmente delle celle; infatti questa è la soluzione adottata, ma con una variante: si controlla unicità della soluzione, poichè sudoku con soluzioni multiple sono eccessivamente semplici e quindi non adatti al test.

Un altro tema è stato la difficoltà dei sudoku, di cui non esiste una definizione univoca, essa è rimandata alla faciltà di risoluzione da parte degli algoritmi utilizzati. Per valutare la difficoltà e l'unicità delle soluzioni ho quindi usato `https://www.thonky.com/`, che è in grado di contare il numero di soluzioni e di assegnare un grado di difficoltà da 1 a 6. E' stato generato un set di puzzle (*puzzleD.txt*) di con diverse difficoltà (D) e un set con complessità randomica (*puzzle.txt*) tramite *GeneratePuzzle(difficulty)*.

2.2 General Implementation

L'implementazione di Backtraking differisce tra MAC e forward checking solo nella la parte dell'inferenza. Per la scelta della variabile da assegnare si utilizza l'euristica MRV. Per l'ordinamento delle variabili sono stati testati tre metodi di ordinamento: (1) ordine crescente, (2) ordine basato sul numero di occorrenze del valore nei vicini e (3) ordine basato sul less costraining value. Nella figura

1 è mostrato il risultato del test su 300 suduku utilizzanto la stessa versione di backtraking (in questo caso con MAC).

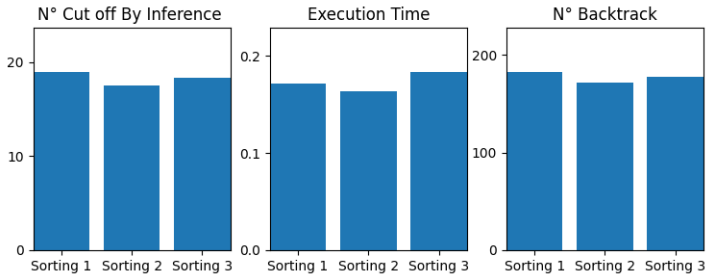

Figure 1: Sorting Test

I risultati mostrano come il secondo ordinamento sia il migliore, sia in termini di numero di backtrack che nel tempo di esecuzione; è questo il metodo utilizzato per i test successivi.
Il sudoku è rappresentato attraverso una matrice 9x9 ed è incapsulato all'interno di una classe che contiene anche il dominio per tutte le celle, esso è generato dalla funzione *GetDomain()* che controlla i valori preassegnati del puzzle e li elimina dal dominio dei vicini; nella classe sono presenti anche altri metodi ausiliari.

2.3 MAC And Forward Checking Implementation

I due algoritmi sono implementati in modo molto simile poichè forward checking esegue un sotto insieme dei controlli di MAC; i due condividono anche la funzione ausiliaria *Revise(x, y)*, responsabile di fare il cut off del valore di y dal dominio della variabile x. Il forward checking si limita ad eseguire revise per tutte le 20 variabili vincolate a quella considerata, e termina con False se il domino di uno di esse è vuoto dopo il cut off. MAC esegue lo stesso controllo, inoltre tiene traccia delle variabili a cui è stato ristretto il domino e controlla se esse sono in grado di rispettare i vincoli con i propri vicini.

3 Test

I test sono stati eseguiti su due tipi di dataset: uno contenente suduku di difficoltà casuale e l'altro sudduviso in base alla complessita con valore da 1 a 6. Nei test si terrà traccia del tempo di esecuzione medio e del numero medio di backtrack.

3.1 Random Sudoku

Entrambe le varianti sono state eseguite su un dataset di 300 sudoku di difficoltà casuale con un unica soluzione.

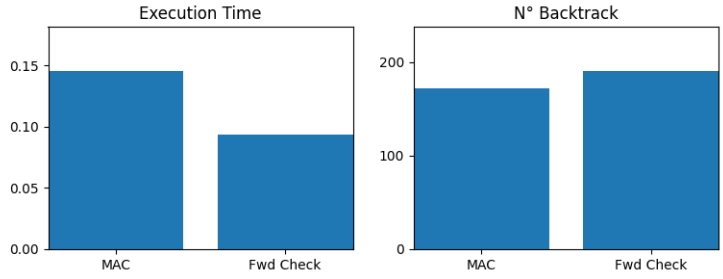

Figure 2: Random Sudoku Test

Dai risultati del test si evince come Forward Checking sia più veloce a trovare soluzione, si nota anche come il numero di backtrack eseguiti da MAC sia inferiore.

3.2 Defined Difficulty Sudoku

In questi test vengono presi in considerazione su dataset di sudoku con difficoltà definita. Questa prova ha lo scopo di individuare classi di difficoltà nel quale il rapporto tra MAC e Forward Checking varia, sembra sensato che un inferenza più potente dia il meglio di sè in sudoku più complessi. Di seguito sono mostrati i risultati con sudoku a difficoltà 2 e 6.

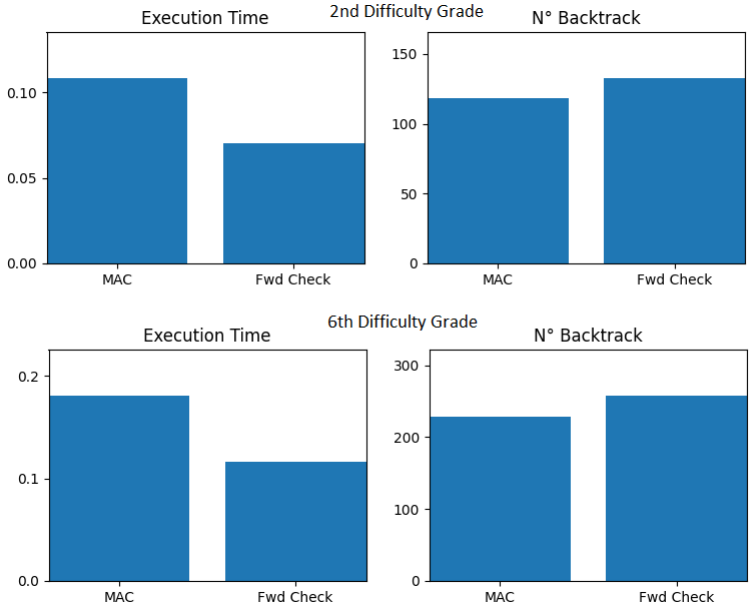

Figure 3: Defined Difficulty Sudoku Test

Si nota come al crescere della difficoltà, in valore assoluto, siano aumentati entrambi i parametri testati, ma che il rapporto tra i due tipi di inferenza è rimasto sostanzialmente identico.

3.3 No MRV Test

Per quanto riguarda il numero di problemi chiusi in anticipo grazie all'inferenza, esso risulta sostanzialmente uguale in entrambe le versioni. Questo è dovuto al fatto che MAC controlla se è possibile mantenere l'arc-consistancy, cioè è in grado di identificare quando due variabili con vincolo alldiff hanno dominio composto da uno stesso valore.
MAC scopre una deadend solo poche iterazioni prima del forward checking poichè si adotta l'euristica del MRV, che prende in considerazione per prime le variabili con i dominii più piccoli e quindi ha un'alta probabilità di scegliere una delle due variabili in conflitto. Di seguito è mostrato il risultato di un test senza utilizzare MRV, prendendo una variabile casuale ad ogni iterazione.

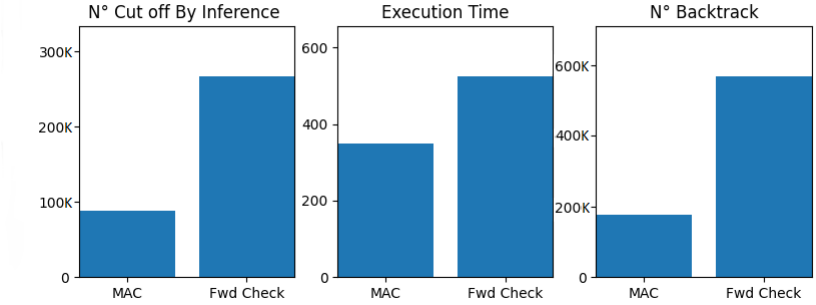

Figure 4: No MRV Test

É immediato notare come in questo caso MAC risulti molto più efficacie. I tempi di esecuzione sono comunque decisamente più alti della versione con MRV.

4 Conclusioni

MAC e Forward Checking sono due approcci molto simili all'inferenza nell'algoritmo backtraking, MAC è in grado di predire prima una deadend con un costo di esecuzione maggiore; Forward Checking invece è meno potente, ma comunque efficacie e soprattuto più veloce. La differenza tra i due viene alterata dalla presenza di MRV, che rende l'inferenza meno potente sempre la più rapida, l'unico punto a favore di MAC rimane il numero di backtrack che è sempre inferiore. In conclusione per la risoluzione di sudoku con MRV la scelta migliore è Forward Checking.