



Prendere decisioni con Programmazione con vincoli

Esercizio 2

Dessì Leonardo [0001141270]

Spini Riccardo [0001084256]

Corso di laurea magistrale in Informatica
Alma Mater Studiorum
Università di Bologna

Ottobre 2023

1 | Problema delle N-Regine

La tabella seguente riporta i guasti e il tempo totale del modello Alldifferent implementato utilizzando i vincoli globali o scomponendoli.

N	GC tutto diverso		Decomposizione	
	Non riesce	Tempo	Non riesce	Tempo
28	78.947 3	s 566 ms	417.027 9	s 168 ms
29	31.294 0	s 665 ms	212.257 2	s 196 ms
30	1.588.827 23	s 468 ms	7.472.978	1 minuto 7 s

Analizzando i risultati possiamo vedere che la presenza di vincoli globali ha un forte impatto su questo modello specifico per il problema delle n-regine.

Osservando i risultati possiamo notare che:

• **Numero di errori:** nella colonna "Alldifferent GC", il numero di errori è inferiore.

Ciò suggerisce che l'approccio basato sui vincoli globali è più efficace nell'eliminare soluzioni incoerenti e spesso porta a una convergenza più rapida.

• **Tempo di esecuzione:** l'approccio di "scomposizione" richiede più tempo nella maggior parte dei casi, indicando che è computazionalmente più costoso. L'"Alldifferent GC" è più veloce, probabilmente a causa di tecniche di propagazione specializzate associate a vincoli globali.

Le ragioni principali di tale comportamento sono:

I vincoli globali spesso portano a un'inferenza più potente durante la propagazione, che può ridurre più rapidamente lo spazio di ricerca. Questo è il motivo per cui i risultati di "Alldifferent GC" mostrano meno guasti. I vincoli globali sono ottimizzati per una propagazione efficiente e possono avere algoritmi dedicati che forniscono una propagazione forte ed efficiente.

La scomposizione in vincoli più semplici potrebbe non sempre fornire una propagazione efficace.

In alcuni casi, una semplice combinazione di sottovincoli non cattura tutta la potenza del vincolo globale. Inoltre, i vincoli globali forniscono un mezzo per garantire che tutte le variabili in un problema di vincoli siano simultaneamente coerenti con l'arco globale (GAC) o coerenti con la tenda (BC). Al contrario, quando si scompone un vincolo globale, in genere si ottiene la coerenza dell'arco solo su vincoli più piccoli, solitamente vincoli binari che coinvolgono coppie di variabili.

L'utilizzo di algoritmi ad hoc dedicati a specifici vincoli globali può essere più efficiente rispetto alla propagazione dei singoli vincoli della scomposizione. Gli algoritmi ad hoc spesso sfruttano il calcolo incrementale, nel senso che non ricalcolano tutto ciascuno

tempo, con conseguente esecuzione più rapida. Il calcolo incrementale nel contesto della programmazione dei vincoli si riferisce a un'ottimizzazione che evita di dover ripetere la soluzione di un problema da zero quando vengono apportate modifiche ai vincoli o alle variabili del problema. Infatti:

• Alla prima chiamata, alcuni risultati parziali vengono memorizzati nella cache.

• Alla chiamata successiva verranno utilizzati i dati memorizzati nella cache.

Inoltre, i vincoli globali aiutano a modellare problemi complessi in modo più conciso. Possono esprimere vincoli difficili da rappresentare utilizzando solo vincoli primitivi. Ciò è particolarmente vantaggioso quando il problema ha naturalmente strutture o modelli globali.

2 | Problema del manifesto

Il modello implementato è il seguente.

• Parametro:

- n: questo è un parametro intero che rappresenta il numero di poster.
- w: parametro intero che rappresenta la larghezza del foglio di carta.
- h: parametro intero che rappresenta l'altezza del foglio di carta.
- ws: array di lunghezza n contenente le larghezze dei poster.
- hs: array di lunghezza n contenente le altezze dei manifesti.

• Variabili:

- xpos: array di variabili intere, che rappresentano le posizioni orizzontali dei manifesti sulla carta.
- ypos: array di variabili intere, che rappresentano le posizioni verticali dei poster sul foglio di carta.

• Domini:

- $xpos[i]$: per ogni poster i assume valori compresi tra 0 e $w - 1$.
- $ypos[i]$: per ciascun poster i assume valori compresi tra 0 e $h - 1$.

• Vincoli:

– forall(i in $1..n$)($xpos[i] + ws[i] \leq w$): questo vincolo assicura che la somma delle coordinate orizzontali del poster i ($xpos[i]$) e della larghezza del poster stesso ($ws[i]$) non superino la larghezza totale del foglio di carta (w).

In altre parole, impedisce ai poster di uscire dal bordo destro del foglio.

– forall(i in $1..n$)($ypos[i] + hs[i] \leq h$): questo vincolo è simile al precedente, ma influenza le coordinate verticali e l'altezza dei poster, assicurando che i poster non escano dal bordo inferiore del foglio.

* Nel **modello Naïve** abbiamo utilizzato : forall(i, j in $1..n$ where $i < j$)($xpos[i] + ws[i] \leq xpos[j] \vee xpos[j] + ws[j] \leq xpos[i] \vee ypos[j] \geq ypos[i] + hs[i] \vee ypos[j] + hs[j] \leq ypos[i]$)

Questo vincolo garantisce che i poster non si sovrappongano tra loro.

* Abbiamo utilizzato nel vincolo **del modello globale** diffn ($xpos, ypos, ws, hs$): questo vincolo garantisce che i poster non si sovrappongano tra loro. Essa impone che le regioni occupate dai manifesti siano disgiunte.

La tabella seguente riporta i fallimenti e il tempo totale del Problema Poster implementato utilizzando il modello Naïve o il modello Globale.

Esempio	Modello ingenuo		Modello globale	
	Non riesce	Tempo	Non riesce	Tempo
19x19	1.678.013	15 s 769 ms	300.649	2 s 438 ms
20x20	2.504.120	23s 709ms	2.030	225ms

Analizzando i risultati possiamo vedere che la presenza di vincoli globali ha un forte impatto su questo modello specifico per il Problema del Poster.

Il **modello Naïve** utilizza la **scomposizione** dei vincoli, il che significa che il problema globale viene scomposto in **sottovincoli più piccoli e semplici**, ciascuno dei quali ha un algoritmo di propagazione noto. In questo caso, il modello Naïve potrebbe aver scomposto il problema in un **gran numero** di sottovincoli più piccoli. Ciò può portare a un'elevata **complessità computazionale** e richiede **molto tempo** per trovare una soluzione. Inoltre, il **tempo di calcolo** aumenta significativamente all'aumentare della dimensione del problema (ad esempio, da 19x19 a 20x20). Il modello Naïve richiede molto tempo per cercare una soluzione a causa della complessità della scomposizione e della mancanza di un'efficace propagazione globale.

Il **modello globale** utilizza vincoli globali specifici che **catturano meglio** la **complessa struttura** del **problema**. Questi vincoli globali incorporano la propagazione specializzata, che

sfrutta la struttura del problema per ridurre il numero di ricerche e il tempo necessario per trovare una soluzione. Il modello globale offre prestazioni molto migliori rispetto al modello Naïve in termini di tempo e numero di fallimenti. L'efficiente propagazione globale e il mantenimento della coerenza dell'arco globale (GAC) riducono significativamente i tempi di ricerca. Inoltre, il modello Globale può risolvere il problema 20x20 in meno di un secondo, mentre il modello Naïve ne impiega diversi minuti.

Il modello globale è più efficiente e performante perché incorpora vincoli globali specifici e utilizza la propagazione specializzata. Questi vincoli globali riducono significativamente il numero di soluzioni parziali da esaminare durante la ricerca e rendono più facile individuare soluzioni ottimali o soddisfacenti. Inoltre, il modello globale utilizza il calcolo incrementale, il che significa che può evitare di ricalcolare tutto ogni volta che viene chiamato, migliorando ulteriormente l'efficienza. In generale, i vincoli globali e la propagazione specializzata possono fare una differenza significativa nelle prestazioni del risolutore, soprattutto per problemi complessi come questo.

Le immagini seguenti mostrano una rappresentazione grafica di una soluzione al problema del poster con i dati dell'istanza specificati. Abbiamo implementato questa rappresentazione con uno script in Python. La figura (2.1) rappresenta una soluzione dell'istanza 19 x 19 e la figura (2.2) un Istanza 20 x 20.

Si noti che ci sono lacune nella figura (2.1) in posizioni sparse poiché:

$$\sum_{i=1}^N wsi \cdot hsi < w \cdot h$$

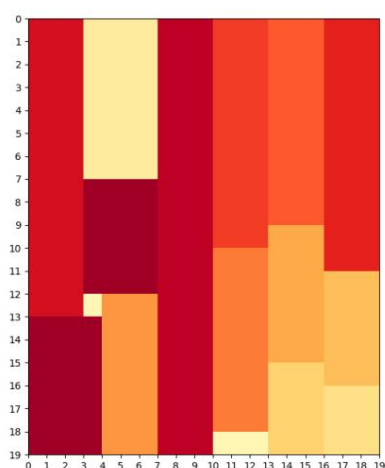


Figura 2.1: Soluzione 19 x 19

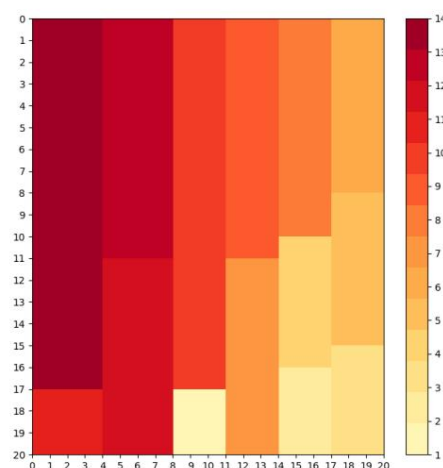


Figura 2.2: Soluzione 20 x 20

3 | Problema di sequenza

La tabella seguente riporta gli errori e il tempo totale del problema Sequence implementato da utilizzando il modello base, il modello base con vincolo implicito, il modello globale e il modello globale implicito con vincolo implicito.

N	Base		Base + Implicito		Globale		Globale + Implicito	
	Fallisce il tempo	Fallisce il tempo	Fallisce il tempo	Fallisce il tempo	Fallisce il tempo	Fallisce il tempo	Fallisce il tempo	Fallisce il tempo
500	618	34s 495	25 s 989 0 s 361 ms 493 0 s 215 ms					
1000	1.743 1h 21m 99s 1m 38s 1.989 1s 144ms 993 0s 487ms							

3.1 | Passando da Base a Globale

Si vede subito che il **modello Base** rispetto al **modello Global** ne ha meno fallimenti ma impiega molto più tempo per raggiungere la soluzione. Ciò accade perché gran parte di il tempo impiegato nel processo di risoluzione nella versione base viene impiegato nella compilazione. Ciò è dovuto al fatto che il compilatore genera centinaia di migliaia di valori booleani variabili per rappresentare il meta-vincolo. Infatti, come si può vedere nelle figure 3.1 e 3.2, il compilatore con il modello base con n = 500 genera centinaia di migliaia di variabili ausiliarie

```
%%%mzn-stat: evaluatedReifiedConstraints=250000
%%%mzn-stat: flatBoolVars=250000
%%%mzn-stat: flatIntConstraints=500500
%%%mzn-stat: flatIntVars=250500
%%%mzn-stat: flatTime=9.63938
%%%mzn-stat: method=satisfy
%%%mzn-stat: paths=0
%%%mzn-stat-end
```

Figura 3.1: Statistiche di compilazione con Modello base e n = 500

```
%%%mzn-stat: flatIntConstraints=1
%%%mzn-stat: flatIntVars=500
%%%mzn-stat: flatTime=0.106413
%%%mzn-stat: method=satisfy
%%%mzn-stat: paths=0
%%%mzn-stat-end
```

Figura 3.2: Statistiche di compilazione con Modello globale e n = 500

La differenza fondamentale tra l'utilizzo dei metavincoli e la scomposizione manuale è che nel primo caso il risolutore crea automaticamente queste variabili, mentre nel secondo caso il modello deve farlo manualmente. Pertanto, la compilazione lenta è dovuta principalmente a creando queste variabili booleane.

Abbiamo notato anche una grande differenza nella propagazione, come si può vedere nelle Figure 3.3 e 3.2

```

%%%mzn-stat: failures=618
%%%mzn-stat: initTime=7.229
%%%mzn-stat: nodes=1246
%%%mzn-stat: peakDepth=251
%%%mzn-stat: propagations=359141208
%%%mzn-stat: propagators=500500
%%%mzn-stat: restarts=0
%%%mzn-stat: solutions=1
%%%mzn-stat: solveTime=12.502
%%%mzn-stat: variables=500500
%%%mzn-stat-end
%%%mzn-stat: nSolutions=1
%%%mzn-stat-end
Finished in 38s 590msec.

```

Figura 3.3: Risoluzione delle statistiche con il modello Base e n = 500

```

%%%mzn-stat: failures=989
%%%mzn-stat: initTime=0.003
%%%mzn-stat: nodes=1984
%%%mzn-stat: peakDepth=251
%%%mzn-stat: propagations=8924
%%%mzn-stat: propagators=2
%%%mzn-stat: restarts=0
%%%mzn-stat: solutions=1
%%%mzn-stat: solveTime=0.247
%%%mzn-stat: variables=500
%%%mzn-stat-end
%%%mzn-stat: nSolutions=1
%%%mzn-stat-end
Finished in 449msec.

```

Figura 3.4: Risoluzione delle statistiche con il modello globale e n = 500

Questo perché, a nostro avviso, il modello Base avendo un **numero molto elevato di vincoli** richiede una **maggior propagazione**, invece nel modello Globale, il vincolo globale GCC ha una visione unica e globale su tutte le variabili decisionali coinvolte quindi di conseguenza non richiede tutte così tanta propagazione. Infatti possiamo vedere anche che il solveTime è molto più alto nel modello Base, proprio per il motivo spiegato sopra.

3.2 | Passando da Base + Implicito ÷ Globale + Implicito

In questo caso possiamo vedere che dalla tabella risulta che i fallimenti dei due modelli (con l'aggiunta dei vincoli impliciti) sono praticamente uguali. Notiamo inoltre che al variare di n, il **modello Base + Implicito** presenta due fallimenti in più rispetto al **modello Globale + Implicito**, a differenza dei modelli visti in precedenza in cui i fallimenti del modello Base erano significativamente inferiori.

Se parliamo del tempo di esecuzione totale, come prima, notiamo che il modello Base + Implicito impiega molto più tempo del modello Globale + Implicito. Ciò accade per lo stesso motivo di prima nel caso dell'implementazione senza implicita.

Mentre nel confronto **base ÷ globale** avevamo una differenza sostanziale sia tra il tempo di compilazione che quello di risoluzione, in questo caso, confrontando **base+implicito ÷ globale+implicito** la differenza è esclusivamente in tempo di compilazione. Nel secondo caso, infatti, i tempi di risoluzione rimangono ancora molto bassi e non vi è alcuna differenza sostanziale tra **base+implicito** e **globale+implicito** come possiamo vedere nelle figure 3.5 e 3.6. Questo, a nostro avviso, è dovuto al fatto che, grazie ai vincoli impliciti, entrambi i modelli dispongono di più informazioni utili sul problema e, di conseguenza, la propagazione effettuata è molto più efficiente.


```

%%%mzn-stat: failures=495
%%%mzn-stat: initTime=6
%%%mzn-stat: nodes=994
%%%mzn-stat: peakDepth=252
%%%mzn-stat: propagations=4229254
%%%mzn-stat: propagators=8876
%%%mzn-stat: restarts=0
%%%mzn-stat: solutions=1
%%%mzn-stat: solveTime=0.356
%%%mzn-stat: variables=500500
%%%mzn-stat-end
%%%mzn-stat: nSolutions=1
%%%mzn-stat-end
Finished in 24s 556msec.

```

Figura 3.5: Risoluzione delle statistiche con Modello base + Implicito e $n = 500$

```

%%%mzn-stat: failures=493
%%%mzn-stat: initTime=0.004
%%%mzn-stat: nodes=991
%%%mzn-stat: peakDepth=251
%%%mzn-stat: propagations=4962
%%%mzn-stat: propagators=3
%%%mzn-stat: restarts=0
%%%mzn-stat: solutions=1
%%%mzn-stat: solveTime=0.085
%%%mzn-stat: variables=500
%%%mzn-stat-end
%%%mzn-stat: nSolutions=1
%%%mzn-stat-end
Finished in 282msec.

```

Figura 3.6: Risoluzione delle statistiche con il modello globale + Implicito e $n = 500$

3.3 | Esiste un vincolo implicito che ora diventa ridondante in il modello Globale + Implicito? Perché?

Il primo dei vincoli impliciti non migliora il risultato, e il secondo vincolo implicito è sufficiente per esprimere lo stesso vincolo. Il **primo vincolo implicito** migliora il modello **base** perché la scomposizione del vincolo di cardinalità globale non ha la visione completa della sequenza, mentre il **vincolo globale** può **considerare** tutti i **valori contemporaneamente**. Il **vincolo di cardinalità globale** conta quante volte il valore v_j , assegnato alla variabile, appare nella sequenza e questo conteggio può influenzare il conteggio di altri valori. La **scomposizione dei vincoli**, invece, **crea** un **vincolo separato** per **ciascun valore**, perdendo così la visione globale del problema. Il **primo vincolo implicito non migliora la propagazione** perché fornisce informazioni che il **vincolo di cardinalità globale** già **possiede** a causa del **terzo parametro** che conta le occorrenze.

Ogni O_j conta quante volte v_j appare nella sequenza, fornendo così la cardinalità di X_i (dove X_i è assegnato a v_j). Infatti, il vincolo GC, può raccogliere i conteggi e limitare il numero di conteggi al numero di occorrenze di v_j e quindi può già conoscere tutte le informazioni del primo vincolo implicito.