Problemi di Soddisfacimento di Vincoli

Ing. Federico Bergenti

E-mail *federico.bergenti@unipr.it*
Telefono +39 0521 90 *6929*

Intelligenza Artificiale (AI)

- Ambizioso progetto nato già agli albori dell'Informatica
- Lo scopo era di realizzare
  - Macchine intelligenti
  - Macchine capaci di interagire con il mondo reale (robot)

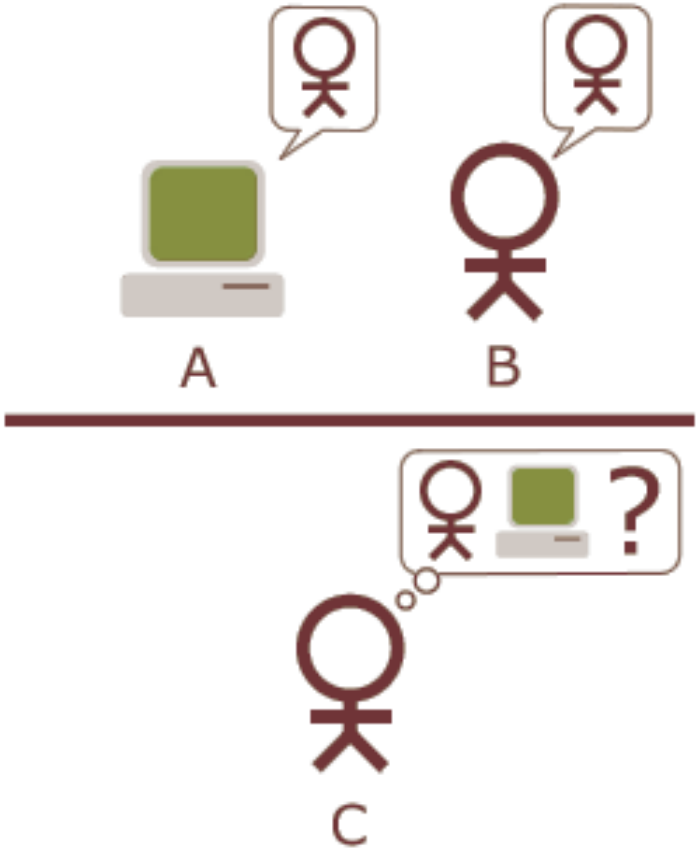
- Recentemente si cerca di realizzare
  - Macchine in grado di risolvere problemi complessi
  - Macchine dotate di comportamento razionale
  - Macchine capaci di interagire con mondi complessi e dinamici (Internet, il Web)

Test di Turing (1/2)

- Ma cosa caratterizza un *comportamento intelligente*?
- Alan Turing (1912-1954) propone nel 1948 un test empirico
- Il test di Turing è ancora utile per capire di cosa si occupa l'AI

Test di Turing (2/2)

- Due persone *B* e *C* sono giocatori di scacchi (inesperti)
- *A* è una macchina realizzata con l'unico scopo di giocare a scacchi
  - E.g., *Deep Blue* della IBM
- *C* è in una stanza isolata e gioca a scacchi, ma non sa se sta giocando contro *B* o contro *A*
- Se *C* non riesce a decidere se sta giocando contro *B* o contro *A*, allora *A* può essere considerato intelligente almeno quanto *B*!



Eliza (J. Weizenbaum, 1966)

- Ci sono stati vari tentativi di realizzare macchine che superassero il test di Turing
- Eliza si comporta come uno psicoterapeuta della scuola di Rogers
- Eliza si limita a
  - Identificare le parti della struttura delle frasi
  - Riformulare in modo interrogativo la frase appena ricevuta

```
> Hello, I am Eliza.
* Hi
> How are you today...What would you like to discuss?
* I have problems at school
> Can you elaborate on that?
* I don't pass exams
> Do you wish to be able to pass exams?
* Yes, sure
> I see.
* Math exams are too difficult for me
> Come, come, elucidate your thoughts.
* I don't see a meaning in math
> Do you wish to be able to see a meaning in math?
```

Problemi Matematici ed AI

- Il problema principale della AI
  - Non è tanto realizzare macchine intelligenti
  - È realizzare macchine che sembrino tali agli occhi dell'uomo
- Una macchina in grado di risolvere problemi che comunemente riteniamo complessi è tipicamente ritenuta intelligente
  - Risolvere velocemente un Sudoku
  - Risolvere un problema di Cripto-aritmetica
  - ...e tanti altri che citeremo nel seguito...
- Riferirci a problemi matematici ci permette di
  - Avere un criterio obiettivo per stabilire se una risposta è corretta o no
  - Disinteressarci dei problemi del linguaggio naturale
    - Comprensione delle frasi
    - Generazione di frasi di senso compiuto

Sudoku – Celle e Valori

Le celle possono assumere un valore numerico

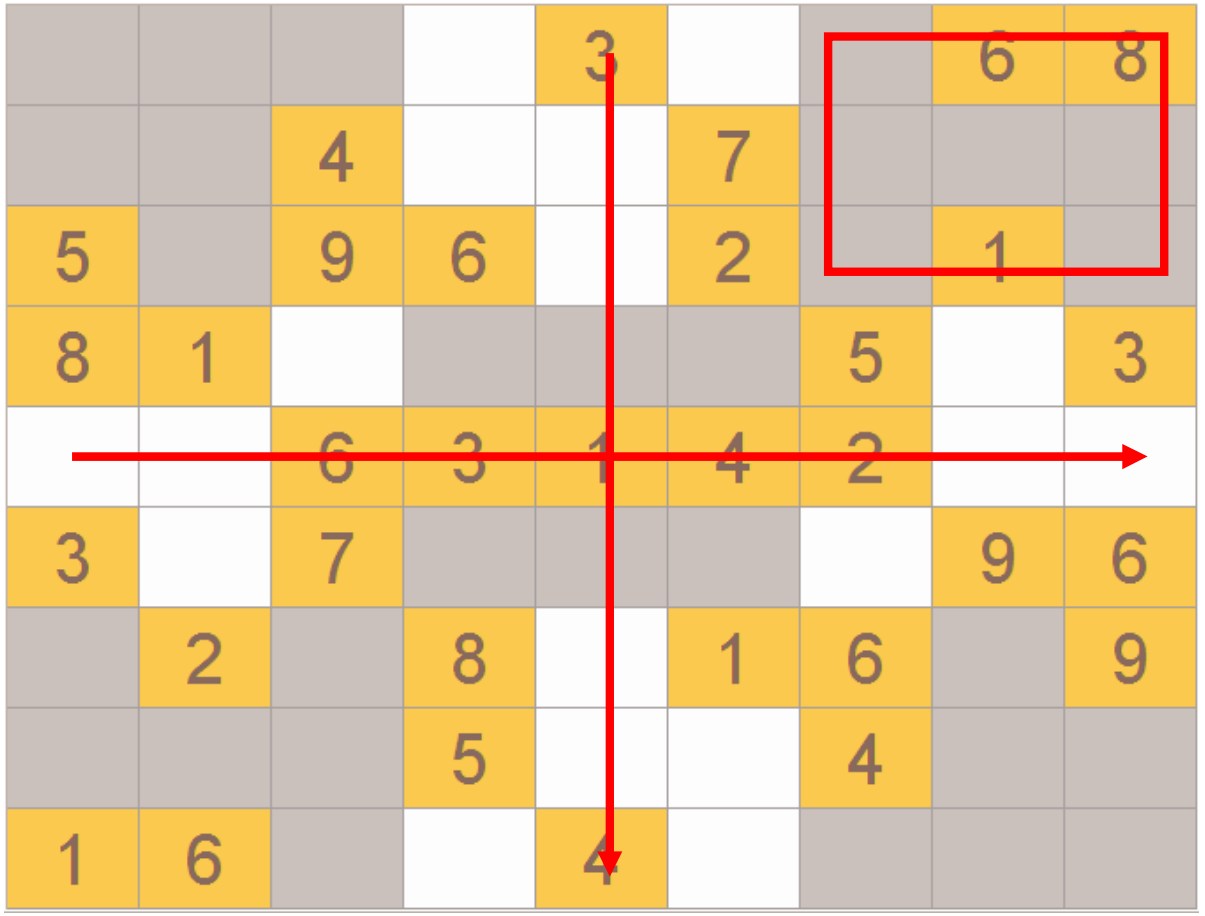
I valori numerici possibili per le celle sono gli interi compresi tra 1 e 9 (estremi inclusi)

Sudoku – Regole

Le regole sono semplici

I valori nelle celle devono essere tutti diversi tra loro

- Per ogni riga
- Per ogni colonna
- Per ogni blocco (3x3)



Problemi Cripto-Aritmetici

- Il problema è quello di associare ad ogni simbolo una cifra
- I simboli assumono un valore numerico
- I valori numerici possibili per i simboli sono gli interi tra 0 e 9 (estremi inclusi)
- Le regole sono
  - Simboli diversi sono associati a valori diversi
  - In questo caso ■≠0, ♠≠0

♠	♣	♥	♦	+
■	□	☺	♣	=

■	□	♥	♣	☻
9	5	6	7	+
1	0	8	5	=
1	0	6	5	2

Definizione di CSP

- Un Problema di Soddisfacimento di Vincoli (CSP, *Constraint Satisfaction Problem*) è composto da
  - Un insieme di variabili
    - Le celle del Sudoku, i simboli dei problemi Cripto-aritmetici
  - Un insieme di domini da cui attingere i valori da assegnare alle singole variabili
    - [1..9] per il Sudoku, [0..9] per il problemi Cripto-aritmetici
  - Un insieme di vincoli
    - Le regole del Sudoku o dei problemi Cripto-aritmetici

- Ogni variabile ha un suo dominio
  - Se X è una variabile, D_X è il suo dominio
  - D_X è l'insieme dei valori che X può assumere

Soluzione di un CSP

- Una soluzione di un CSP è

  Un isieme di valori presi dai domini che assegnati alle rispettive variabili soddisfano tutti i vincoli

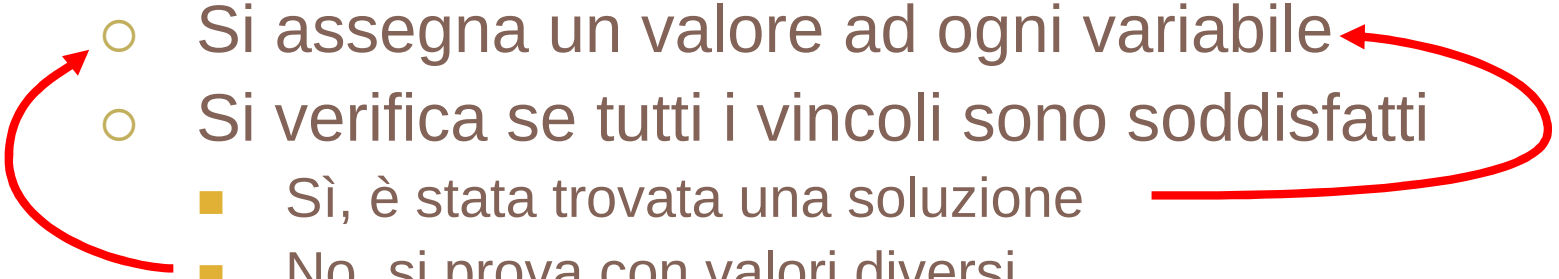
- Un CSP può avere
  - Una soluzione
    - Il Sudoku è ben posto
  - Zero soluzioni
    - Il Sudoku è sbagliato...c'è stato un errore di stampa
  - Più soluzioni
    - Il Sudoku non è ben posto...non dovrebbe succedere

CSP ed AI

- Molti problemi affrontati dalla AI sono in realtà CSP
- Molte tecniche sviluppate dalla AI si applicano bene alla ricerca di soluzioni di CSP
- Vediamo alcune tecniche che la AI ha proposto per risolvere i CSP
  - Sono molto generali ed applicabili sempre
  - Le vediamo applicate al Sudoku e ad altri problemi più semplici

Generate & Test

- La tecnica di soluzione più semplice è detta Generate & Test (G&T)
  - Si assegna un valore ad ogni variabile
  - Si verifica se tutti i vincoli sono soddisfatti
    - Sì, è stata trovata una soluzione
    - No, si prova con valori diversi

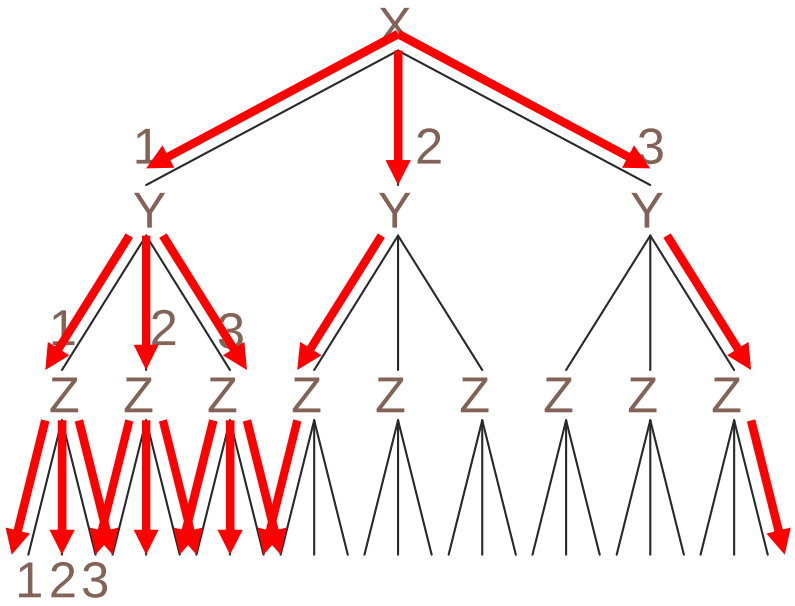


- Il procedimento continua finché non ci sono più assegnamenti nuovi da provare
  - Nel frattempo si è passati per (tutte) le soluzioni

Albero di Ricerca del G&T

Consideriamo il CSP

- Variabili X, Y, Z
- Domini $D_X=D_Y=D_Z=\{1, 2, 3\}$
- Vincoli X≠Y e Y≠Z e X≠Z



X	Y	Z
1	1	1
1	1	2
1	1	3
1	2	1
1	2	2
1	2	3
1	3	1
1	3	2
1	3	3
2	1	1
...		
3	3	3

G&T Applicato al Sudoku

- Si mette un numero in ogni cella vuota
- Si verifica che le regole siano rispettate
- Se qualche regola è violata, si prova con altri numeri

Caratteristiche del G&T

- Se abbiamo *V* variabili ed un dominio con *D* valori le foglie dell'albero sono

$$D^V$$

- Il G&T non è realistico per problemi complessi
  - Nel caso del Sudoku, con 31 celle già riempite, abbiamo $9^{50} \approx 5 \cdot 10^{47}$ possibilità da esplorare!

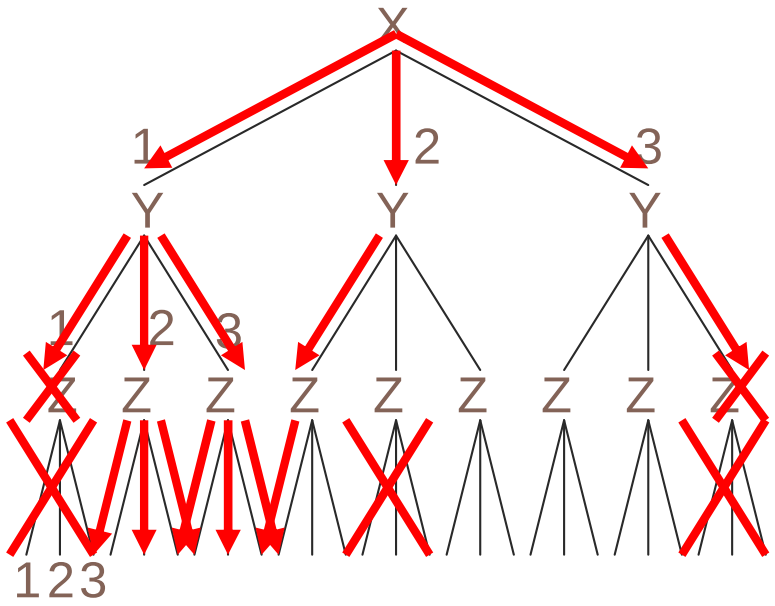
Standard Backtracking

- Modificando di poco il G&T si ottiene lo Standard Backtracking (SBT)
  - A seguito di ogni assegnamento si verifica se tutti i vincoli sono soddisfatti
    - Sì, possiamo continuare verso la soluzione
    - No, si verifica se la variabile appena assegnata ha ancora valori da provare
      - Sì, possiamo provare con un nuovo valore
      - No, si torna indietro e si sceglie una nuova variabile
- Il procedimento continua finché non ci sono più assegnamenti nuovi da provare
  - Nel frattempo si è passati per (tutte) le soluzioni

Albero di Ricerca dello SBT

Consideriamo il CSP

- Variabili X, Y, Z
- Domini $D_X=D_Y=D_Z=\{1, 2, 3\}$
- Vincoli X≠Y e Y≠Z e X≠Z



X	Y	Z
1	1	1
1	1	2
1	1	3
1	2	1
1	2	2
1	2	3
1	3	1
1	3	2
1	3	3
2	1	1
...		
3	3	3

Caratteristiche dello SBT

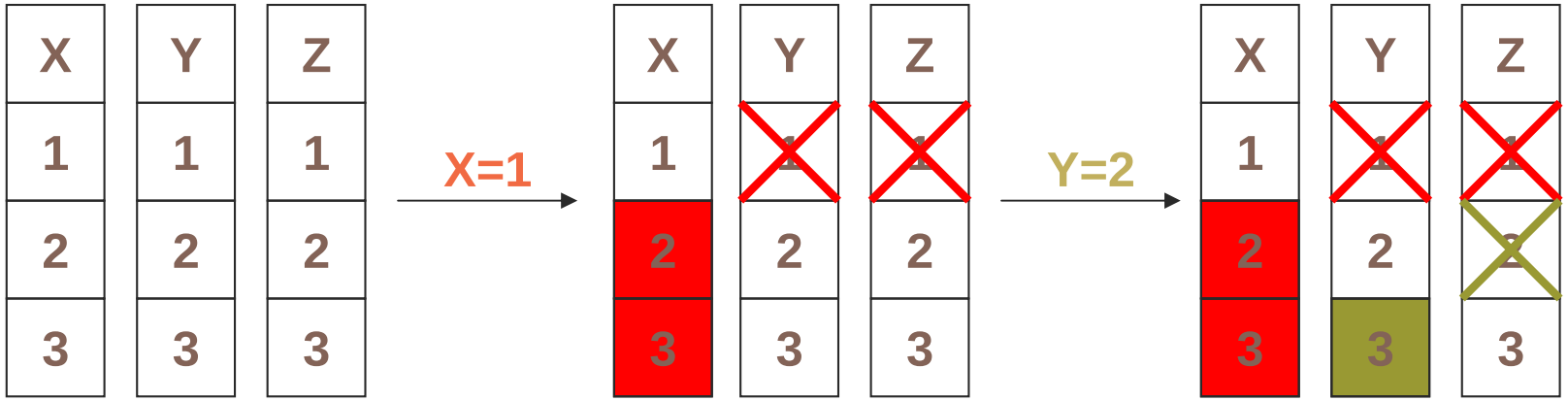
- Il limite teorico di $\mathcal{D}^{\mathcal{V}}$ foglie è sempre valido, però tipicamente non si provano tutte
- SBT sfrutta un principio molto generale

  Nell'attraversare l'albero conviene accorgersi delle inconsistenze il prima possibile

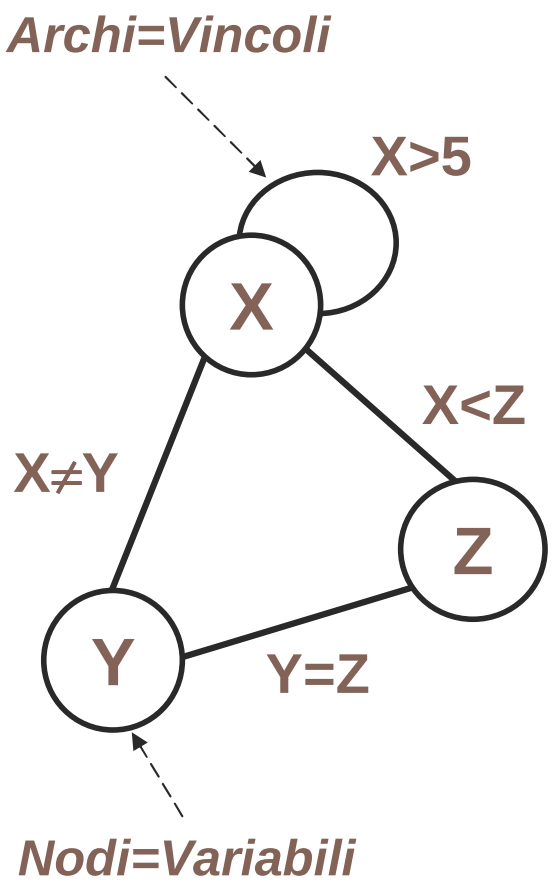
Propagazione dei Vincoli

- In generale

  Una volta assegnato un valore ad una variabile, i vincoli eliminano dei valori possibili dai domini delle altre variabili



Grafo dei Vincoli



- I vincoli possono essere
  - Unari, se coinvolgono solo una variabile
  - Binari, se coinvolgono solo due variabili
- Tutti i CSP possono essere ridotti a CSP binari, cioè contenenti solo vincoli unari e binari
- Per i CSP binari è possibile costruire un grafo dei vincoli

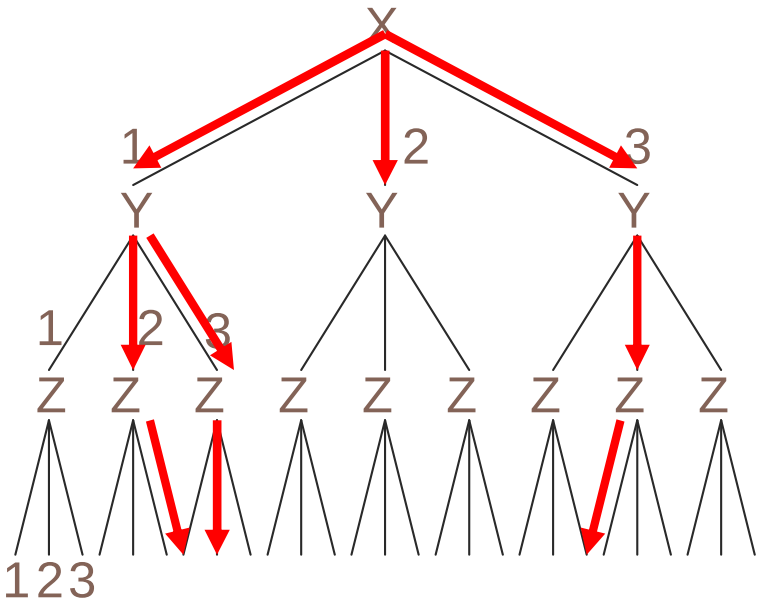
Maintaining Arc-Consistency

- Modificando lo SBT si ottiene il Maintaining Arc-Consistency (MAC)
- Dopo un assegnamento di una variabile, si sfrutta il grafo dei vincoli per togliere i valori non più ammissibili delle altre variabili
- Ha lo stesso comportamento dello SBT, ma ad ogni assegnamento i valori dei domini delle variabili vengono ridotti
  - E quindi si tagliano dei rami dell'albero

Albero di Ricerca del MAC

Consideriamo il CSP

- Variabili X, Y, Z
- Domini $D_X=D_Y=D_Z=\{1, 2, 3\}$
- Vincoli X≠Y e Y≠Z e X≠Z



X	Y	Z
1	1	1
1	1	2
1	1	3
1	2	1
1	2	2
1	2	3
1	3	1
1	3	2
1	3	3
2	1	1
...		
3	3	3

Caratteristiche del MAC

- In generale, MAC è un buon compromesso tra
  - Il numero di valori che tipicamente toglie dai domini
  - Il empo richiesto per l'attraversamento del grafo dei vincoli

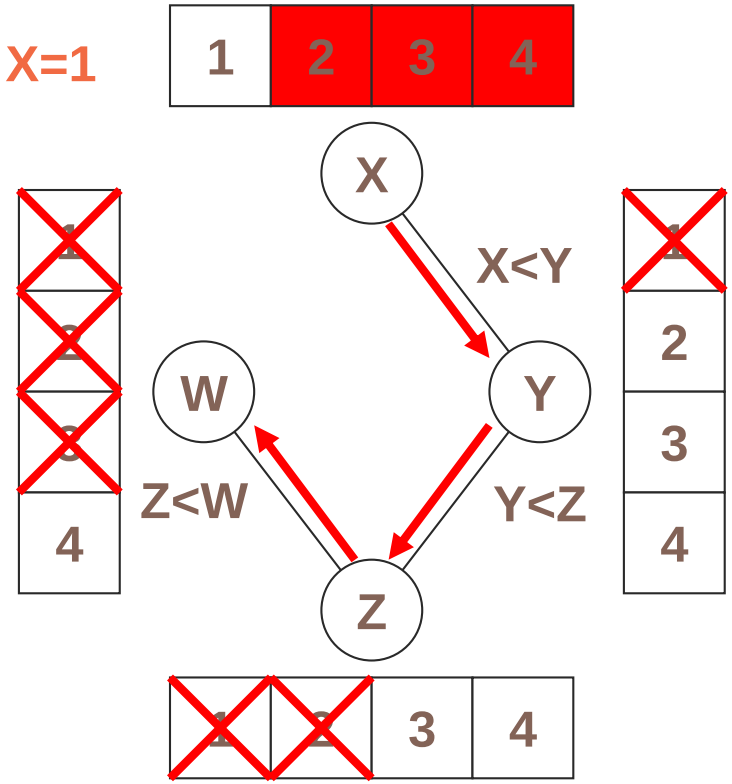
- MAC non è l'unico metodo di propagazione
- I metodi di propagazione richiedono tempo
  - Vantaggiosi se tolgono molti valori dai domini
  - Svantaggiosi se tolgono pochi valori dai domini

Forward Checking

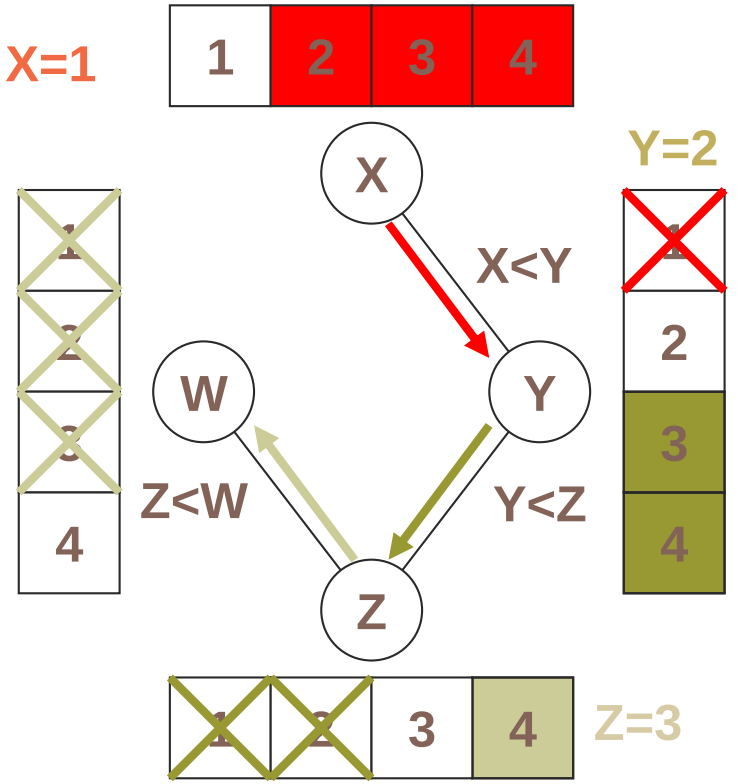
- Spesso si preferisce il Forward Checking (FC) al MAC
  - Variante del MAC che non attraversa tutto il grafo dei vincoli
  - Meno efficace nel rimuovere valori dai domini, ma più veloce nel propagare i singoli assegnamenti
- Vengono attraversate solo le variabili *più vicine* all'ultima variabile assegnata

AC e FC

Arc-Consistency



Forward Checking



Tecniche Euristiche

- Fino ad ora
  - Le variabili vengono assegnate sempre nel loro ordine naturale
  - I valori dei domini vengono assegnati alle variabili nel loro ordine naturale
  - Se una variabile non ha più valori nel proprio dominio si torna a quella immediatamente precedete nell'ordine naturale
- Non è necessario che sia così
  - Le tecniche euristiche spesso forniscono miglioramenti significativi

Dynamic Variable Ordering

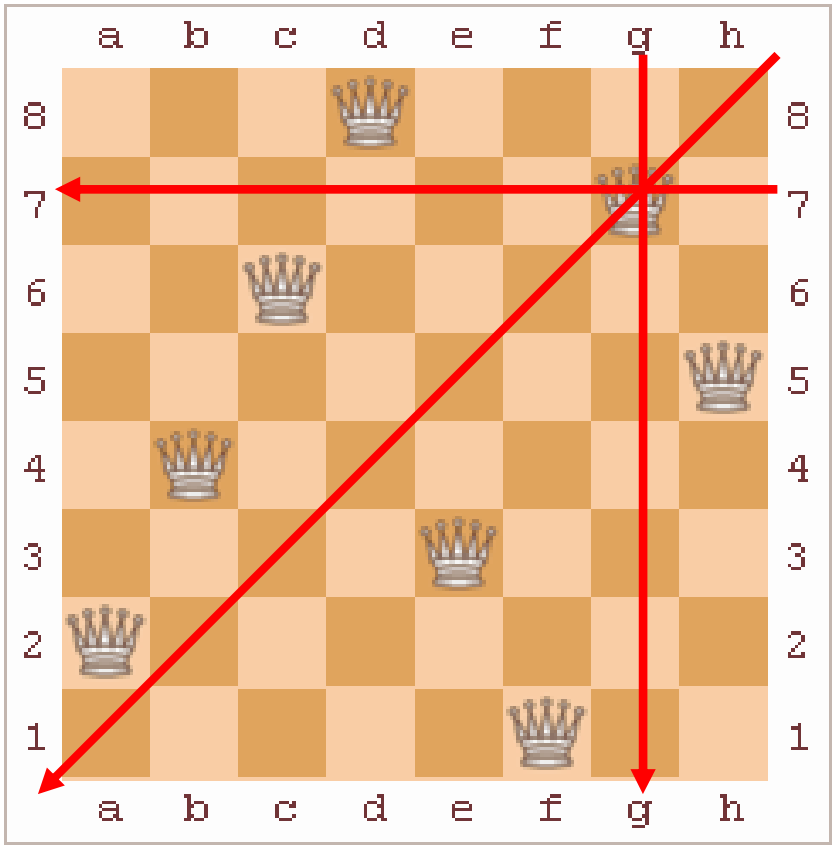
- La tecnica euristica Dynamic Variable Ordering (DVO) modifica la sequenza di assegnazione delle variabili seguendo un criterio semplice

  Conviene assegnare prima le variabili con domini più piccoli

Risultati Quantitativi

- Il Sudoku non è un problema parametrico
  - Non si riesce a modificarlo in base ad un parametro, e.g., il numero di celle nella scacchiera
- Per misurare le prestazioni delle tecniche viste conviene avere un problema parametrico
  - Per svincolarci dalle caratteristiche del caso singolo
  - Per poter stimare come si comporterebbero le varie tecniche in problemi che non riusciamo a sperimentare

Le *n*-Regine (1/2)



- Reso famoso da
  - Carl Friedrich Gauss (1777-1855) per *n*=8
  - Edsger Dijkstra (1930-2002) che lo usò per illustrare lo SBT
- Data una scacchiera *n*x*n*, il problema è posizionare *n* regine in modo che due regine non condividano
  - La stessa riga
  - La stessa colonna
  - La stessa diagonale

Le *n*-Regine (2/2)

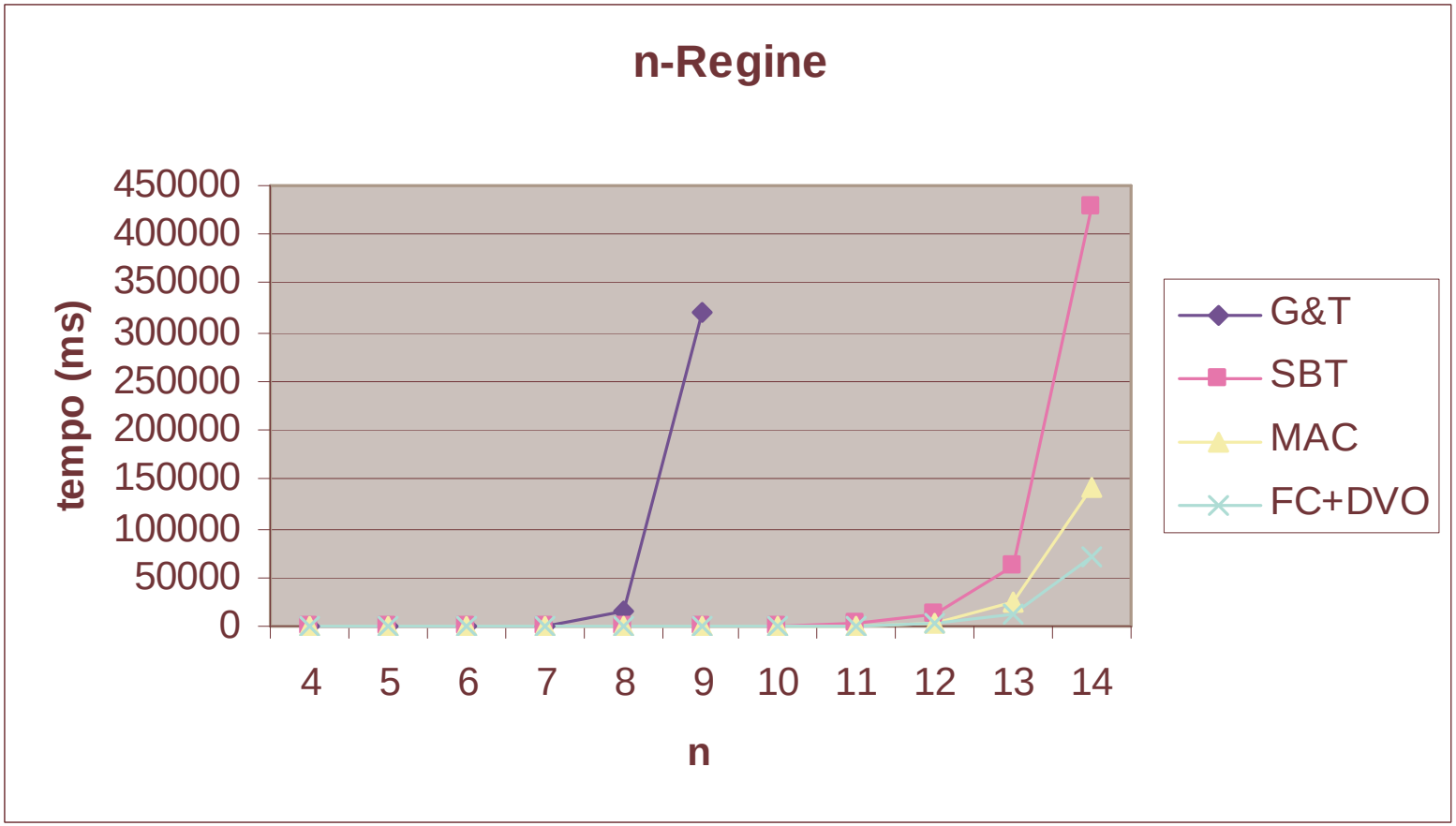
n	# soluzioni
4	2
5	10
6	4
7	40
8	92
9	352
10	724
11	2,680
12	14,200
13	73,712
14	365,596
15	2,279,184

- È un problema simile al Sudoku e ai problemi Cripto-aritmetici
- È sicuramente complesso
  - Basta guardare come varia il numero di soluzioni al variare di *n*
  - In più, per *n*=8 ci sono 283,274,583,552 possibili posizioni delle 8 regine, di cui solo 92 sono soluzioni
- È un esempio classico di CSP complesso

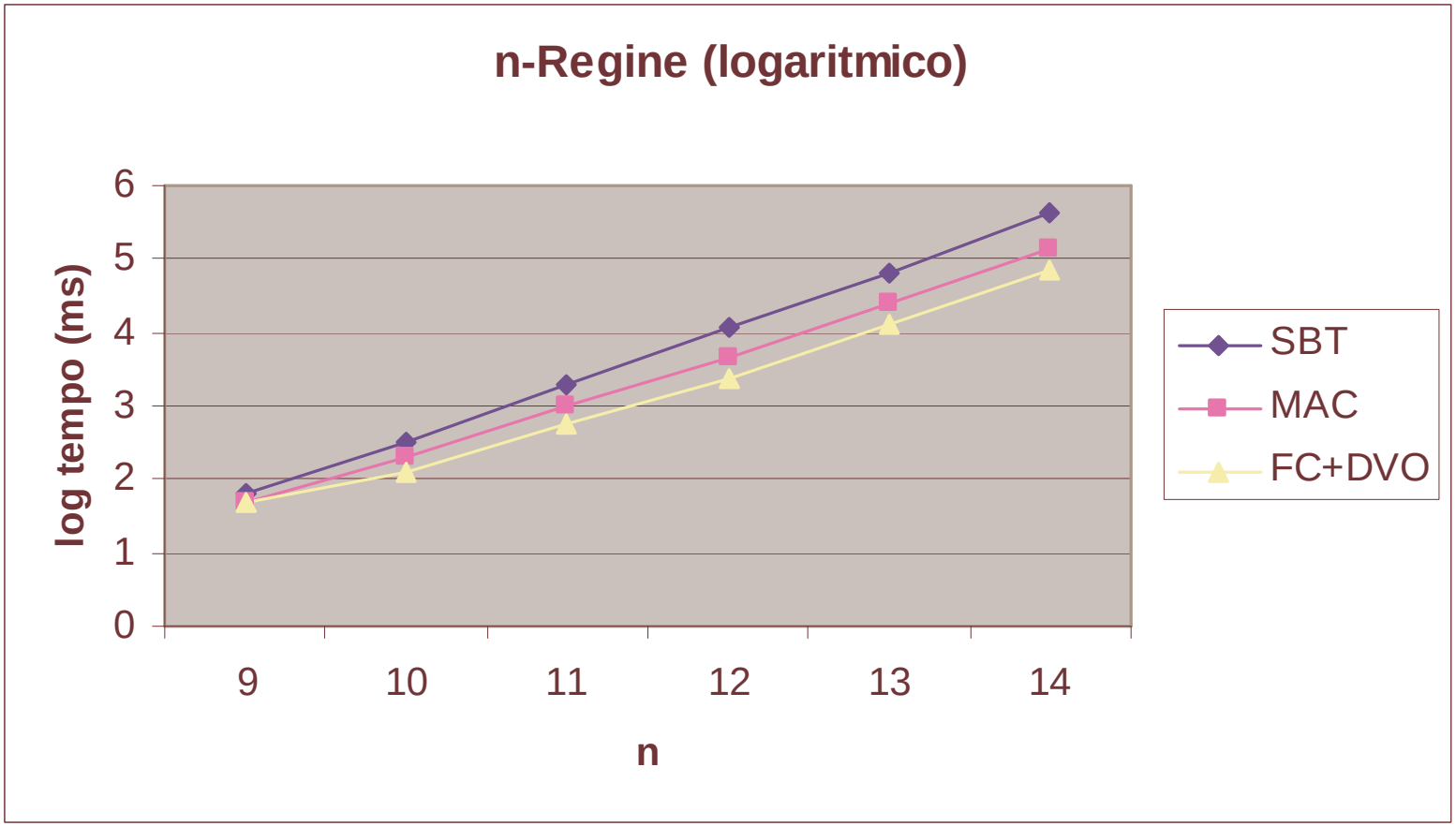
Confronto Quantitativo (1/3)

n	G&T (ms)	SBT (ms)	MAC (ms)	FC+DVO (ms)
4	0	0	0	0
5	0	0	0	16
6	32	0	0	31
7	625	0	0	32
8	15,188	0	15	31
9	321,828	63	47	47
10	> 1 ora	328	204	125
11		1,953	984	547
12		12,016	4,391	2,328
13		62,172	24,219	12,547
14		427,906	143,203	72,047

Confronto Quantitativo (2/3)



Confronto Quantitativo (3/3)



Problemi Intrattabili

- Per FC+DVO, dal grafico otteniamo

  $t = k10^{\alpha n}$

  $k \approx 1.5\ 10^{-5}$, $\alpha \approx 0.7$

- Gordon Moore (1929-) enuncia la famora legge di Moore che dice

  La velocità di una CPU raddoppia ogni 1.5/2 anni

- Fissato il tempo d'attesa, tra 2 anni quante regine in più riusciremo a posizionare?

  $t = 2k10^{\alpha n} = k10^{\alpha(n+\Delta)}$

  $\Delta \approx 0.4$

anni	velocità	n ($t \leq 0.5$s)
2	2GHz	11.4
4	4GHz	11.8
6	8GHz	12.2
8	16GHz	12.6
10	32GHz	13.0
12	64GHz	13.4
14	128GHz	13.8
16	256GHz	14.2
18	512GHz	14.6
20	1THz	15.0

Ma i CSP Servono Davvero?

"Constraint programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming…"

Eugene C. Freuder, Constraints, Aprile 1997

"Were you to ask me which programming paradigm is likely to gain most in commercial significance over the next 5 years I'd have to pick Constraint Logic Programming, even though it's perhaps currently one of the least known and understood."

Dick Pountain, BYTE, Febbraio 1995

Applicazioni

- Problemi di allocazione di risorse
  - Distribuzione degli aerei negli stand negli aeroporti
  - Distribuzione dei container nei porti e nei cargo
  - Pianificazione delle attività del personale
    - Gestione dei turni
    - Gestione dei periodi di carico
- Problemi di scheduling
  - Pianificazione della produzione
  - Produzione della tabella oraria di treni e trasporti urbani
- Gestione e configurazione di reti di telecomunicazione
  - Pianificazione della stesura dei cavi
  - Posizionamento degli access point in reti WiFi
- Progettazione di circuiti digitali
- Realizzazione di assistenti intelligenti per applicazioni Web

In Conclusione

- I CSP sono una classe molto grande e importante di problemi
- Trovano applicazioni pratiche nei campi più disparati
- La loro soluzione realizza comportamenti che possono essere apprezzati come intelligenti

- C'è ancora molta ricerca sull'argomento
  - Nuove tecniche particolarmente efficaci introdotte di recente
  - Nuove possibilità aperte da elaboratori sempre più potenti
  - Nuovi campi applicativi (Internet e il Web)