UNIVERSITÀ DEGLI STUDI DI FIRENZE – CDL IN INGEGNERIA INFORMATICA

Relazione progetto

Intelligenza Artificiale A.A. 2018/19

Lorenzo Bucci
08/02/2019

Introduzione

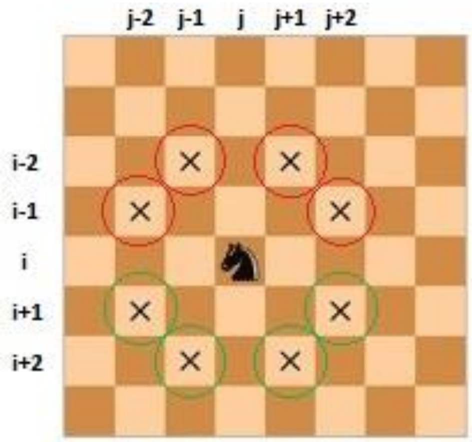
Testo: Si consideri la seguente variante del classico problema delle n regine. È assegnata una scacchiera *n x n* ed un intero *k*. Il problema consiste nel disporre *k* cavalli sulla scacchiera in modo da non avere attacchi. Si formuli il problema come CSP e si sviluppi un modello in un ambiente a scelta tra MiniZinc e Numberjack. Si studi empiricamente il tempo di risoluzione in funzione di *n* e *k* nei due modelli.

Il problema è stato formulato e risolto come CSP mediante la libreria Python "Numberjack" e l'uso del solver "MiniSat".

Le **variabili** sono contenute in una matrice *n x n* i cui valori appartengono al **dominio** {0,1}. Lo 0 rappresenta la cella vuota, il valore 1 rappresenta la presenza di un cavallo nella cella.

I **vincoli** che sono stati imposti sono due:

1. Il numero di cavalli sulla scacchiera deve essere uguale a *k.*
2. Per ciascuna cella *(i, j)* della scacchiera non può essere presente un cavallo sia nella cella *(i, j)* che nelle celle:
   a. *(i+1, j-2)*
   b. *(i+2, j-1)*
   c. *(i+2, j+1)*
   d. *(i+1, j+2)*



In una forma più estesa quest'ultimo vincolo può essere dedotto semplicemente dalla figura a fianco.
In realtà i vincoli per le celle presenti nelle righe *i-1* e *i-2*, cerchiati in rosso nell'immagine, sono superflui perché sono già stati considerati durante l'inserimento dei vincoli per le celle stesse (che sono precedenti alla *(i, j)*).
Per questo motivo è possibile dimezzare il numero di vincoli totali e diminuire sensibilmente i tempi di risoluzione del problema.

Durante la scrittura del codice e dopo aver eseguito i primi test è risultato palese che per risolvere il problema è sufficiente disporre i cavalli solo sulle caselle di un determinato colore in modo da avere una disposizione alternata nella scacchiera. Ovviamente questa forte euristica non è stata considerata altrimenti il CSP sarebbe degenerato in una semplice costruzione di una matrice a valori alterni.

Considerazioni test

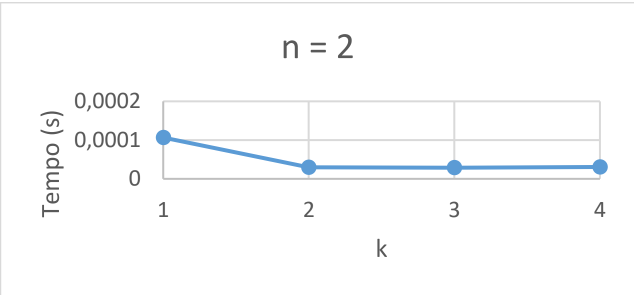
È stato svolto un test che verifica il tempo di risoluzione del problema (e non di caricamento dei vincoli) al variare di *n* e di *k*. Durante il test si è cercato di ridurre al minimo eventuali fattori esterni che avrebbero potuto falsare i risultati.

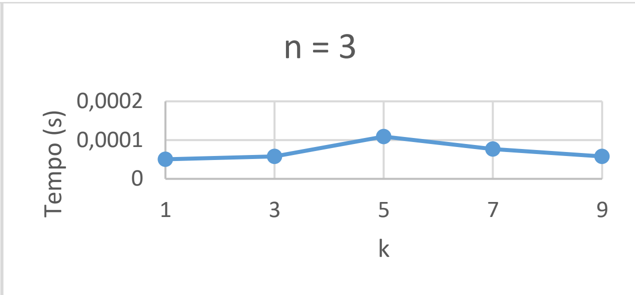
I **limiti** con cui è stato testato il problema sono:

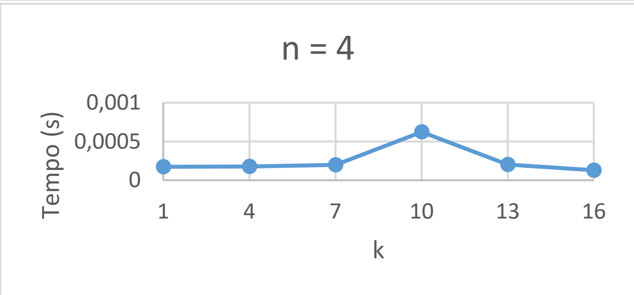
- *n = 18*, perché con *n = 19* il tempo di risoluzione ha superato i 30 minuti ed il test è stato interrotto volontariamente.
- $k = n^2$, per considerare il caso limite in cui ogni cella della scacchiera è occupata da un cavallo. Per ogni test su *n*, *k* è stato incrementato di *n – 1* partendo da *k = 1* (*k = 0* era privo di senso) in modo da avere *n + 2* tempi misurati.

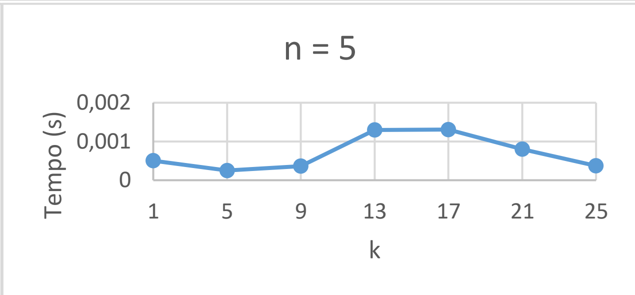
Risultati sperimentali

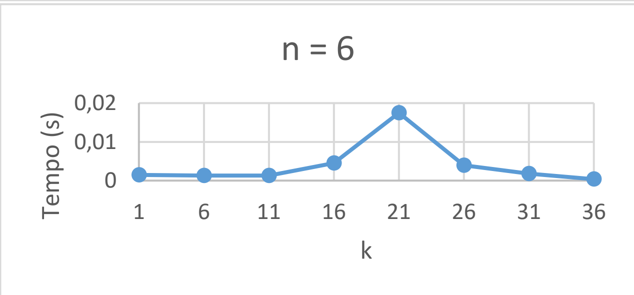
Di seguito sono riportati i grafici dei dati ottenuti dal test e contenuti nel file Excel risultante.

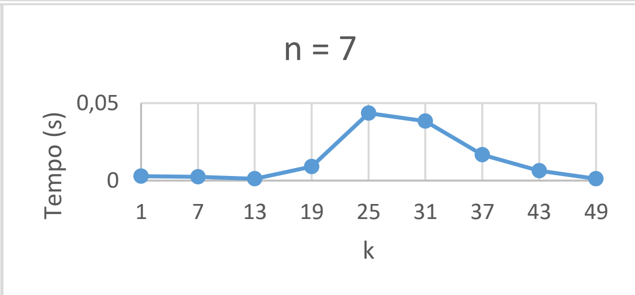


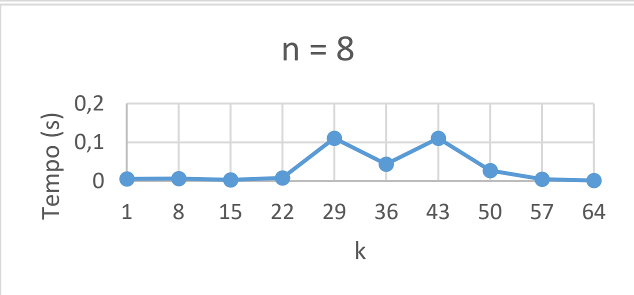


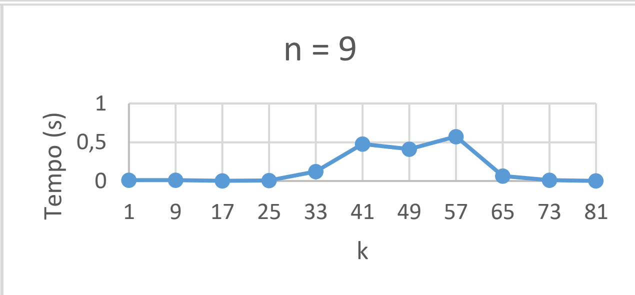


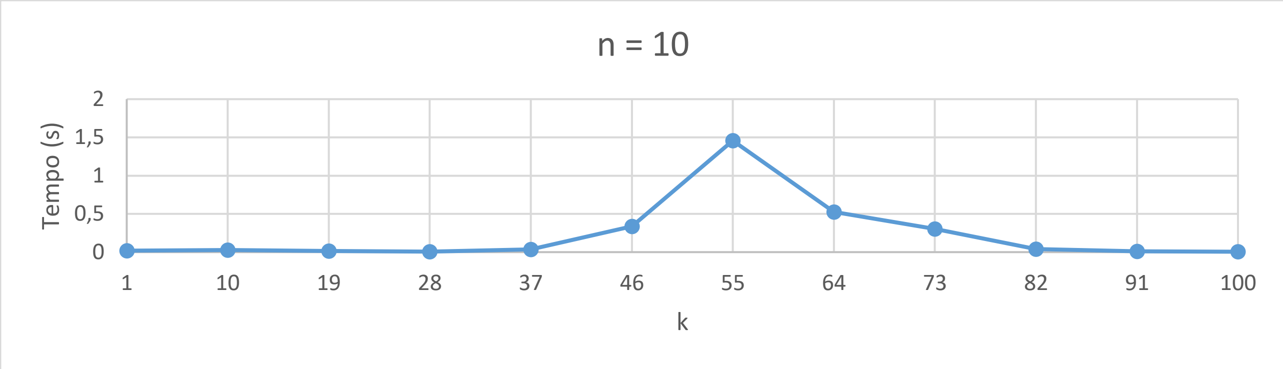


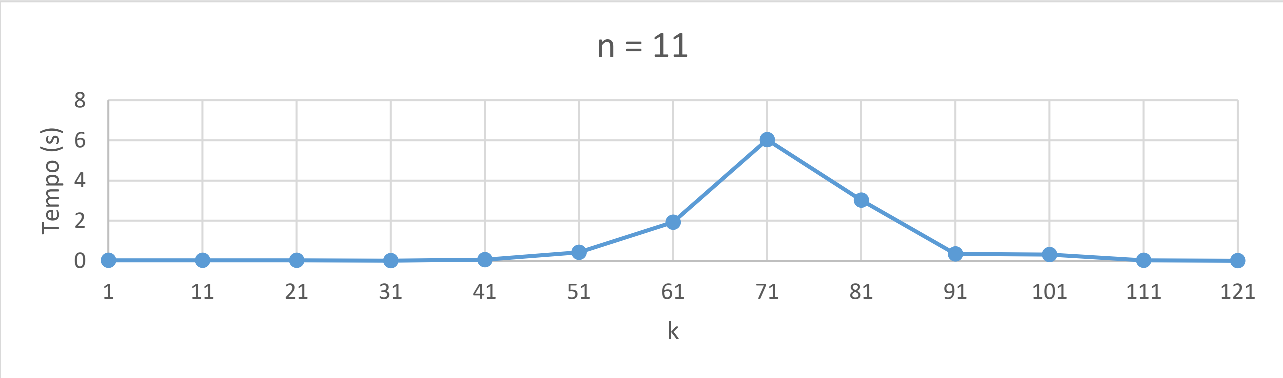


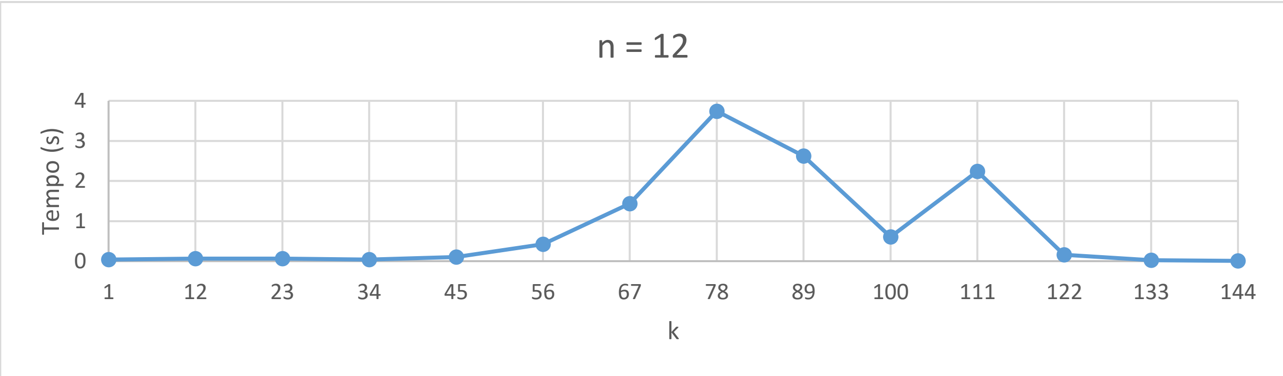


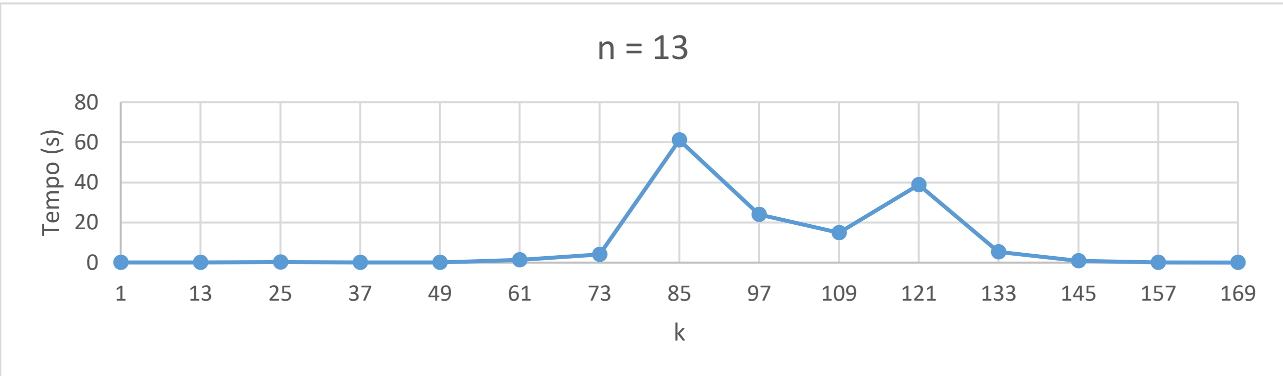


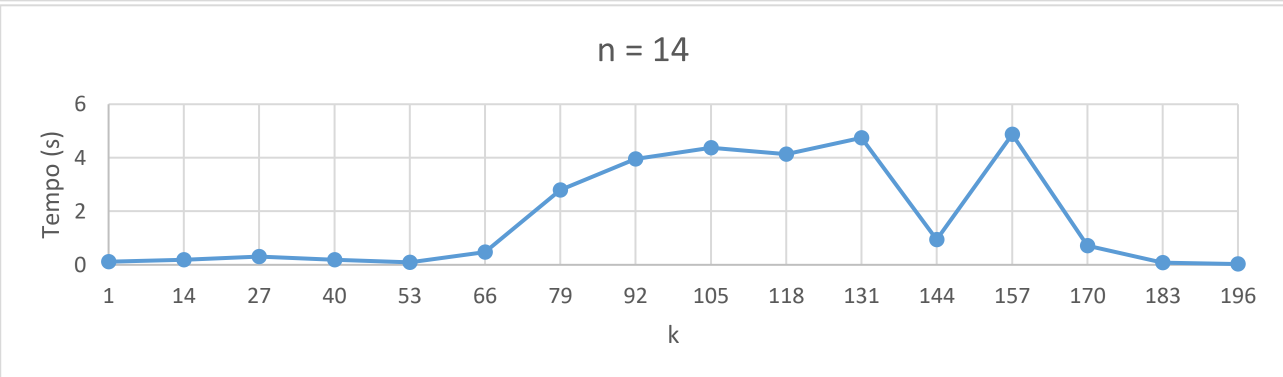


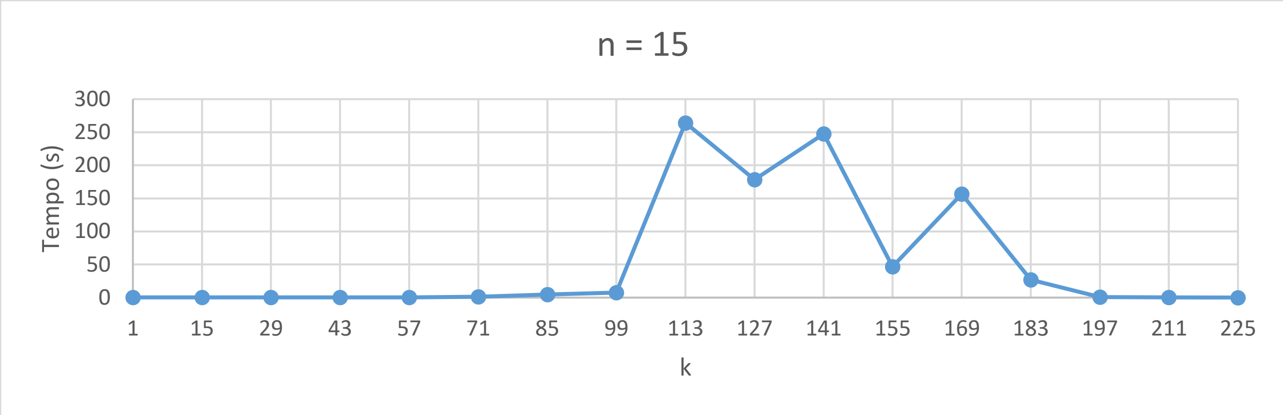

n = 10
Tempo (s)
2
1,5
1
0,5
0
1
10
19
28
37
46
55
64
73
82
91
100
k

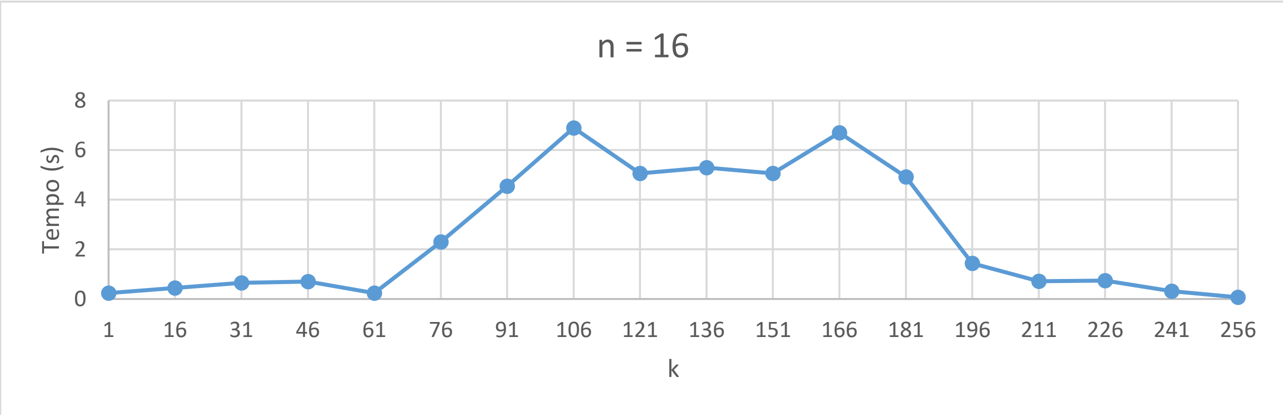

n = 11
Tempo (s)
8
6
4
2
0
1
11
21
31
41
51
61
71
81
91
101
111
121
k

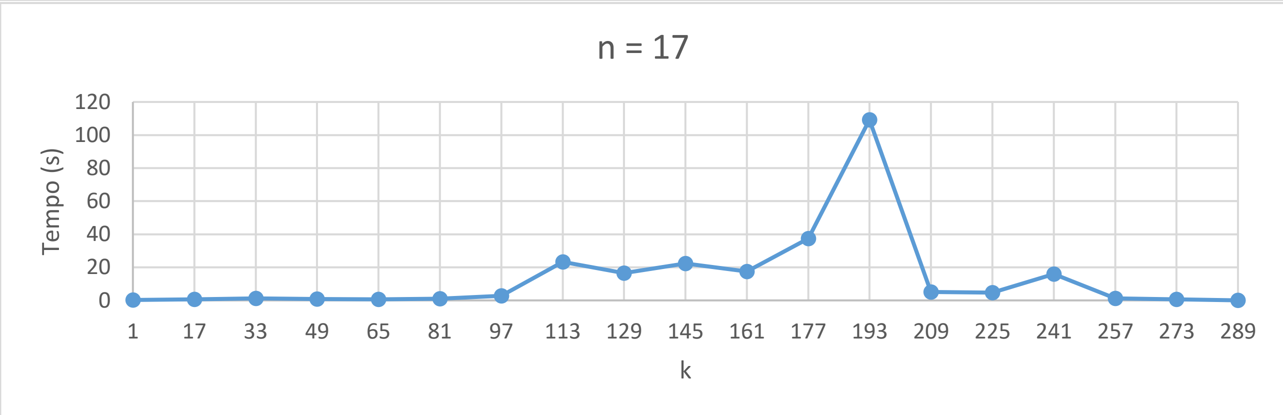

n = 12
Tempo (s)
4
3
2
1
0
1
12
23
34
45
56
67
78
89
100
111
122
133
144
k

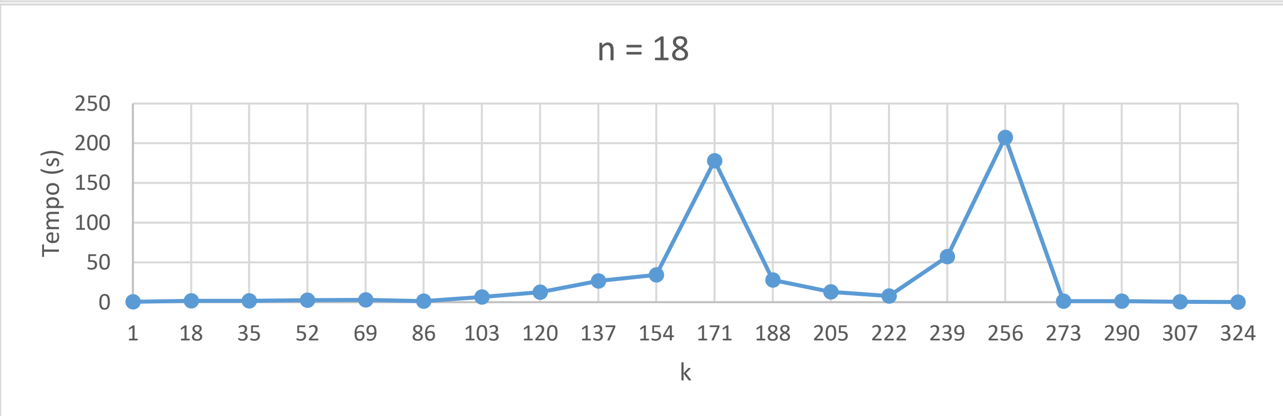

n = 13
Tempo (s)
80
60
40
20
0
1
13
25
37
49
61
73
85
97
109
121
133
145
157
169
k


n = 14
Tempo (s)
6
4
2
0
1
14
27
40
53
66
79
92
105
118
131
144
157
170
183
196
k

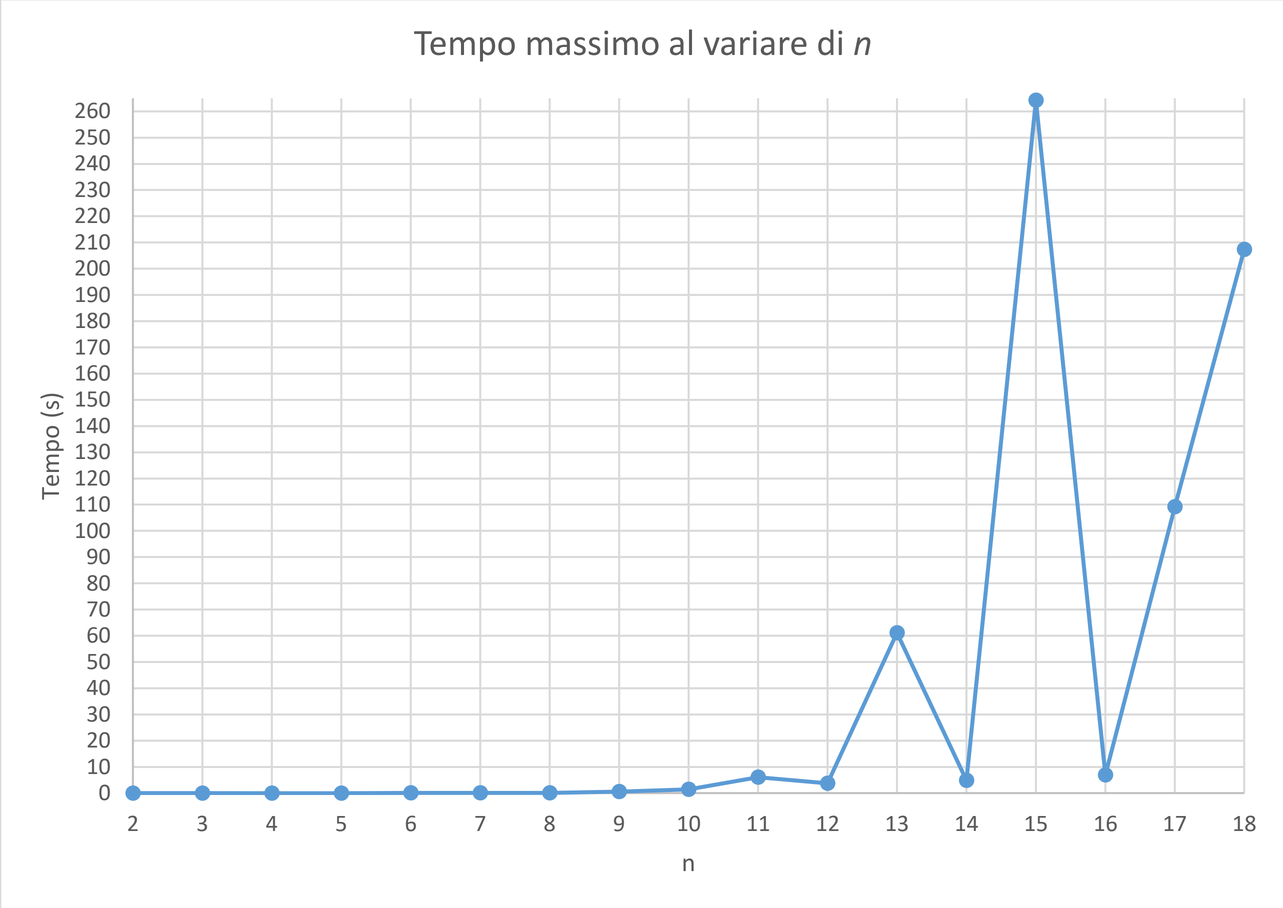

n = 15
Tempo (s)
300
250
200
150
100
50
0
1 15 29 43 57 71 85 99 113 127 141 155 169 183 197 211 225
k


n = 16
Tempo (s)
8
6
4
2
0
1 16 31 46 61 76 91 106 121 136 151 166 181 196 211 226 241 256
k


n = 17
Tempo (s)
120
100
80
60
40
20
0
1 17 33 49 65 81 97 113 129 145 161 177 193 209 225 241 257 273 289
k


n = 18
Tempo (s)
250
200
150
100
50
0
1 18 35 52 69 86 103 120 137 154 171 188 205 222 239 256 273 290 307 324
k

Di seguito è riportato un grafico che rappresenta il tempo massimo impiegato per ogni *n* tra tutti i *k* considerati per quell'*n*.



Conclusioni

Dai grafici si può dedurre che la complessità computazionale del problema diventa alta quanto vogliamo disporre un numero di cavalli *k* vicino alla soglia di soddisfacibilità che si trova a $\frac{n^2}{2}$. Prima di raggiungere questa soglia il problema ha una soluzione, dopo è impossibile trovarne una.
In alcuni casi si nota la stessa difficoltà anche con $k = \frac{3}{4}n^2$ quando il problema è insoddisfacibile.
Si può concludere quindi che il risolutore è molto veloce a: trovare una soluzione quando $k \ll n^2$ oppure a stabilire che il problema è insoddisfacibile quando $k \approx n^2$.

Si evidenzia inoltre una tendenza a risolvere molto più velocemente problemi con *n* pari anziché dispari dove il tempo di risoluzione è, solitamente, assai maggiore.