Relazione progetto Ricerca Operativa
Modello per la risoluzione del problema Kropki

Andrea Adorni, Alex Malavolta

Febbraio 2023

1 Introduzione

1.1 Cos'è kropki

Kropki è un gioco simile al sudoku, si compone di una matrice NxN con N un quadrato perfetto. La dimensione finale della matrice viene definita in base alla difficoltà del gioco che si vuole risolvere. È costituito da N righe e N colonne, le quali costituiscono la matrice principale di gioco. È composto ulteriormente da una sottomatrice nxn che costituirà le varie sezioni di gioco.

1.2 Regole

- Un numero non può essere presente più di una volta nella stessa riga
- Un numero non può essere presente più di una volta nella stessa colonna

- Un numero non può essere presente più di una volta nella stessa sezione
- Le celle in corrispondenza di un pallino nero devono essere una il doppio dell'altra
- Le celle in corrispondenza di un pallino bianco devono essere una il consecutivo dell'altra

Sono stati sviluppati due modelli risolutivi del problema kropki:

- Un modello semplificato, dove vengono dati dei numeri che devono essere mantenuti fissi durante la risoluzione.
- Un modello risolutivo generalizzato, dove non vengono forniti numeri iniziali e tutte le celle sono inizializzate vuote e si può scegliere la dimensione della matrice da risolvere.

2 Modello semplificato

Il gioco ammette una sola soluzione, quindi ci troviamo davanti ad un problema CSP (Constrain Satisfaction Problem). Non avremo quindi un obiettivo o di massimo, ma solo la ricerca di una soluzione ammissibile. Il modello matematico sviluppato è il seguente:

La tupla $[x, y, n]$ rappresenta rispettivamente il numero di righe, il numero di colonne e il numero associato.

$$\sum_{n=1}^{N} cella_{xyn} = 1 \quad \forall\, x, y \in \{1..N\} \tag{1}$$

Le variabili $cella_{xyn}$ sono variabili binarie che valgono 1 se la cella contiene il numero n, altrimenti 0. Di conseguenza, il vincolo impone

che all'interno di ogni singola cella sia presente al massimo un solo numero.
I seguenti tre vincoli riguardano le regole del sudoku.

$$\sum_{y=1}^{N} cella_{xyn} = 1 \quad \forall\, x, n \in \{1..N\} \tag{2}$$

$$\sum_{x=1}^{N} cella_{xyn} = 1 \quad \forall\, y, n \in \{1..N\} \tag{3}$$

Definiamo RIQ come un insieme discreto da 1 a N, con un passo incrementale pari alla radice di N.

$$\sum_{cx=rx}^{rx+\sqrt{N}} \sum_{cy=ry}^{ry+\sqrt{N}} cella_{cx\,cy\,n} = 1 \quad \forall\, rx \in RIQ,\, ry \in RIQ,\, n \in 1..N \tag{4}$$

Il vincolo (2) fornisce limitazioni sulle righe, ossia dice che ogni riga non deve contenere lo stesso numero più di una volta.
Il vincolo (3) fornisce limitazioni sulle colonne, ossia dice che ogni colonna non deve contenere lo stesso numero più di una volta.
Il vincolo (4) fornisce limitazioni sulle sezione, ossia la stessa sezione non può contenere due numeri uguali.

Passando ai vincoli aggiuntivi di Kropki:
Il vincolo (5) e (6) permettono di controllare che le celle aventi un pallino bianco in corrispondenza di due righe abbiano i numeri consecutivi.

$$cella_{b\,j\,k+1} + cella_{b\,j\,k-1} \geq cella_{ajk} \quad \begin{array}{l} \forall\, k \in \{1..N\} :\ k+1 < N+1 \wedge k-1 > 1 \\ \forall (j,a,b) \in BIANCOC \end{array} \tag{5}$$

$$cella_{a\,j\,k+1} + cella_{a\,j\,k-1} \geq cella_{bjk} \quad \begin{array}{l} \forall\, k \in \{1..N\} :\; k+1 < N+1 \,\wedge\, k-1 > 1 \\ \forall (j,a,b) \in BIANCOC \end{array} \tag{6}$$

Il vincolo (7) e (8) permettono di controllare che le celle aventi un pallino bianco in corrispondenza di due colonne abbiano i numeri consecutivi.

$$cella_{i\,b\,k+1} + cella_{i\,b\,k-1} \geq cella_{iak} \quad \begin{array}{l} \forall\, k \in \{1..N\} :\; k+1 < N+1 \,\wedge\, k-1 > 1 \\ \forall (i,a,b) \in BIANCOR \end{array} \tag{7}$$

$$cella_{i\,a\,k+1} + cella_{i\,a\,k-1} \geq cella_{ibk} \quad \begin{array}{l} \forall\, k \in \{1..N\} :\; k+1 < N+1 \,\wedge\, k-1 > 1 \\ \forall (i,a,b) \in BIANCOR \end{array} \tag{8}$$

Il vincolo (9) e (10) permettono di controllare che le celle aventi un pallino nero in corrispondenza di due righe abbiano i numeri uno il doppio dell'altro.

$$cella_{b\,i\,2k} + cella_{b\,i\,k/2} \geq cella_{aik} \quad \begin{array}{l} \forall\, k \in \{1..N\} :\; 2k < N+1 \,\wedge\, k \pmod 2 = 0 \\ \forall (i,a,b) \in NEROC \end{array} \tag{9}$$

$$cella_{a\,i\,2k} + cella_{a\,i\,k/2} \geq cella_{bik} \quad \begin{array}{l} \forall\, k \in \{1..N\} :\; 2k < N+1 \,\wedge\, k \pmod 2 = 0 \\ \forall (i,a,b) \in NEROC \end{array} \tag{10}$$

Il vincolo (11) e (12) permettono di controllare che le celle aventi un pallino nero in corrispondenza di due colonne abbiano i numeri uno il doppio dell'altro.

$$cella_{i\,b\,2k} + cella_{i\,b\,k/2} \geq cella_{iak} \quad \forall\, k \in \{1..N\} : \ 2k < N+1 \wedge k \pmod 2 = 0 \quad \forall (i,a,b) \in NEROR \tag{11}$$

$$cella_{i\,a\,2k} + cella_{i\,a\,k/2} \geq cella_{ibk} \quad \forall\, k \in \{1..N\} : \ 2k < N+1 \wedge k \pmod 2 = 0 \quad \forall (i,a,b) \in NEROR \tag{12}$$

Ipoteticamente questi vincoli bastano per risolvere l'istanza fornita dal problema. Se volessimo però applicare questo modello a istanze diverse, o anche più difficili, il modello non funziona.

3 Modello Generalizzato

Si è sviluppato quindi ulteriormente un modello generalizzato in grado di risolvere un problema di Kropki senza restrizioni iniziali e partendo da una matrice di dimensione N (scelta a piacere). Per la risoluzione di questo modello si è dovuto aggiungere ulteriori vincoli per quanto riguarda i pallini neri, andando a discriminare i numeri pari da quelli dispari e specificando le casistiche in cui i doppi sono maggiori della cifra massima. Di conseguenza, di seguito, verranno descritti i vincoli relativi ai soli pallini neri, visto che il resto è rimasto invariato:

$$cella_{b\,i\,2k} + cella_{b\,i\,k/2} \geq cella_{aik} \quad \begin{array}{l} \forall\, k \in \{1..N\} : \; 2k < N+1 \,\wedge\, k \pmod 2 = 0 \\ \forall (i,a,b) \in NEROC \end{array} \tag{13}$$

$$cella_{b\,i\,k/2} \geq cella_{aik} \quad \begin{array}{l} \forall\, k \in \{1..N\} : \; 2k > N+1 \,\wedge\, k \pmod 2 = 0 \\ \forall (i,a,b) \in NEROC \end{array} \tag{14}$$

$$cella_{b\,i\,2k} \geq cella_{aik} \quad \begin{array}{l} \forall\, k \in \{1..N\} : \; 2k < N+1 \,\wedge\, k \pmod 2 ! = 0 \\ \forall (i,a,b) \in NEROC \end{array} \tag{15}$$

$$cella_{b\,i\,k} = 0 \quad \begin{array}{l} \forall\, k \in \{1..N\} : \; 2k > N+1 \,\wedge\, k \pmod 2 ! = 0 \\ \forall (i,a,b) \in NEROC \end{array} \tag{16}$$

$$cella_{a\,i\,k} = 0 \quad \begin{array}{l} \forall\, k \in \{1..N\} : \; 2k > N+1 \,\wedge\, k \pmod 2 ! = 0 \\ \forall (i,a,b) \in NEROC \end{array} \tag{17}$$

I vincoli precedentemente descritti si riferiscono alle celle sopra e sotto ai pallini neri orizzontali. Il (13) impone che se in una casella c'è un numero pari il cui doppio è minore della cifra massima, allora il suo adiacente deve essere esclusivamente o il doppio o la metà. Il (14) impone che se in una casella c'è un numero pari il cui doppio è maggiore della cifra massima, allora il suo adiacente deve essere esclusivamente la metà. Il (15) impone che se in una casella c'è un numero dispari il cui doppio è minore della cifra massima, allora il suo adiacente deve essere esclusivamente il doppio. Il (16) e il (17) impongono che se in una casella c'è un numero dispari il cui doppio è maggiore della cifra massima, allora quel numero non deve essere presente nella cella.

$$cella_{i\,b\,2k} + cella_{i\,b\,k/2} \geq cella_{iak} \quad \begin{array}{l} \forall\, k \in \{1..N\} :\ 2k < N+1 \,\wedge\, k \pmod 2 = 0 \\ \forall (i,a,b) \in NEROR \end{array} \tag{18}$$

$$cella_{i\,b\,k/2} \geq cella_{iak} \quad \begin{array}{l} \forall\, k \in \{1..N\} :\ 2k > N+1 \,\wedge\, k \pmod 2 = 0 \\ \forall (i,a,b) \in NEROR \end{array} \tag{19}$$

$$cella_{i\,b\,2k} \geq cella_{iak} \quad \begin{array}{l} \forall\, k \in \{1..N\} :\ 2k < N+1 \,\wedge\, k \pmod 2 ! = 0 \\ \forall (i,a,b) \in NEROR \end{array} \tag{20}$$

$$cella_{i\,b\,k} = 0 \quad \begin{array}{l} \forall\, k \in \{1..N\} :\ 2k > N+1 \,\wedge\, k \pmod 2 ! = 0 \\ \forall (i,a,b) \in NEROR \end{array} \tag{21}$$

$$cella_{i\,a\,k} = 0 \quad \begin{array}{l} \forall\, k \in \{1..N\} :\ 2k > N+1 \,\wedge\, k \pmod 2 ! = 0 \\ \forall (i,a,b) \in NEROR \end{array} \tag{22}$$

I vincoli precedentemente descritti si riferiscono alle celle a destra e a sinistra dei pallini neri. Le tipologie di vincoli sono le medesime di quelle precedentemente esplicate.

4 Implementazione

Abbiamo suddiviso l'implementazione in due cartelle, (Kropki_generalizzato) contiene i file .mod,.dat,.run per la risoluzione del problema generale. (Kropki_normale) contiene i medesimi file ma con l'implementazione semplificata del problema. Per la risoluzione di entrambi i problemi si è utilizzato cplex come risolutore, in quanto la versione gratuita di gurobi non accetta problemi con un numero di vincoli e variabili sopra una certa soglia.

```
ampl: include kropki_generalizzato.run;
CPLEX 20.1.0.0: optimal integer solution; objective 0
0 MIP simplex iterations
0 branch-and-bound nodes
Objective = find a feasible point.
  2  4  1  9  7  8  6  3  5
  6  5  7  1  3  4  9  2  8
  9  3  8  5  6  2  7  4  1
  1  6  5  2  9  7  3  8  4
  8  9  3  4  1  6  2  5  7
  7  2  4  8  5  3  1  6  9
  3  1  6  7  4  5  8  9  2
  4  7  2  3  8  9  5  1  6
  5  8  9  6  2  1  4  7  3
ampl:
```

Figure 1: Risultato esecuzione generalizzata

```
ampl: include kropki_normale.run;
CPLEX 20.1.0.0: optimal integer solution; objective 0
0 MIP simplex iterations
0 branch-and-bound nodes
Objective = find a feasible point.
  2  4  1  9  7  8  6  3  5
  6  5  7  1  3  4  9  2  8
  9  3  8  5  6  2  7  4  1
  1  6  5  2  9  7  3  8  4
  8  9  3  4  1  6  2  5  7
  7  2  4  8  5  3  1  6  9
  3  1  6  7  4  5  8  9  2
  4  7  2  3  8  9  5  1  6
  5  8  9  6  2  1  4  7  3
ampl:
```

Figure 2: Risultato esecuzione normale

Ulteriormente nel file Kropki_generalizzato.dat sono stati inseriti i dati di una ulteriore istanza del problema .