Sudoku kot poseben primer problema prevoza

Dragana Božović

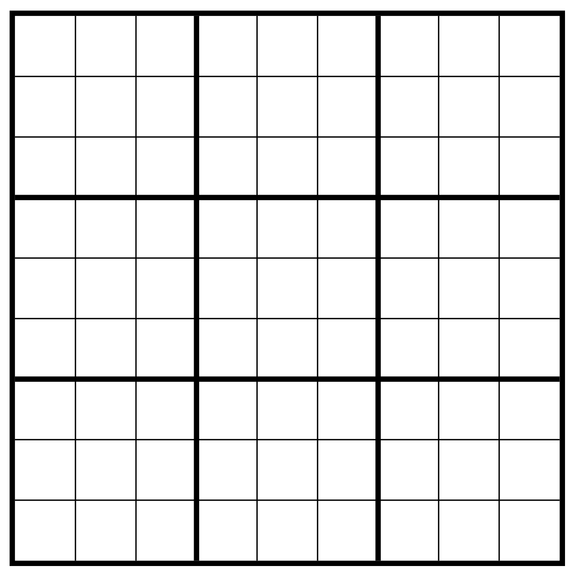
→ Čeprav je Sudoku na Japonskem postal priljubljen leta 1980, se je na zahod razširil šele leta 2004. Hitro je postal ena izmed najbolj uspešnih in priljubljenih ugank. Sudoku 9×9 mreža je danes vsakdanji del številnih časopisov. V svetu, kjer obstajajo aplikacije že za skoraj vse, tudi za Sudoku najdemo digitalne različice. V prejšnji številki Preseka [5] smo se naučili nekaj o problemu prevoza, tokrat se bomo poglobili v problem, ga povezali s Sudoku uganko in obravnavali Sudoku kot poseben primer problema prevoza.

Igra Sudoku

Spomnimo se najprej, kako je Sudoku uganka sploh sestavljena. Sudoku kvadrat lahko definiramo kot mrežo velikosti 9×9, pri čemer vsaka vrstica, vsak stolpec in vsak od devet označenih 3×3 blokov oz. kvadratov (slika 1) vsebujejo števila od 1 do 9, vsako natanko enkrat. Sudoku uganka je uganka, pri kateri so na začetku nekatera polja zapolnjena, nekatera pa prazna. Cilj reševalca je, da izpolni prazna polja tako, da je rezultat pravilen Sudoku kvadrat. Sudoku uganke delimo glede na težavnost na stopnje od 1 do 5 [2]. Med težavnostjo Sudokuja in številom začetnih števil, podanih v Sudoku mreži, ni nikakršne povezave. Obstajajo zelo težki Sudokuji, ki imajo zelo veliko podanih začetnih števil, in tudi zelo lahki, ki jih imajo zelo malo [3].

S priljubljenostjo igre so se pojavile še različne (drugačne in težje) vrste Sudokuja, med njimi [4]:

- *Sudoku-X*
  Pri tem Sudokuju moramo upoštevati še, da se morajo števila od 1 do 9 v diagonalah pojaviti le enkrat – tako nekateri Sudokuji pridobijo enolično rešitev.
- *Sodo-lihi Sudoku*
  Tak Sudoku ima določena polja osenčena. Na podlagi podane uganke razberemo, ali v osenčena polja vpisujemo soda ali liha števila.
- *Geometrijski Sudoku*
  Kvadrat tega Sudokuja je sestavljen iz različnih geometrijskih likov, imenovanih nonomine. Razmejeni so z debelejšo črto. Sestavlja jih devet polj, v katere je potrebno vpisati števila od 1 do 9.
- *Barvni Sudoku*
  To je Sudoku, pri katerem moramo upoštevati še, da se smejo števila od 1 do 9 v označenih obarvanih likih pojaviti le enkrat.



SLIKA 1.
Označeni 3×3 bloki oz. kvadrati

Sudoku kot poseben problem prevoza

Spomnimo se osnovne formulacije problema prevoza [5], s katerim minimiziramo stroške prevoza dobrin iz različnih izvornih točk (npr. skladišč) v različne končne točke (npr. trgovine):

- minimiziraj $z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{i,j} \cdot x_{i,j}$

  pri pogojih:

  $$\sum_{j=1}^{n} x_{i,j} \le s_i \quad \text{za } i = 1,\ldots,m$$

  $$\sum_{i=1}^{m} x_{i,j} \ge d_j \quad \text{za } j = 1,\ldots,n$$

  $$x_{i,j} \ge 0 \quad \text{za } i = 1,\ldots,m \text{ in } j = 1,\ldots,n$$

- Število s_i predstavlja zalogo enot v izvorni točki i, kjer je $i = 1,\ldots,m$.
- Število d_j predstavlja, koliko enot naroča končna točka j, kjer je $j = 1,\ldots,n$.
- S številom $c_{i,j}$ označimo, koliko stane prevoz ene enote od izvorne točke i do končne točke j, za $i = 1,\ldots,m$ in $j = 1,\ldots,n$.
- S številom $x_{i,j}$ označimo, koliko enot pošljemo od izvorne točke i do končne točke j, za $i = 1,\ldots,m$ in $j = 1,\ldots,n$.

Za podrobnejšo razlago si poglejte prejšnjo številko Preseka [5].

Reševanje uganke Sudoku zahteva, da 81 števil razporedimo v 81 celic Sudoku mreže (oštevilčenje celic vidimo na sliki 2).

Zato v preoblikovanju Sudoku uganke v problem prevoza uporabimo 81 končnih točk, ki jih moramo upoštevati, vsaka s povpraševanjem ene enote (v vsako celico zapišemo natanko eno število). Na voljo imamo 81 števil, in sicer števila od 1 do 9, vsako natanko devet krat. V enačbi 1 torej velja $m = 9$ in $n = 81$. Naj bosta s množica izvornih točk (v prejšnji številki Preseka [5] množica skladišč) in d množica končnih točk (v prejšnji številki Preseka [5] množica trgovin) takšni, da je $s_i = 9$, za $i = 1,\ldots,9$ (vsako od devetih števil se pojavi devet krat), in $d_j = 1$, za $j = 1,\ldots,81$ (vsak kvadratek Sudoku matrike 9×9 prejme eno število od omenjenih 81). Vsaka od devetih izvornih točk i lahko pošlje k vsaki od 81 končnih točk j eno enoto po ceni $c_{i,j}$. Ilustracija opisanega je prikazana na sliki 3.

V nadaljevanju bomo opisali, kako določimo matriko cen, ki je prisotna pri vseh oblikah problema prevoza. Postopek izračuna matrike cen je povzet po [1]. Pri modeliranju različnih Sudoku ugank, množici izvornih točk s in končnih točk d ostajata nespremenjeni. Strošek za ceno $c_{i,j}$ nastane pri prevozu ene enote od izvorne točke i do končne točke j. Zato je potrebno oblikovati matriko cen c, ki določi ceno prevoza ene enote od izvorne do končne točke.

S pomočjo osnovne Sudoku matrike bomo najprej definirali devet pomožnih matrik Π_i, $i = 1,\ldots,9$ (za vsako število od 1 do 9 eno), velikosti 9×9, nato bomo iz teh matrik sestavili končno matriko cen. Matriko Π_1 ustvarimo na naslednji način:

1. Začnemo z matriko ($i = 1$)

$$\Pi_1 = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}.$$

2. Poiščemo število 1 v osnovni matriki (če le-to obstaja).

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81

SLIKA 2.
Oštevilčenje celic.

3. Na mesto, kjer smo v osnovni matriki našli število 1, damo v matriko Π_1 število -1, vsa ostala mesta v isti vrstici, istem stolpcu in 3×3 kvadratu, v katerem se število nahaja, pa povečamo za 1.

4. Točki 2 in 3 ponovimo za vsako število 1 v osnovni matriki.

Na enak način ustvarimo vse preostale matrike Π_i, $i = 2, \ldots, 9$.

Poglejmo si, kako postopek deluje na konkretnem primeru. Vzemimo Sudoku uganko s slike 4 in izračunajmo Π_9 (opazujemo torej mesta, kjer se v osnovni matriki nahaja število 9).

Končno matriko cen sestavimo tako, da posamezna manjša matrika predstavlja eno vrstico velike matrike. Torej i-ta vrstica končne matrike je matrika Π_i, pri čemer vrstice matrike Π_i postavimo zaporedoma eno zraven druge in dobimo vrstico dolžine 81. Tako je dimenzija matrike cen, ki jo dobimo, enaka 9×81. Izkaže se, da ima za tako določeno matriko cen vsaka optimalna rešitev vrednost ciljne funkcije enako $z = -N$, kjer je N število števil, podanih v začetni Sudoku uganki.

Sedaj, ko imamo določeno tudi matriko cen, lahko zapišemo Sudoku kot poseben problem prevoza. Naj bo $M = \{1, 2, \ldots, 9\}, N = \{1, 2, \ldots, 81\}, R = \{1, 2, 3\}$. Formulirajmo omejitve našega problema prevoza:

$$\text{minimiziraj} \quad z = \sum_{i=1}^{9} \sum_{j=1}^{81} c_{i,j} \cdot x_{i,j}$$

1. $\sum_{j=1}^{81} x_{i,j} = s_i \qquad i \in M$

2. $\sum_{i=1}^{9} x_{i,j} = d_j \qquad j \in N$

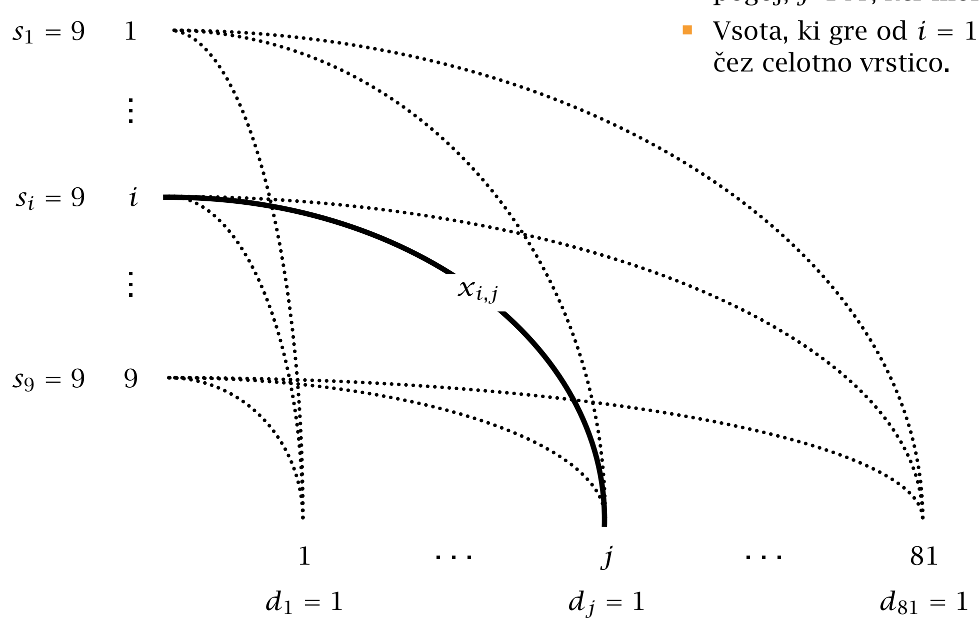
3. $x_{i,j} \in \{0, 1\} \qquad i \in M, j \in N$

Te tri točke smo razložili in opisali v prejšnji številki Preseka [5]. Za pravilno Sudoku rešitev manjkajo še dodatne omejitve:

V vsaki vrstici se vsako število od 1 do 9 pojavi natanko enkrat:

4. $\sum_{i=1}^{9} x_{k,(j-1)\cdot 9+i} = 1 \qquad j \in M, k \in M$

- Število k nam pove, za katero število preverjamo pogoj; $k \in M$, ker moramo preveriti za vsa števila.
- Število j nam pove, za katero vrstico preverjamo pogoj; $j \in M$, ker moramo preveriti za vse vrstice.
- Vsota, ki gre od $i = 1$ do $i = 9$, označuje sprehod čez celotno vrstico.



SLIKA 3. Prevozni model Sudoku uganke.

Primer. $k = 1$, $j = 3$: preverjamo, ali pogoj velja za število 1 za 3. vrstico. Vsota opiše sprehod čez celice 19, 20, 21, 22, 23, 24, 25, 26, 27 (kot vidimo na sliki 2, so to celice 3. vrstice).

V vsakem stolpcu se vsako število od 1 do 9 pojavi natanko enkrat:

$$5.\quad \sum_{i=1}^{9} x_{k,(i-1)\cdot 9+j} = 1 \qquad j \in M, k \in M$$

- Število k nam pove, za katero število preverjamo pogoj; $k \in M$, ker moramo preveriti za vsa števila.
- Število j nam pove, za kateri stolpec preverjamo pogoj; $j \in M$, ker moramo preveriti za vse stolpce.
- Vsota, ki gre od $i = 1$ do $i = 9$, označuje sprehod čez celotni stolpec.

Primer. $k = 1$, $j = 1$: preverjamo, ali pogoj velja za število 1 za 1. stolpec. Vsota opiše sprehod čez celice 1, 10, 19, 28, 37, 46, 55, 64, 73 (kot vidimo na sliki 2, so to celice 1. stolpca).

V vsakem od devetih označenih kvadratov se vsako število od 1 do 9 pojavi natanko enkrat:

$$6.\quad \sum_{i=1}^{3} \sum_{j=1}^{3} x_{k,(a-1)\cdot 27+(b-1)\cdot 3+(i-1)\cdot 9+j} = 1$$
$$k \in M, a \in R, b \in R$$

- Število k nam pove, za katero število preverjamo pogoj; $k \in M$, ker moramo preveriti za vsa števila.
- Števili a in b nam povesta, za kateri 3×3 kvadrat preverjamo pogoj; $a \in R$, $b \in R$, da tvorimo devet različnih parov (saj imamo devet različnih kvadratov).

4	2		1					9
		6				3	4	7
8				2	6			3
						6		
		1		5		8		
5							2	
3	7		9					1
	8		4	1				5

1	1	1	1	1	1	1	1	-1
0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1

4	2		1					9
		6				3	4	7
8				2	6			3
						6		
		1		5		8		
5							2	
3	7		9					1
	8		4	1				5

1	1	1	2	1	1	1	1	-1
0	0	0	1	0	0	1	1	1
0	0	0	1	0	0	1	1	1
0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	0	0	1
0	0	0	1	1	1	0	0	1
1	1	1	-1	1	1	1	1	2
0	0	0	1	1	1	0	0	1

SLIKA 4.
Primer matrike Π_9.

- Dvojna vsota označuje sprehod čez celoten kvadrat.

Primer. $k = 1$, $a = 3$, $b = 1$: preverjamo, ali pogoj velja za število 1 za 3×3 kvadrat v levem spodnjem kotu (glej sliko 1). Vsota opiše sprehod čez celice 55, 56, 57, 64, 65, 66, 73, 74, 75 (kot vidimo na sliki 2, so to celice kvadrata v levem spodnjem kotu).

Sedaj je možno rešiti Sudoku uganko tako, da rešimo problem prevoza, predstavljenega z izvornim vektorjem s, končnim vektorjem d in matriko cen c ter z dodatnimi omejitvami (kot je definirano zgoraj). To lahko naredimo s pomočjo jezika za modeliranje v matematični optimizaciji, kot je recimo AMPL.

Pokazali smo, da lahko Sudoku uganko oblikujemo kot poseben primer problema prevoza.

Primer reševanja

Poglejmo delovanje metode na primeru. Naj bo podana naslednja Sudoku uganka A:

$$A = \begin{array}{|c|c|c|c|c|c|c|c|c|}\hline 4 & 2 & & 1 & & & & & 9 \\ \hline & & 6 & & & & 3 & 4 & 7 \\ \hline & & & & & & & & \\ \hline 8 & & & & 2 & 6 & & & 3 \\ \hline & & & & & & 6 & & \\ \hline & & 1 & & 5 & & 8 & & \\ \hline 5 & & & & & & & 2 & \\ \hline 3 & 7 & & 9 & & & & & 1 \\ \hline & 8 & & 4 & 1 & & & & 5 \\ \hline \end{array}$$

Najprej za vsa števila od 1 do 9 tvorimo matrike Π_i. Za lažjo predstavo si poglejmo matriko za $i = 3$:

$$\Pi^{(3)} = \begin{array}{|c|c|c|c|c|c|c|c|c|}\hline \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{2} \\ \hline \mathbf{2} & \mathbf{1} & \mathbf{1} & 1 & 1 & 1 & -1 & 1 & 2 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \\ \hline 2 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & -1 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 1 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ \hline -1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 2 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ \hline \end{array}$$

Iz matrik Π_i, $i = 1, \ldots, 9$, sestavimo matriko cen c, velikosti 9×81, zapisano v tabeli 1.

Odebeljeno označen del matrike Π_3 vidimo na začetku tretje vrstice matrike c.

V jeziku AMPL ustvarimo datoteke .mod, .dat in .txt (pomen in sestavo teh smo pojasnili v prejšnji številki Preseka [5]).

Algorithm 1 AMPL model (.mod)

```
set M := {1..9};
set N := {1..81};
set R := {1..3};
param c {M,N};
param s {i in M};
param d {j in N};
var x {i in M,j in N} binary;
minimize Sudoku:
sum {i in M, j in N} c[i,j]*x[i,j];
subject to ponudbe {i in M}:
sum {j in N} x[i,j] = s[i];
subject to povpraševanje {j in N}:
sum {i in M} x[i,j] = d[j];
subject to vrstica {i in M, k in M}:
sum {n in M} x[k,(i-1)*9+n] = 1;
subject to kvadrat {i in R, j in R, k in M}:
sum {l in R, n in R} x[k,((i-1)*27+(j-1)*3+
    (l-1)*9+n)] = 1;
subject to stolpec {i in M, k in M}:
sum {n in M} x[k,(n-1)*9+i] = 1;
```

Algorithm 2 AMPL model (.dat)

```
param s := 1 9 2 9 3 9 4 9 5 9 6 9 7 9 8 9 9
9;
param d := 1 1 2 1 3 1 4 1 5 1 6 1 7 1 8 1 9
1 10 1 11 1 12 1 13 1 14 1 15 1 16 1 17 1 18
1 19 1 20 1 21 1 22 1 23 1 24 1 25 1 26 1 27
1 28 1 29 1 30 1 31 1 32 1 33 1 34 1 35 1 36
1 37 1 38 1 39 1 40 1 41 1 42 1 43 1 44 1 45
1 46 1 47 1 48 1 49 1 50 1 51 1 52 1 53 1 54
1 55 1 56 1 57 1 58 1 59 1 60 1 61 1 62 1 63
1 64 1 65 1 66 1 67 1 68 1 69 1 70 1 71 1 72
1 73 1 74 1 75 1 76 1 77 1 78 1 79 1 80 1 81
1;
param c:  # matrika cen c
```

Algorithm 3 AMPL model (.txt)

```
solve;
display x;
```

S pomočjo AMPL reševalnika kot rezultat dobimo matriko x, velikosti 9×81, zapisano v tabeli 2.

Iz nje moramo razbrati rešitev. Indeksi stolpcev označujejo številke celic (oštevilčenje celic je prikazano na sliki 2), indeksi vrstic pa števila od 1 do 9, ki jih vpisujemo v Sudoku mrežo. Če je v stolpcu j v vrstici i število 1, to pomeni, da imamo v celici številka j v Sudoku mreži število i. Kot vidimo, imamo v stolpcu 6 v vrstici 7 število 1. To pomeni, da se v celici številka 6 v Sudoku mreži nahaja število 7. Na ta način preberemo naslednjo rešitev Sudoku uganke A:

$A =$

4	2	3	1	6	7	5	8	9
1	5	6	2	9	8	3	4	7
7	9	8	5	3	4	2	1	6
8	4	5	7	2	6	1	9	3
9	3	7	8	4	1	6	5	2
2	6	1	3	5	9	8	7	4
5	1	4	6	7	3	9	2	8
3	7	2	9	8	5	4	6	1
6	8	9	4	1	2	7	3	5

Literatura

[1] M. Mansour, *Sudoku as a special transportation problem* (online), 28. 6. 2013. `http://arxiv.org/pdf/1210.2584v1.pdf`.

[2] J. Rosenhouse, L. Taalman, *Taking Sudoku Seriously: The Math Behind the World's Most Popular Pencil Puzzle*, Oxford University Press, New York, 2011.

[3] D. Green, Conceptis Sudoku difficulty levels explained (online), 28. 6. 2013. `http://www.conceptispuzzles.com/index.aspx?uri=info/article/2`.

[4] Wikipedia: Sudoku (online), 28. 6. 2013. `http://sl.wikipedia.org/wiki/Sudoku`.

[5] D. Božović in A. Taranenko, *Linearni problem prevoza*, Presek **41**, št. 2, 24–27.

$c =$

1	1	2	−1	2	1	1	1	2	0	0	1	· · ·	2	2	2
1	−1	1	1	2	1	1	2	1	1	1	1	· · ·	1	1	1
1	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**2**	**2**	**1**	**1**	· · ·	1	0	1
−1	1	1	2	1	1	2	2	2	2	2	2	· · ·	1	2	1
1	0	0	0	1	0	0	0	1	1	0	0	· · ·	1	1	−1
1	1	1	0	0	1	1	0	0	1	1	−1	· · ·	1	0	0
0	1	0	0	0	0	1	1	1	1	2	1	· · ·	0	0	1
1	1	0	0	0	0	1	0	0	1	1	0	· · ·	2	1	1
1	1	1	2	1	1	1	1	−1	0	0	0	· · ·	0	0	1

TABELA 1.

$x =$

0	0	0	1	0	0	0	0	0	1	0	0	· · ·	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	· · ·	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	· · ·	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	· · ·	0	0	0
0	0	0	0	0	0	1	0	0	0	1	0	· · ·	0	0	1
0	0	0	0	1	0	0	0	0	0	0	1	· · ·	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	· · ·	1	0	0
0	0	0	0	0	0	0	1	0	0	0	0	· · ·	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	· · ·	0	0	0

TABELA 2.

× × ×