Riešenie Sudoku v tabuľkových procesoroch

Štefan Peško*

Katedra matematických metód, Fakulta riadenia a informatiky, Žilinská univerzita v Žiline, Univerzitná 8215/1, 010 26 Žilina, Slovenská republika
Email: pesko@frcatel.fri.uniza.sk

Abstrakt. V príspevku sa chceme podeliť o našu skúsenosť s riešením obľúbeného puzzle *Sudoku* v tabuľkových procesoroch *Gnumeric* a *Excel* bez potreby ich procedurálneho programovania. Tento prístup našiel dobrú odozvu u študentov. Motivoval ich zamýšľať sa nad možnosťou modelovania optimalizačných úloh pomocou matematického programovania pri pohodlnom využití riešičov v procesoroch.

Kľúčové slová. Puzzle *Sudoku*, matematické programovanie, tabuľkové procesory

1. Úvod

Pri riešení viacerých praktických optimalizačných úloh operačnej analýzy, aplikovaných najmä v dopravnej logistike, sme zistili, že súčasné tabuľkové procesory *Gnumeric* [2] pod OS Linux a známejší *Excel* pod Windows ponúkajú možnosť ich pohodlného modelovania, podrobný výklad možno nájsť v prácach [3],[4] a [5]. Snažili sme sa ponúknuť získané skúsenosti študentom, no darilo sa nám zaujať nanajvýš dvoch – troch študentov v každom krúžku. Situácia sa výrazne zmenila, keď sa nám podarilo nájsť spôsob ako môžeme modelovať

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

Obr. 1: Zadanie ľahkého puzzle Sudoku

*Výzkum bol podporovaný grantovou agentúrou APVV v rámci projektu SK-CZ 09006

a efektívne riešiť puzzle *Sudoku* pomocou matematického programovania v tabuľkových procesoroch. Študenti konečne začali medzi sebou diskutovať a pýtať sa „Ako?“ a „Prečo?“.

2. Sudoku

Sudoku [1] je puzzle v hracom poli štvorcovej mriežky obyčajne 9×9 (na takú sa i my ďalej obmedzíme), ale niekedy aj 16×16 i väčšej. Je tiež známa ako hra „Number Place“, ktorá bola v Japonsku veľmi populárna práve pod názvom *Sudoku*, čo je skratka japonskej frázy v zmysle „čísla musia zostať nezávislé“. Podľa všetkého ju vymyslel Howard Garns v roku 1979. Dnes je obľúbená aj na Slovensku a hrá sa takto:

Do hracieho poľa 9×9 sa dopĺňajú čísla od 1 do 9. Každý z 9 boxov po 3×3 políčkach, každý riadok aj každý stĺpec musí každé z čísel $1 - 9$ obsahovať iba raz. Je dané nejaké prípustné rozmiestnenie čísel (zadanie puzzle), ktoré má jediné riešenie. Na obr. 1 máme príklad zadania takého *Sudoku* a na obr. 2 jeho riešenie, kde zvýraznené čísla boli zadané.

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

Obr. 2: Riešenie ľahkého Sudoku

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

Obr. 3: Čísla boxov v puzzle

My sa ďalej obmedzíme na hľadanie riešenia pomocou matematického programovania. Na formuláciu príslušných modelov budeme potrebovať nasledujúce označenie:

Nech matica $\mathrm{A} = (a_{ij})$ udáva počiatočné rozmiestnenia čísel $1, \ldots, 9$. Prvok $a_{ij} = 0$, ak políčko (i, j) nie je vyplnené. Nech je $\mathrm{B} = (b_{ij})$ matica, ktorej prvky udávajú číslo boxu políčka (i, j) ako vidieť z obrázku 3. Nech prvky matice $\mathrm{S} = (s_{ij})$ udávajú riešenie *Sudoku*, t. j. prípustné rozmiestnenie daných aj doplnených čísel tabuľky. V uvedených maticiach riadok a stĺpec zodpovedá príslušnému riadku a stĺpcu tabuľky.

3. Priraďovacia úloha s podmienkou v boxoch

Ak si pozorne všimneme len umiestnenie čísla 1 v tabuľke na obrázku 4 vidíme, že v políčkach $(4, 2)$ a $(9, 7)$ sú zadané **1** a ostatné 1 bolo treba doplniť tak, aby v každom riadku, stĺpci a boxe bola práve jedna jednotka. Podobne to platilo aj po voľbe iného čísla v riešení *Sudoku*. Toto pozorovanie nás vedie k formulácii priraďovacej úlohy s dodatočnou obmedzujúcou podmienkou jediného priradenia čísla v každom boxe.

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

Obr. 4: Umiestnenie 1 v puzzle

Ďalej pevne zvolíme nejaké číslo k a budeme sa ho snažiť prípustne umiestniť na voľné políčka tabuľky, pričom umiestnenia ostatných čísel ignorujeme. Poznamenajme, že voľnými políčkami teraz chápeme všetky políčka, kde ešte nie je umiestnené číslo k.

Položíme $c_{ij} := 1$ ak $a_{ij} = k$ inak $c_{ij} := 0$. Potom hodnota premennej $x_{ij} = 1$ znamená, že v políčku (i, j) je umiestnené číslo k. Treba riešiť nasledujúcu úlohu lineárneho bivalentného programovania – Assignment Problem with Boxes (APB1):

$$\sum_{i=1}^{9}\sum_{j=1}^{9} c_{ij}x_{ij} \to \max, \tag{1}$$

$$\sum_{j=1}^{9} x_{ij} = 1 \qquad i \in \{1, \dots, 9\}, \tag{2}$$

$$\sum_{i=1}^{9} x_{ij} = 1 \qquad j \in \{1, \dots, 9\}, \tag{3}$$

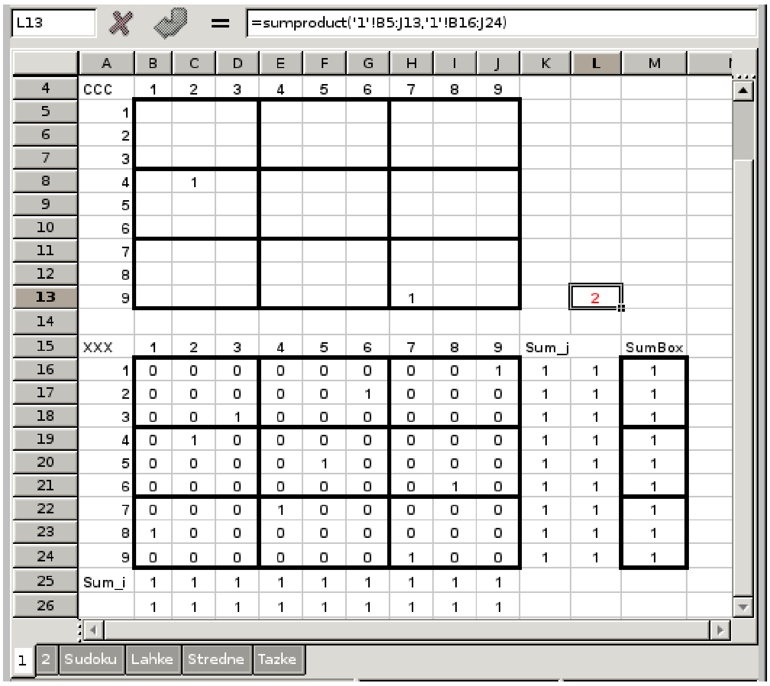
$$\sum_{i=1}^{9}\sum_{j=1 | b_{ij}=l}^{9} x_{ij} = 1 \qquad l \in \{1, \dots, 9\}, \tag{4}$$

$$x_{ij} \in \{0, 1\} \qquad i, j \in \{1, \dots, 9\}. \tag{5}$$

Maximálna hodnota cieľovej funkcie (1) nám pri korektnom zadaní zabezpečí, že čísla k budú umiestnené v požadovaných políčkach. V našom ilustračnom príklade na obr. 4 bolo zvolené číslo $k = 1$ a hodnota cieľovej funkcie bude rovná 2, čo zodpovedá jednému z riešení s $x_{4,2} = x_{9,7} = 1$, nakoľko do políčok $(4, 2)$ a $(9, 7)$ musíme umiestniť **1**.

Riadkové (2) a stĺpcové (3) súčty zabezpečia jediné umiestnie čísla v riadkoch a stĺpcoch. Boxové súčty (4) zaručia, že aj v každom boxe bude umiestnené práve jedno číslo. Ale tieto podmienky platia zmysluplne len vtedy, ak je splnená bivalentná obligátorná podmienka (5). Poznamenajme, že úloha (1),(2),(3),(5) je formuláciou maximalizačnej priraďovacej úlohy, kde dokonca može byť podmienka (5) oslabená požiadavkou len nezápornosti premenných a potom riešená ako úloha lineárneho programovania (LP).

Teraz si ukážeme ako možeme pohodlne riešiť úlohu APB1 v tabuľkových procesoroch *Gnumeric* resp. *Excel.* V liste „1“ zošitu Sudoku2.gnumeric resp.


L13 =sumproduct('1'!B5:J13,'1'!B16:J24)

	A	B	C	D	E	F	G	H	I	J	K	L	M
4	CCC	1	2	3	4	5	6	7	8	9			
5	1												
6	2												
7	3												
8	4		1										
9	5												
10	6												
11	7												
12	8												
13	9							1				2	
14													
15	XXX	1	2	3	4	5	6	7	8	9	Sum_j		SumBox
16	1	0	0	0	0	0	0	0	0	1	1	1	1
17	2	0	0	0	0	0	1	0	0	0	1	1	1
18	3	0	0	1	0	0	0	0	0	0	1	1	1
19	4	0	1	0	0	0	0	0	0	0	1	1	1
20	5	0	0	0	0	1	0	0	0	0	1	1	1
21	6	0	0	0	0	0	0	0	1	0	1	1	1
22	7	0	0	0	1	0	0	0	0	0	1	1	1
23	8	1	0	0	0	0	0	0	0	0	1	1	1
24	9	0	0	0	0	0	0	1	0	0	1	1	1
25	Sum_i	1	1	1	1	1	1	1	1	1			
26		1	1	1	1	1	1	1	1	1			

Obr. 5: Riešenie úlohy APB1

Sudoku3.xls na obr. 5 označíme oblasť – maticu $CCC = B5 : J13$ [1]. Zadanie puzzle je v ňom určené hodnotami 1 v bunkách $CCC[4,2] = C8$ a $CCC[9,7] = H13$, ostatné nevyplnené políčka procesor interpretuje ako 0. Podobne definujeme maticu $XXX = B16 : J24$, v ktorej budeme hľadať riešenie puzzle. Pre lepšiu orientáciu v oblastiach CCC aj XXX sú bloky puzzle orámované.

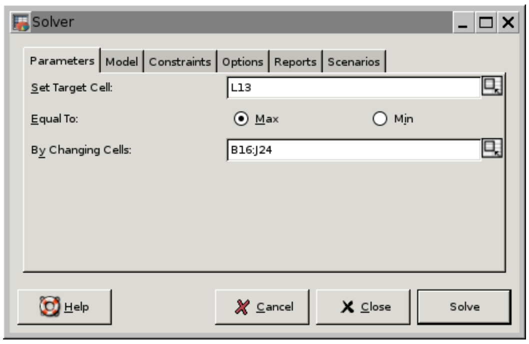
V bunke $L13 := sumproduct(B5 : J13, B16 : J24)$ definujeme cieľovú funkciu (1) ako skalárny súčin matíc CCC a XXX. V oblasti $Sum_i = B25 : J25$ definujeme ľavú stranu podmienky (2) tak, že dáme do bunky $B25$ príslušný súčet, t. j. $B25 := sum(B16 : B24)$ a potom ju rozkopírujeme do oblasti Sum_i v ktorej tak dostaneme aj ostatné stĺpcové súčty matice XXX. Podobne v oblasti

[1] Pri definovaní oblastí (tabuliek) budeme ďalej používať znak = na rozdiel od znaku := pre obsah ich buniek (čísla, reťazce, tabuľkové formule).

$Sum_j = K16 : K24$ definujeme ľavú stranu podmienky (3) riadkovými súčtami matice XXX . Oblasť $B25 : J25$ aj oblasť $L16 : L24$ obsahujú 1 a budú použité ako pravé strany podmienok (2) a (3).

Ešte nám zostáva definovať oblasť $SumBox = M16 : M24$ súčtov v boxoch matice XXX, ktorá definuje ľavú stranu podmienky (4). Tu musíme postupovať pomalšie. Najskôr položíme $M16 := sum(B16 : D18)$, $M17 := sum(E16 : G18)$ a $M18 := sum(H16 : J18)$, čím definujeme súčty v boxoch $1, 2, 3$ matice. Potom oblasť $M16 : M18$ skopírujeme do oblasti $M19 : M21$ a $M22 : M24$, čím definujeme súčty v boxoch $4, 5, 6$ a boxoch $7, 8, 9$.

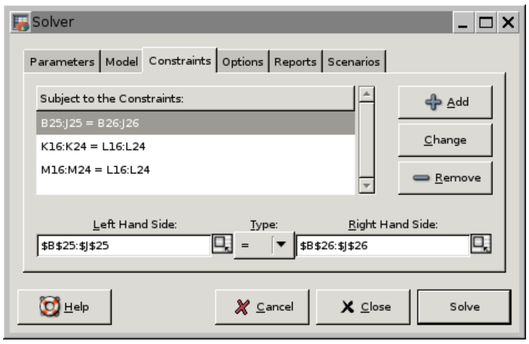
Ak vynulujeme všetky prvky matice XXX, potom vidíme, že sa nám vynulovali aj oblasti Sum_i, Sum_j a $SumBox$. Našim cieľom je však aby tieto oblasti boli jednotkové. Môžeme to spraviť buď hádaním hodnôt $0, 1$ v XXX, alebo pomocou solveru (riešiča) pre úlohu celočíselného lineárneho programovania.



Obr. 6: Parametre solveru v Gnumericu pre úlohu APB1

Na obrázku 6 máme definované parametre solveru v Gnumericu odkazmi: na bunku $L13$, v ktorej je cieľová fukcia, cieľ optimalizácie Max a meniace sa bunky matice XXX – premenné modelu. Z obrázku 7 vidíme, že štrukturálne obmedzujúce podmienky (2), (3), (4) sa zadávajú v tvare:

$$B25 : J25 = B26 : J26,$$
$$K16 : K24 = L16 : L24,$$
$$M16 : M24 = L16 : L24.$$



Obr. 7: Obmedzujúce podmienky APB1

Potom už len stačí stlačiť voľbu Solve a čakať na výsledok.

Riešenie je v matici XXX, na obrázku 5. Vidíme, že rešpektuje všetky obmedzujúci podmienky, t. j. jediný výskyt 1 v každom riadku stĺpci i bloku, pričom zohľadňuje zadanie puzzle. Je zaujímavé, že ak nahradíme (5) slabšou podmienkou (relaxáciou) nezápornosti premenných, potom opäť dostávame bivalentné riešenie.

Tu vzniká priestor na zvedavé otázky: „Môžeme vždy relaxovať podmienku bivalentnosti premenných?“, „Môžeme vhodnou úpravou ocenenia priradenia zaručiť, aby niektoré políčka tabuľky neboli obsadené?“.

4. Dve koordinované priraďovacie úlohy s podmienkou v boxoch

Vráťme sa k obrázku 2 a budeme si všímať len umiestnenia čísel 1 a 2. Ako vidíme, jedná sa o dvojicu priraďovacích úloh s podmienkou v boxoch (pre každé číslo jedna úloha), ktoré sú naviac koordinované požiadavkou, aby sa na žiadnom políčku čísla neprekrývali. Jedno z prípustných riešení je na obrázku 8.

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

Obr. 8: Umiestnenie čísel 1 a 2 v puzzle

Pretože umiestňujeme čísla 1 a 2, potrebujeme toto označenie: Nech $c_{ij1} := 1$ ak $a_{ij} = 1$ inak 0 a $c_{ij2} := 1$ ak $a_{ij} = 2$ inak 0. Potom budeme očakávať hodnotu premennej $x_{ijk} = 1$, ak bude v políčku (i, j) číslo $k \in \{1, 2\}$.

Treba teda riešiť úlohu lineárneho bivalentného programovania – 2 Assignment Problems with Boxes (APB2):

$$\sum_{i=1}^{9}\sum_{j=1}^{9}\sum_{k=1}^{2} c_{ijk}x_{ijk} \to \max, \tag{6}$$

$$\sum_{j=1}^{9} x_{ijk} = 1 \qquad i \in \{1, \dots, 9\}, k \in \{1, 2\}, \tag{7}$$

$$\sum_{i=1}^{9} x_{ijk} = 1 \qquad j \in \{1, \dots, 9\}, k \in \{1, 2\}, \tag{8}$$

$$\sum_{i=1}^{9}\sum_{j=1|b_{ij}=l}^{9} x_{ijk} = 1 \qquad l \in \{1, \dots, 9\}, k \in \{1, 2\}, \tag{9}$$

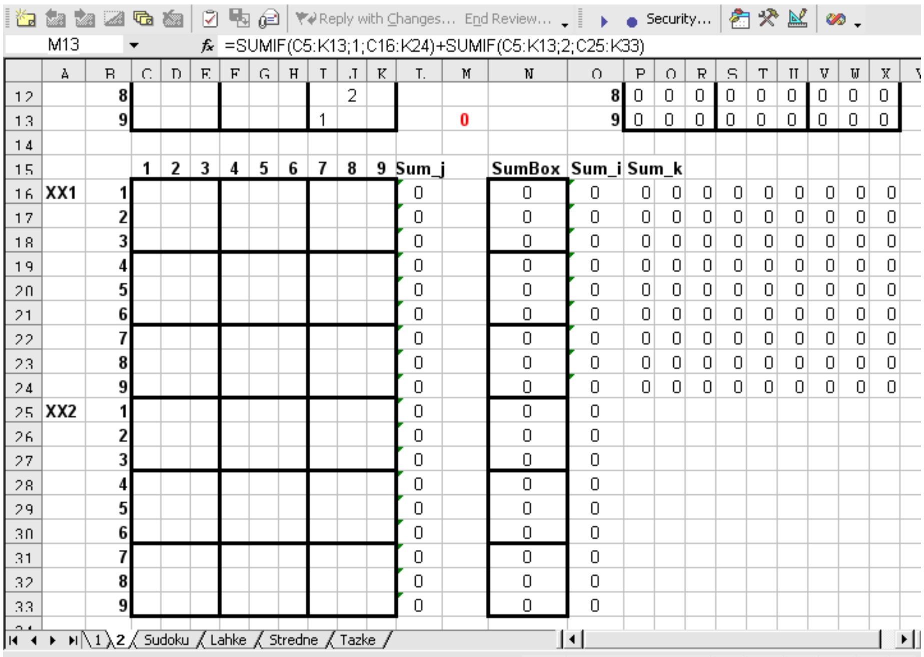
$$x_{ij1} + x_{ij2} \le 1 \qquad i, j \in \{1, \dots, 9\}, \tag{10}$$

$$x_{ijk} \in \{0, 1\} \qquad i, j \in \{1, \dots, 9\}, k \in \{1, 2\}. \tag{11}$$

Prípustné riešenie je pre každé fixované číslo $k \in \{1, 2\}$ formulované samostatnou úlohou APB1 v tvare (6)–(9), (11). Až koordinačná podmienka (10) zabezpečí, aby na jedno políčko (i, j) bolo umiestnené nanajvýš jedno z čísel $1, 2$.

V liste „2“ zošitu Sudoku2.gnumeric, resp. Sudoku3.xls na obr. 9 máme opäť označenú oblasť $CCC = C5 : K13$, z ktorej vidíme len posledné dva riadky. Teraz pre zmenu uprednostníme zošit *Excelu*. Zadanie puzzle z obr. 8 je v ňom definované hodnotami $1, 2$ v bunkách $CCC[4, 2] = CCC[9, 7] := 1$ a $CCC[4, 1] =$

$CCC[8,8] := 2$. Oblasť premenných je určená súvislou oblasťou, ktorá obsahuje matice $XX1 = C16 : K24$ pre čísla 1 a $XX2 = C25 : K33$ pre čísla 2.



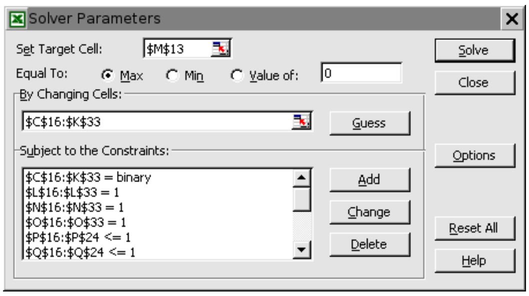
Obr. 9: Oblasti riešenia a podmienok úlohy APB2

Cieľovú funkciu (6) definujeme v bunke $M13$ ako súčet dvoch podmienených súčtov formulou

$$M13 := sumif(C5 : K13, 1, C16 : K24) + sumif(C5 : K13, 2, C25 : K33),$$

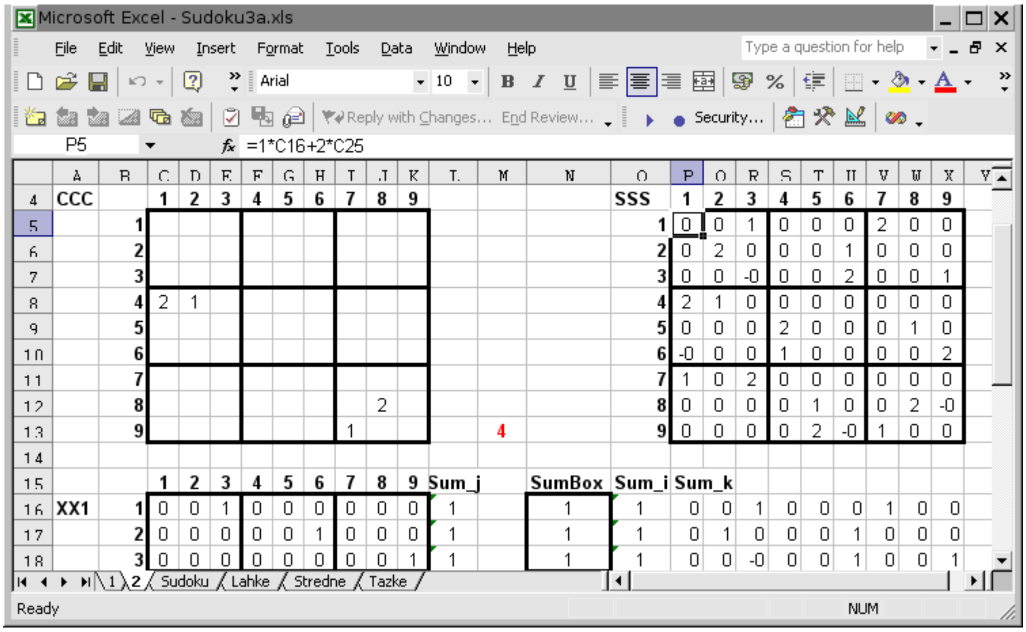
čo je v našom príklade rovné $XX1[4,2] + XX1[9,7] + XX2[4,1] + XX2[8,8]$.

Obsah oblastí $Sum_i = O16 : O33$, $Sum_j = L16 : L33$ a $SumBox = N16 : N33$ sa definuje analogicky ako v zošite „1“ súčtami v podmienkach (7), (8) a (9). Koordinačná podmienka (10) sa tiež realizuje jednoducho. Stačí položiť $P16 := C15 + C25$ a potom bunku $P16$ rozkopírovať do oblasti $Sum_k = P16 : X24$.



Obr. 10: Parametre solveru úlohy APB2

Na obrázku 10 máme definované parametre solveru v *Exceli* odkazmi: na bunku $M13$, v ktorej je cieľová funkcia, cieľ optimalizácie Max a meniace sa bunky oblasti $C16 : K33$. V okne s obmedzujúcimi podmienkami vidíme, že na rozdiel od podobného okna v Gnumericu potrebujeme explicitne požadovať bivaletnosť premenných v tvare $\$C\$16 : \$K\$33 = binary$. Na druhej strane v ďalších obmedzujúcich podmienkach môžeme ich pravé strany zadávať pohodlne jediným číslom 1 miesto zodpovedajúcej oblasti jednotiek.



Obr. 11: Riešenia úlohy APB2

Po stlačení voľby $\boxed{\text{Solve}}$ v solveri na obr. 10 dostaneme riešenie puzzle na obr. 11, v matici $SSS = P5 : X13$. Jej prvky získame z matíc $XX1, XX2$ tak, že najskôr definujeme bunku $P5 := 1 * C16 + 2 * C26$, a potom ju rozkopírujeme do oblasti matice SSS. V matici SSS sme dostali iné riešenie, než na obr. 8, čo nás neprekvapí, ak si uvedomíme, že toto riešenie nezohľadňuje požiadavky na umiestnenie ďalších čísel $3 - 9$.

Poznamenajme, že ak by matica CCC obsahovala iné čísla $\{a, b\}$ než $\{1, 2\}$ stačí ich príslušne nahradiť v bunkách $M13$ a SSS. Napríklad tak, že matica $XX1$ bude rozhodovať o umiestnení čísla a a matica $XX2$ o čísle b.

Tu opäť môžeme pokračovať so zvedavými otázkami: „Môžeme aj v tomto modeli relaxovať podmienku bivalentnosti premenných?“ alebo „Oceňme naviac číslom $-k$ políčka tabuľky, ktoré sme už obsadili pri riešení *Sudoku* číslom k, ale $k \notin \{a, b\}$. Povedie postup, ktorý postupne priradí dvojice $(1, 2), (3, 4), (5, 6), (7, 8)$ a nakoniec mu ostanú voľné políčka pre číslo 9 vždy k hľadanému riešeniu puzzle?“.

5. 9 koordinovaných priraďovacích úloh s podmienkou v boxoch

Zovšeobecnenie problému APB2 na model 9-tich koordinovaných priraďovacích úloh s podmienkou v boxoch, ktorý už rieši celé puzzle *Sudoku*, je teraz jasný.

Nech je hodnota premennej $x_{ijk} = 1$, ak je číslo k umiestnené v políčku (i, j) hracieho poľa, a nech $c_{ijk} := 1$ ak $a_{ij} = k$ inak 0. Potom stačí riešiť úlohu lineárneho bivalentného programovania – 9 Assignmet Problems with Boxes (APB9):

$$\sum_{i=1}^{9}\sum_{j=1}^{9}\sum_{k=1}^{9} c_{ijk}x_{ijk} \to \max, \tag{12}$$

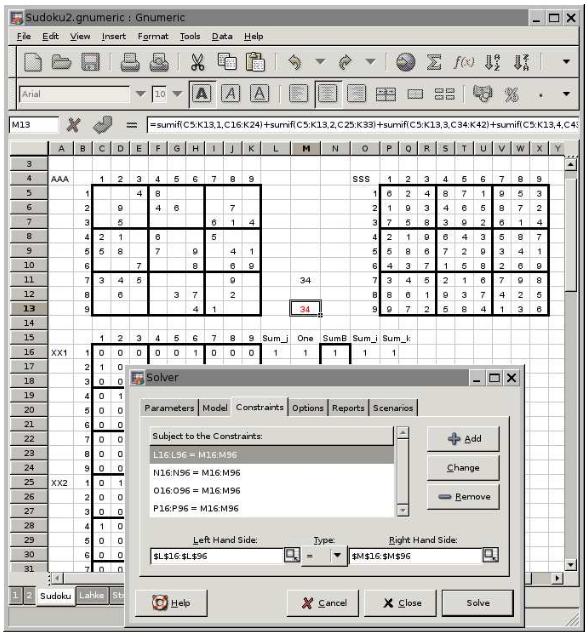
$$\sum_{j=1}^{9} x_{ijk} = 1 \qquad i, k \in \{1, \dots, 9\}, \tag{13}$$

$$\sum_{i=1}^{9} x_{ijk} = 1 \qquad j, k \in \{1, \dots, 9\}, \tag{14}$$

$$\sum_{i=1}^{9}\sum_{j=1|b_{ij}=l}^{9} x_{ijk} = 1 \qquad k, l \in \{1, \dots, 9\}, \tag{15}$$

$$\sum_{k=1}^{9} x_{ijk} = 1 \qquad i, j \in \{1, \dots, 9\}, \tag{16}$$

$$x_{ijk} \in \{0, 1\} \qquad i, j, k \in \{1, \dots, 9\}. \tag{17}$$



Obr. 12: Riešenia Sudoku ako úlohy APB9

Prípustné riešenie je pre každé číslo k formulované samostatnou úlohou v tvare (12)-(15), (17). Koordinačná podmienka (16) nám zabezpečí, aby na jedno políčko (i, j) bolo umiestnené práve jedno z čísel $1, \ldots, 9$.

Teraz sa vrátime ku *Gnumericu* – *Excel* totiž neumožňuje riešiť úlohy s $9 \times 9 \times 9$ premennými ako požaduje úloha APB9. V liste „Sudoku" zošitu Sudoku2.gnumeric máme na obr. 12 označenú oblasť $AAA = C5 : K13$. Oblasť premenných je tu určená súvislou oblasťou, ktorá obsahuje deväť matíc $XX1 = C16 : K24$, $XX2 = C25 : K33, \ldots, XX9 = C88 : K96$.

Cieľovú funkciu (12) definujeme v bunke $M13$ ako súčet deviatich podmienených súčtov formulou

$$M13 := sumif(AAA, 1, XX1) + sumif(AAA, 2, XX2) + \cdots + sumif(AAA, 9, XX9).$$

Obsah oblastí $Sum_i = O14 : O96$, $Sum_j = L14 : L96$ a $SumBox = N14 : N96$, $Sum_k = L14 : L96$ sa definujú analogicky ako v zošite „1" súčtami premenných na ľavých stranách podmienok (13), (14), (15) a (16). V okne $\boxed{\text{Constraint}}$ na obr. 12 vidíme ich zodpovedajúce maticové realizácie. Riešenie *Sudoku* potom nájdeme v matici $SSS = P5 : X13$, ktorú definujeme rozkopírovaním bunky $P5$ určenej $P5 := C16 + 2*C25 + 3*C34 + 4*C43 + 5*C52 + 6*C61 + 7*C70 + 8*C79 + 9*C88$ do celej oblasti SSS.

6. Záver

Riešenie puzzle *Sudoku* pomocou matematického programovania ukázalo možnosť, ako hravou formou presvedčiť študentov o užitočnosti modelovania a riešenia niektorých optimalizačných úloh. Pohodlným ladením modelov pomocou solverov v rôznych tabuľkových procesoroch, keď sa po krátkom čase stačí obmedziť na formulácie úloh, získavajú študenti cennú skúsenosť aj s kvalitou ich solverov. Tento prístup tiež umožňuje demonštrovať potrebu návrhu heuristík v prípade riešenia úloh, ktoré nie sú riešiteľné dostupnými solvermi – *Excel* verzus *Gnumeric*.

Literatúra

1. Davis, T.: *The Matematics of Sudoku*, Preliminary, (2006) http://www.geometer.org/mathcircles/sudoku.pdf
2. *The Gnumeric Manual*, http://www.gnome.org/projects/gnumeric/doc/gnumeric.shtml
3. Peško, Š.: *Vybrané modely logistiky v EXCELi*, Učebné texty k cvičeniam, (2002) http://frcatel.fri.utc.sk/~pesko/volk.zip
4. Peško, Š.: *Pohodlná optimalizácia reálnych úloh v tabuľkových procesoroch*, Slovak Society for Operations Research, 7th international seminar, APPLICATION OF QUANTITATIVE METHODS IN RESEARCH AND PRACTICE, pp. 29 - 35, Remata, ISBN 80-225-2079-9, (2005)
5. Peško, Š.: *Využitie tabuľkového procesoru Gnumeric vo výučbe a výskume*, In: 6^{th} International Conference, Aplimat, Part IV, Open Source Software in Research and Education, feb. 6-9, Bratislava, pp. 41 - 52, ISBN 978-80-969562-7-2, (2007)