Kaj imata skupnega mačja hrana in sudoku?

Uroš Hekić, *Gimnazija Ptuj*
Ana Reberc, *ll. gimnazija Maribor*
Jasna Urbančič, *Gimnazija Vič, Ljubljana*
David Gajser (mentor), *Fakulteta za matematiko in fiziko, Ljubljana*

POVZETEK

Naša skupina je najprej poskrbela za pripravo uravnoteženega mačjega obroka, nato pa poiskala način za avtomatsko reševanje sudokuja. Namesto lastnikov mačk lahko sudoku sedaj reši računalnik, mačke pa tako dobijo vso potrebno pozornost. Mijav.

Uvod

Linearno programiranje, poznano tudi kot linearna optimizacija, je problem maksimiziranja oziroma minimiziranja linearne "kriterijske" funkcije. Začetki reševanja tega problema segajo že v drugo polovico 18. stoletja, ko se je Joseph Fourier ukvarjal z reševanjem sistema linearnih neenačb. Po njem je poimenovana tudi Fourier-Motzkinova eliminacija, znana tudi kot metoda FME. Leta 1939 je Leonid Kantorovich formuliral prve probleme linearnega programiranja, ki so jih reševali v vojne namene med drugo svetovno vojno. Metoda reševanja je ostala skrivnost vse do leta 1947, ko je George B. Dantzig izdal metodo simpleksov, John von Neumann pa razvil teorijo dualnosti in jo apliciral na področje teorije iger. Po vojni se je linearna optimizacija uporabljala v številnih panogah industrije.

Linearno programiranje se pogosto uporablja v mikroekonomiji in pri upravljanju podjetij na področjih, kot so načrtovanje, proizvodnja in transport. Večina podjetij bi rada maksimizirala dobiček ali minimizirala stroške z omejenimi viri. Mi smo se s takim problemom srečali pri mešanju sestavin za mačjo hrano.

Oblika linearnega programa

Naj bosta m in n naravni števili, kjer n predstavlja število spremenljivk, ki jih označimo z $x_1,\ x_2 \dots x_n$, m pa število omejitev. Poleg m in n imamo še $n+m+mn$ realnih števil, ki jih navadno označimo:

- $c_1,\ c_2 \dots c_n$,
- $b_1,\ b_2 \dots b_m$,
- $a_{1,1},\ a_{1,2} \dots a_{1,n}$,
  $a_{2,1},\ a_{2,2} \dots a_{2,n}$,
  $\dots$
  $a_{m,1},\ a_{m,2} \dots a_{m,n}$.

Linearni program je sestavljen iz kriterijske funkcije in omejitev. *Kriterijska funkcija* je oblike

$$\min:\ c_1x_1 + \ c_2x_2 + \cdots + c_nx_n,$$

namesto min lahko uporabimo tudi max (odvisno od tega, ali želimo kriterijsko funkcijo minimizirati ali maksimizirati). *Omejitve* so naslednje oblike, kjer $\square$ označuje $\geq$, $\leq$ ali $=$:

$$\begin{aligned} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n \ &\square\ b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n \ &\square\ b_2 \\ &\dots \\ a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{m,n}x_n \ &\square\ b_m. \end{aligned}$$

Če gre za *celoštevilski linearni program*, med omejitve dodamo še

$$x_1,\ x_2 \dots x_n \in \mathbb{Z}.$$

Rešitev [celoštevilskega] linearnega programa je taka n-terica $(x_1^*, x_2^* \dots x_n^*)$, za katero veljajo vse omejitve in pri kateri kriterijska funkcija doseže min (oziroma max).

Mešanje Mačje hrane

Slika 1: Mucki po našem optimalnem obroku

Na pločevinki mačje hrane piše, da je v 100g hrane toliko gramov posameznih snovi:

beljakovine	maščobe	vlaknine	sol
≥ 8	≥ 6	≤ 2	$\leq 0,4$

Mačjo hrano želimo proizvajati po kar najnižji ceni. Uporabljamo naslednje sestavine (podatki so za gram sestavine):

sestavina	cena v evrih	beljakovine	maščobe	vlaknine	sol
piščanec	0,013	0,100	0,080	0,001	0,002
govedina	0,008	0,200	0,100	0,005	0,005
ovčetina	0,010	0,150	0,110	0,003	0,007
riž	0,002	0,000	0,010	0,100	0,002
pšenični otrobi	0,005	0,040	0,010	0,150	0,008
gel	0,001	0,000	0,000	0,000	0,000

Rešitev: Linearni program nastavimo tako, da bo imel šest spremenljivk, $x_1, x_2, x_3, x_4, x_5, x_6$, ki predstavljajo odstotke določene sestavine v končnem izdelku. Konstante b_1, b_2, b_3, b_4 predstavljajo zgornje oziroma spodnje meje za vsebnost posamezne snovi, $c_1, c_2, c_3, c_4, c_5, c_6$ predstavljajo ceno sestavin v evrih za en gram, $a_{1,1} \dots a_{6,4}$ pa predstavljajo vsebnost posamezne snovi v sestavini. Kriterijska funkcija za ta primer je torej

$$\min: \ x_1 \cdot 0,013 + x_2 \cdot 0,008 + x_3 \cdot 0,010 + x_4 \cdot 0,002 + x_5 \cdot 0,005 + x_6 \cdot 0,001,$$

omejitve pa so

$$x_1 \cdot 0,100 + x_2 \cdot 0,200 + x_3 \cdot 0,150 + x_4 \cdot 0,000 + x_5 \cdot 0,040 + x_6 \cdot 0,000 \geq 8$$
$$x_1 \cdot 0,080 + x_2 \cdot 0,100 + x_3 \cdot 0,110 + x_4 \cdot 0,010 + x_5 \cdot 0,010 + x_6 \cdot 0,000 \geq 6$$
$$x_1 \cdot 0,001 + x_2 \cdot 0,005 + x_3 \cdot 0,004 + x_4 \cdot 0,100 + x_5 \cdot 0,150 + x_6 \cdot 0,000 \leq 2$$
$$x_1 \cdot 0,002 + x_2 \cdot 0,005 + x_3 \cdot 0,007 + x_4 \cdot 0,002 + x_5 \cdot 0,008 + x_6 \cdot 0,000 \leq 0,4$$

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 100$$
$$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0.$$

Reševanje linearnega programa v Pythonu

Podatke in pogoje smo vnesli v Python in s pomočjo knjižnice *PuLP* rešili ta linearni program. Ugotovili smo, da je najbolje hrano sestaviti iz 60% govedine in 40% gela.

Reševanje sudokuja

Sudoku je japonska miselna igra, pri kateri mora reševalec izpolniti polja kvadrata velikosti 9×9 s številkami od 1 do 9, pri čemer se v vsaki vrstici, stolpcu in manjšem kvadratu velikosti 3×3 le enkrat pojavi vsaka številka od 1 do 9.

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

Slika 2: Primer sudokuja

Rešitev: Linearni program ima $9^3 = 729$ spremenljivk oblike $x_{i,j,k}$, kjer urejeni par (i,j) predstavlja koordinate polja v tabeli, k pa število na tem polju v tabeli:

$$x_{i,j,k} = \begin{cases} 1; & \text{če je } k \text{ na polju } (i,j) \\ 0; & \text{če } k \text{ ni na polju } (i,j) \end{cases} .$$

Kriterijska funkcija v tem primeru nima pravega pomena, zato je lahko karkoli. Želimo le zadostiti naslednjim pogojem:

$$\begin{aligned}
x_{i,j,k} &\in \mathbb{Z} && \forall\, i,j,k \in \{1,2\ldots 9\} \\
0 \le x_{i,j,k} &\le 1 && \forall\, i,j,k \in \{1,2\ldots 9\} \text{ (Ta pogoj nam skupaj s prejšnjim da } x_{i,j,k} \in \{0,1\}\text{).} \\
\sum_{k=1}^{9} x_{i,j,k} &= 1 && \forall\, i,j \in \{1,2\ldots 9\} \text{ (V vsakem polju je natanko eno število.)} \\
\sum_{i=1}^{9} x_{i,j,k} &= 1 && \forall\, j,k \in \{1,2\ldots 9\} \text{ (V vsakem stolpcu vsako število nastopa natanko enkrat.)} \\
\sum_{j=1}^{9} x_{i,j,k} &= 1 && \forall\, i,k \in \{1,2\ldots 9\} \text{ (V vsaki vrstici vsako število nastopa natanko enkrat.)} \\
\sum_{i=1}^{3}\sum_{j=1}^{3} x_{i+z_1,j+z_2,k} &= 1 && \forall\, k \in \{1,2\ldots 9\} \wedge z_1,z_2 \in \{0,3,6\},
\end{aligned}$$

kjer zadnji pogoj pove, da v vsakem kvadratu velikosti 3×3 nastopa vsako število natanko enkrat.

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

Slika 3: Rešitev sudokuja iz slike 2

Reševanje celoštevilskega linearnega programa v Pythonu

Podatke in pogoje smo vnesli v Python in s pomočjo knjižnice *PuLP* rešili sudoku. Rešitev je na sliki 3.

Zaključek

Predstavili smo primera linearnih programov, a reševanja nismo opisali, ker smo ga prepustili računalniku. Velika razlika med celoštevilskimi linearnimi programi in linearnimi programi je v tem, da so znane metode za reševanje linearnih programov, s katerimi lahko poljuben linearni program z računalnikom "hitro" rešimo. Za celoštevilske linearne programe pa kljub navidezni enostavnosti take metode niso znane. Če bi našli postopek, s katerim bi lahko "hitro" rešili poljuben celoštevilski linearni program, bi rešili enega izmed "Millenium prize problems" in tako zaslužili milijon dolarjev.

Viri

[1] *Linear Programming*, `mathworld.wolfram.com/LinearProgramming.html`, citirano dne 21. 8. 2012.

[2] *A blending problem*, `http://www.coin-or.org/PuLP/CaseStudies/a_blending_problem.html`, citirano dne 21. 8. 2012.

[3] *Sudoku problem*, `http://www.coin-or.org/PuLP/CaseStudies/a_sudoku_problem.html`, citirano dne 21. 8. 2012.

[4] *Sudoku*, `http://www.brainbashers.com/showsudoku.asp`, citirano dne 21. 8. 2012.

[5] *Latex Mathematics*, `http://en.wikibooks.org/wiki/LaTeX/Mathematics`, citirano dne 21. 8. 2012.

[6] *Linear programming*, `http://en.wikipedia.org/wiki/Linear_programming`, citirano dne 21. 8. 2012.

[7] *Od mucka do mačke*, `http://whiskas.si/mucki/o-muckih/od-mucka-do-macke/`, citirano dne 21. 8. 2012.