6th International Conference
APLIMAT 2007
Faculty of Mechanical Engineering - Slovak University of Technology in Bratislava
Session: Open Source Software in Research and Education

VYUŽITIE TABUĽKOVÉHO PROCESORU GNUMERIC VO VÝUČBE A VÝSKUME

PEŠKO, Štefan, (SK)

Abstrakt. V príspevku sa chceme podeliť o naše skúsenosti s pomerne málo využívanou možnosťou pohodlného riešenia niektorých optimalizačných úloh v tabuľkovom procesore *Gnumeric* bez potreby ich procedurálneho programovania. Tento prístup našiel uplatnenie vo výskume pri tvorbe základných modelov praktických logistických problémov, aj prekvapujúco dobrú odozvu u študentov pri precvičovaní poznatkov z predmetov operačnej analýzy.

Kľúčové slová. Tabuľkový procesor, neprocedurálne programovanie, operačná analýza.

USING GNUMERIC SPREADSHEET IN EDUCATION AND RESEARCH

Abstract. We like to share our experience with not so widely used possibility of very convenient solution of some optimisation problems in spreadsheet *Gnumeric* without procedural programming. This approach can be used in the research for the creation of the basic logistic models. We used it also in teaching of operations research with surprisingly good feedback from students.

Key words and phrases. Spreadsheet, nonprocedural programming, operations research.

Mathematics Subject Classification. Primary 97D40, 97C70; Secondary 90B06.

1 Úvod

Pri riešení viacerých praktických optimalizačných úloh operačnej analýzy, aplikovaných najmä v dopravnej logistike, sme zistili [5], že i súčasný tabuľkový procesor Gnumeric [3] pod OS GNU/Linux ponúka možnosť ich pohodlného riešenia. Z ponuky Gnumericu sa totiž stačí obmedziť len na niektoré jeho základné funkcie, ktoré umožňujú jeho neprocedurálne programovanie a na *Riešiteľ (Solver)* na riešenie úloh lineárneho, resp. celočíselného lineárneho programovania.

2 Od Excelu ku Gnumericu

Aj keď má väčšina študentov skúsenosti s nejakým tabuľkovým procesorom, najčastejšie s Excelom od Microsoftu, nezaškodilo im zopakovať niekoľko pravidiel o práci s *bunkami* tabuľky. Oplatí sa hneď na začiatku sústrediť hlavne na *absolútne* ($\$A\$3$), *relatívne* ($A3$) *a zmiešané odkazy* ($\$A3, A\$3$) buniek. Ich zmysel sa dá rukolapne pochopiť pri kopírovaní oblastí.[4]

Ďalej sa nám osvedčilo prejsť na tabuľkové funkcie – hlavne maticové, ktoré výrazne sprehladňujú tvorbu modelov, budeme potrebovať tieto:

- $index(A;i[;j]^5)$ vyberá z oblasti (matice) A obsah v riadku i a stĺpci j ako prvok A_{ij},
- $mmult(A;B)$ vracia maticu $A \cdot B$ rovnú maticovému súčinu matíc A a B,
- $sumproduct(A;B)$ vracia číslo $A \odot B$ rovné skalárnemu súčinu matíc $\sum_i \sum_j A_{ij}B_{ij}$.

Kľúčovým, aj keď nie jediným nástrojom modelovania, je tu v ponuke voľba *Riešiteľ* (obrázok 1), ktorá optimalizuje za nás riešiac úlohy lineárneho (celočíselného) programovania v nasledujúcom tvare (M)LP:

$$\sum_{j=1}^{n} c_j x_j \to \min\ [\max], \tag{1}$$

$$\sum_{j=1}^{n} a_{ij} x_j \leqq \left[\ \geqq,\ =\ \right]\ b_i, \quad i = 1, \ldots, m, \tag{2}$$

$$x_j \geqq 0,\ \left[\text{celé}, \{0,1\}\right], \qquad j = 1, \ldots, n. \tag{3}$$

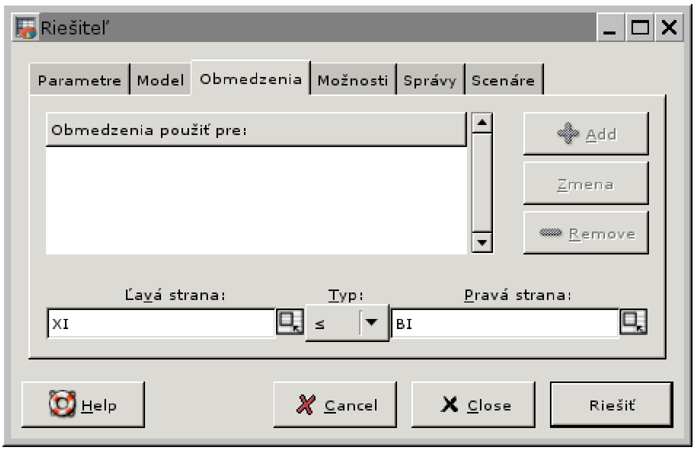
pričom umožňuje zadávať v bunkách požadované lineárne funkcie. A tak cieľovú funkciu (1) zapíšeme v tvare formuly $= sumproduct(C;X)$, kde C, X sú príslušné vektory reprezentované súvislými oblasťami buniek. K pohodliu prispieva možnosť vektorového zápisu obmedzení (2) zhodného typu t. j. ak pre $i \in \{m_1, \ldots, m_2\}$ je $\sum_{j=1}^{n} a_{ij} x_j \leqq b_i$, potom stačí písať do obmedzení len nerovnosť $XI \leqq BI$, kde $XI = sumproduct(A_{m_1};X) : sumproduct(A_{m_2};X)$ je oblasť lineárnych funkcií a $BI = b_{m_1} : b_{m_2}$ je oblasť konštánt.

Úlohu lineárneho programovania môžeme prirodzene zapísať aj v maticom tvare

$$\min\{cx : Ax = b,\ x \geqq 0\},$$

[4] Názvoslovie tu nie je ustálené, pre *oblasť* sa používajú i názvy *pole, tabuľka*.

[5] Nepovinné alebo alternatívne parametre budeme značiť v hranatých zátvorkách [♣].



Obrázok 1: Riešiteľ pre lineárne (celočíselné) programovanie

čo vedie k alternatívnej implementácii úlohy s použitím nielen skalárneho, ale aj maticového súčinu matíc. Pred samotným výberom implementácie sa nám osvedčilo diskutovať so študentami o výhodách a nevýhodách, a tak im poskytnúť možnosť voľby s následným porovnávaním alternatívnych prístupov.

3 Minimalizácia počtu prestupov medzi linkami

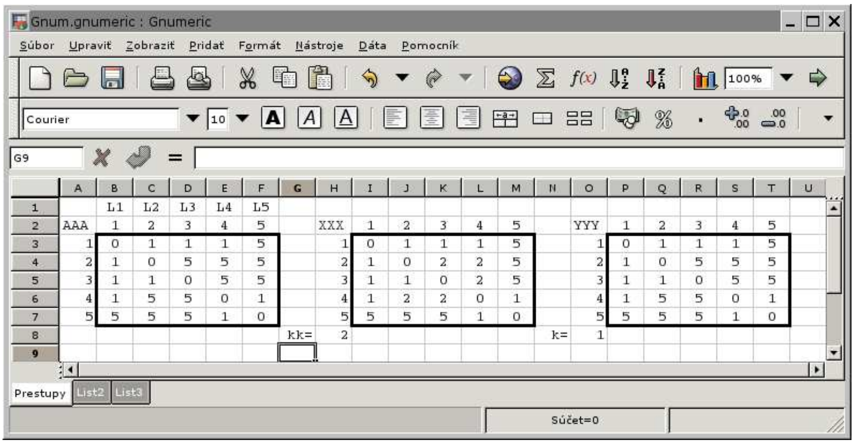
Najskôr uvedieme model, ktorého riešenie nevyžaduje použitie *Riešiteľa*, ale vystačí s citlivo „naprogramovanými“ odkazmi buniek. Pri vyhodnocovaní alternatívneho návrhu vedenia liniek MHD v Nitre vznikla potreba riešiť nasledujúcu optimalizačnú úlohu:

Nech je daná množina n liniek $\mathscr{L} = \{L_i : i \in N\}, N = \{1, 2, \ldots, n\}$ *a matica* $\mathrm{A} = (a_{ij})$, *kde prvok* $a_{ij} = 1$, *ak je možný prestup z linky* $L_i \in \mathscr{L}$ *na linku* $L_j \in \mathscr{L}$ *(t. j. linky majú aspoň jednu spoločnú zastávku) a* $a_{ij} = n$ *v opačnom prípade. Hľadá sa matica* $\mathrm{D} = (d_{ij})$, *kde prvok* d_{ij} *udáva minimálny počet prestupov z linky* L_i *na linku* L_j.

Takto formulovanú úlohu možno riešiť napr. známym Floydovým algoritmom na hľadanie matice minimálnych vzdialeností, ktorý by bolo možné implementovať nasledujúcou procedúrou $\mathrm{D} = \mathit{FLOYD}(\mathrm{A})$:

procedure FLOYD(A)
  $\mathrm{D} = \mathrm{A}$
  for $k \in N$ **do**
    for $i \in N$ **do**
      for $j \in N$ **do**
        if $d_{ij} > d_{ik} + d_{kj}$ **then**
          $d_{ij} = d_{ik} + d_{kj}$
return D

Ak však nepoznáme príslušný procedurálny jazyk tabuľkového procesora a nechceme sa ho učiť, ponúka sa nám možnosť využiť odkaz bunky na jednu bunku alebo na funkciu viacerých buniek oblastí (napr. $kk = if(k < n, k+1, n)$ realizuje počítadlo vonkajšieho cyklu indexu k algoritmu).



Obrázok 2: Floydov algoritmus pomocou odkazov po prvej iterácii

Na obrázku 2 máme v Gnumericu realizáciu Floydovho algoritmu pomocou odkazov pre úlohu s piatimi linkami. Najskôr sú v liste zošitu „Prestupy“ pomenované oblasti $XXX = Prestupy!\$I\$3 : \$M\$7$ a $YYY = Prestupy!\$O\$3 : \$T\$7$, pričom je na začiatku výpočtu oblasť $YYY = AAA$. Ak položíme bunku $I3$ rovnú

$$I3 = min(P3; index(YYY; \$H3; \$O\$8) + index(YYY; \$O\$; P\$2))$$

a rozkopírujeme do celej oblasti XXX, realizujeme k-ty iteračný krok procedúry. Vyššie uvedená formula je vlastne jadrom nášho neprocedurálneho programu. Opakovaným kopírovaním hodnôt oblasti YYY a kk do oblasti XXX a k, dostávame riešenie $XXX = YYY =$ D. Počítadlo v bunke $H8 = if(O8 < A7; O8+1; A7)$ po každom kopírovaní zvýši hodnotu o $+1$, kým nedosiahne hodnotu 5.

Ak definujeme prvky a_{ij} rovné priemernej dobe prestupu medzi linkami L_i a L_j pre linky so spoločnou zastávkou a ∞ v opačnom prípade, potom naším algoritmom môžeme vypočítať tabuľku rozpätia priemernej doby trvania prestupov medzi linkami s hodnotami buniek d_{ij} medzi ľubovoľnými dvoma linkami.

Pri analýze vedenia pätnástich liniek MHD Nitra [2] sme zistili, že na niektorých zastávkach liniek sú potrebné až 3 prestupy a najdlhšia doba priemerného prestupu medzi linkami bola 32 minút.

4 Problém nakupujúceho obchodného cestujúceho

Pri tvorbe okružnej jazdy vozidla môže vzniknúť potreba riešiť nasledujúcu optimalizačnú úlohu:

Nech je daná matica dopravných nákladov medzi uzlami $N = \{0, 1, 2, \ldots, n\}$ *dopravnej siete* $\mathrm{D} = (d_{ij})$*. Množina uzlov obsahuje uzol* 0 *reprezentujúci východiskové miesto obchodného cestujúceho, ktorý hodlá nakúpiť sortiment p druhov tovaru,* $K = \{1, 2, \ldots, p\}$*, v množstvách* $(b_1, b_2, \ldots, b_p)$ *v niektorých predajniach, ktoré sú umiestnené v ostatných uzloch* $i \in M = N - \{0\}$*, a ponúkajú sortiment v množstvách* $(a_{i1}, a_{i2}, \ldots, a_{ip})$ *za jednotkovú cenu* $(c_{i1}, c_{i2}, \ldots, c_{ip})$*.*

Hľadá sa taká pochôdzka obchodného cestujúceho, ktorá začína a končí v 0 *a prechádza predajňami i, v ktorých nakupuje dostatočné množstvá k-teho druhu tovaru* y_{ik}*. Cieľom je minimalizovať celkové dopravné náklady a náklady na nákup vybraného tovaru.*

Pri hľadaní vhodného modelu sa necháme motivovať modelom Millera a kol., pre klasickú úlohu obchodného cestujúceho (bez nákupu, ale navštevujúceho všetky uzly siete), ktorý je uvedený v monografii [1] na str. 26–27, kde je cyklus (pochôdzka) obchodného cestujúceho repezentovaný $0-1$ maticou $\mathrm{X} = (x_{ij})$. Ak je $x_{ij} = 1$ potom hľadaný cyklus obsahuje úsek $i \to j$. My však nepotrebujeme navštíviť všetky predajne, čo docielime, ak položíme $d_{00} = \infty$ a $d_{ii} = 0$ v ostatných prípadoch keď $i \in M$. Potom $x_{ii} = 1$ bude znamenať, že predajňa i nebola navštívená. A tak dostávame nasledujúcu úlohu zmiešaného programovania BTSP:

$$\sum_{i\in N}\sum_{j\in N} d_{ij}x_{ij} + \sum_{i\in M}\sum_{k\in K} c_{ik}y_{ik} \to \min, \tag{4}$$

$$\sum_{j\in N} x_{ij} = 1, \qquad i \in N, \tag{5}$$

$$\sum_{i\in N} x_{ij} = 1, \qquad j \in N, \tag{6}$$

$$u_i + (n+1)x_{ij} - u_j \leqq n, \qquad i, j \in M,\ i \neq j, \tag{7}$$

$$y_{ik} + a_{ik}x_{ii} \leqq a_{ik}, \qquad i \in M,\ k \in K, \tag{8}$$

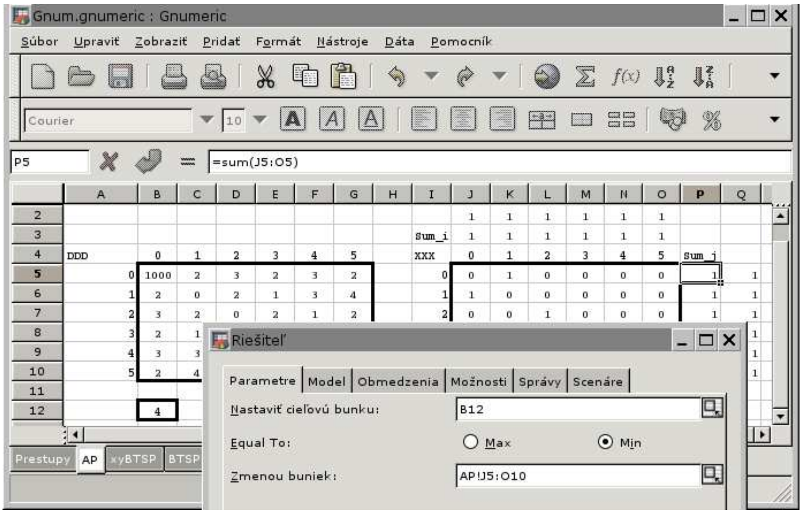
$$\sum_{i\in M} y_{ik} = b_k, \qquad k \in K, \tag{9}$$

$$y_{ik} \geqq 0, \qquad i \in M,\ k \in K, \tag{10}$$

$$u_i \geqq 0, \qquad i \in M, \tag{11}$$

$$x_{ij} \in \{0, 1\}, \qquad i, j \in N. \tag{12}$$

Obmedzujúce podmienky (5), (6), (12) sú podmienkami klasickej priraďovacej úlohy (assignment problem – AP). V liste AP na obrázku 3 máme riešenie príkladu s piatimi miestami. V oblastiach $DDD = AP!\$B\$5 : \$G\$10$ a $XXX = AP!\$J\$5 : \$O\$10$ máme maticu vzdialeností a maticu riešenia. Cieľovou bunkou je $B12 = sumproduct(DDD; XXX)$.



Obrázok 3: Riešenie priraďovacieho problému v BTSP

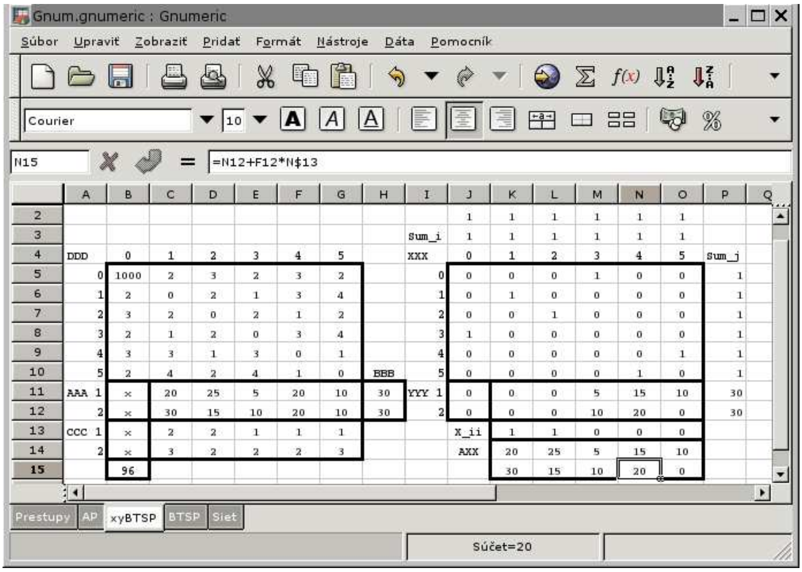
Riadkové a stĺpcové súčty (5), (6) sú v oblastiach Sum_j, Sum_i v tvare súčtových formúl, napr. $P5 = sum(J5 : O5)$. Za zmienku stojí, že stačí rozkopírovať túto bunku do oblasti Sum_j a máme korektne definované všetky jej bunky. Riešením príslušnej úlohy LP sú nasledujúce cykly:

$$0 \to 3 \to 0, 1 \to 1, 2 \to 2, 4 \to 4, 5 \to 5.$$

Vidíme, že úloha nepožadovala explicitne podmienku (12) bivalentnosti premenných v oblasti XXX. Tá je zabezpečená unimodálnosťou obmedzujúcich podmienok (5), (6). Bivalentnosť premenných tak môže byť nahradená slabšou obligátornou podmienkou $x_{ij} \geqq 0$, čo vedie na úlohu lineárneho programovania.

Podmienka (8) v BTSP zabezpečí nulové množstvá tovaru $y_{ik} = 0$ z celého sortimentu predajne i, ak nebude navštívená, t. j. je v triviálnom cykle $i \to i$ definovanom premennou $x_{ii} = 1$. V opačnom prípade $x_{ii} = 0$ pripúšťa nákup z disponibilného množstva tovaru. Kapacitná podmienka (9) umožňuje nákup všetkého požadovaného tovaru. Podmienky (10), (11) sú obligátorné. Anticyklická podmienka (7) s obligátornou podmienkou (11) nám zaručuje, že v riešení úlohy neexistuje netriviálny cyklus neobsahujúci uzol 0.

Na obrázku 4 máme riešenie dvojsortimentovej úlohy $(p = 2)$ najskôr bez anticyklických podmienok, aby sme sa presvedčili o ich potrebe. Najskôr dodefinujeme nové oblasti vstupov a to:
$AAA = xyBTSP!\$C\$11 : \$G\$12$,
$BBB = xyBTSP!\$H\$11 : \$H\$12$,
$CCC = xyBTSP!\$C\$13 : \$G\$14$
a oblasti výstupov $YYY = xyBTSP!\$K\$11 : \$O\$12$, $X_{ii} = xyBTSP!\$K\$6 : \$O\$6$, obsahujúce odkazy na diagonále bunky oblasti XXX a $AXX = xyBTSP!\$K\$14 : \$O\$15$, tvoriacu formuly



Obrázok 4: Riešenie dvojsortimentovej BTSP bez anticyklických podmienok

ľavých strán obmedzení (8). Cieľová bunka $B15$ má tvar formuly

$$B15 = sumproduct(DDD; XXX) + sumproduct(CCC; YYY).$$

Riešenie je pri nutnej požiadavke celočíselnosti premenných oblasti XXX

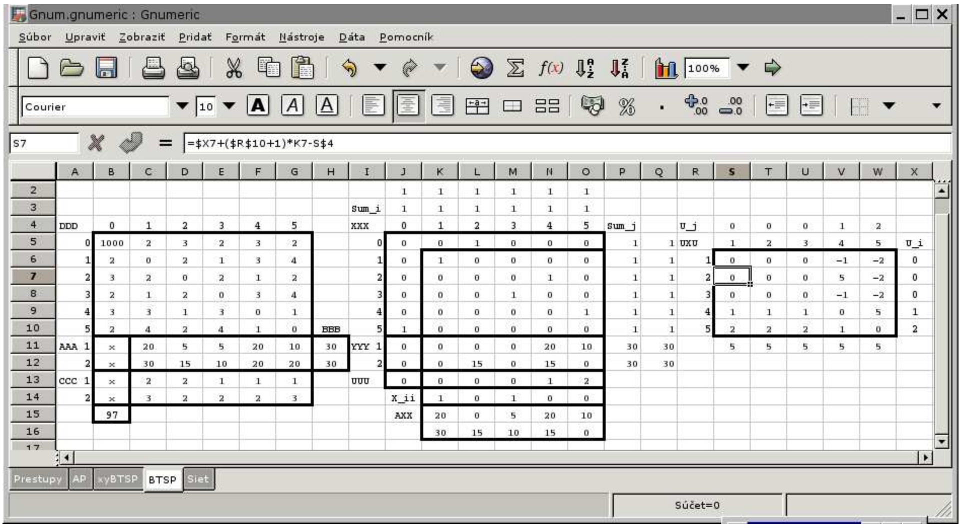
$$0 \to 3 \to 0, 1 \to 1, 2 \to 2, 5 \to 4 \to 5,$$

ale je neprípustné, nakoľko obsahuje netriviálny cyklus $5 \to 4 \to 5$.

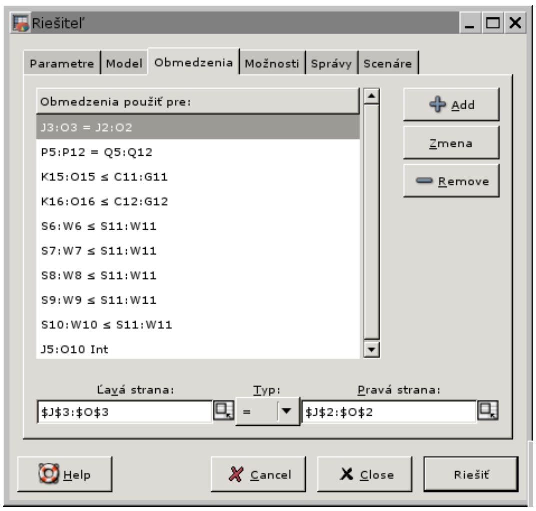
Doplnenie anticyklických podmienok (7) s obligatórnou podmienkou (11) na nové premenné u_i vyžaduje zaviesť ďalšiu oblasť UUU. Pretože *Riešiteľ* umožňuje pracovať len so súvislou oblasťou premenných, vložíme UUU za oblasť YYY, čím získame oblasť premenných $BTSP!J5 : O13$. Na implemetáciu podmienok (7) potrebujeme definovať oblasť $UXU = BTSP!S5 : W10$, ktorej bunka $S7 = \$X7 + (\$R\$10 + 1) * K7 - S\$4$. Po jeho rozkopírovaní do celej oblasti UXU musíme ešte zmeniť diagonálne bunky z formúl na 0 hodnoty, čím ich vylúčime z optimalizácie. Po vložení všetkých podmienok dvojsortimentovej BTSP dostávame riešenie:

$$0 \to 2 \to 4 \to 5 \to 0, 1 \to 1, 3 \to 3,$$

čo znamená, že obchodný cestujúci nakupuje postupne v predajniach $2, 4, 5$. Príslušné množstvá požadovaného sortimentu nájdeme v oblasti YYY.



Obrázok 5: Oblasti pre formuláciu dvojsortimentovej BTSP

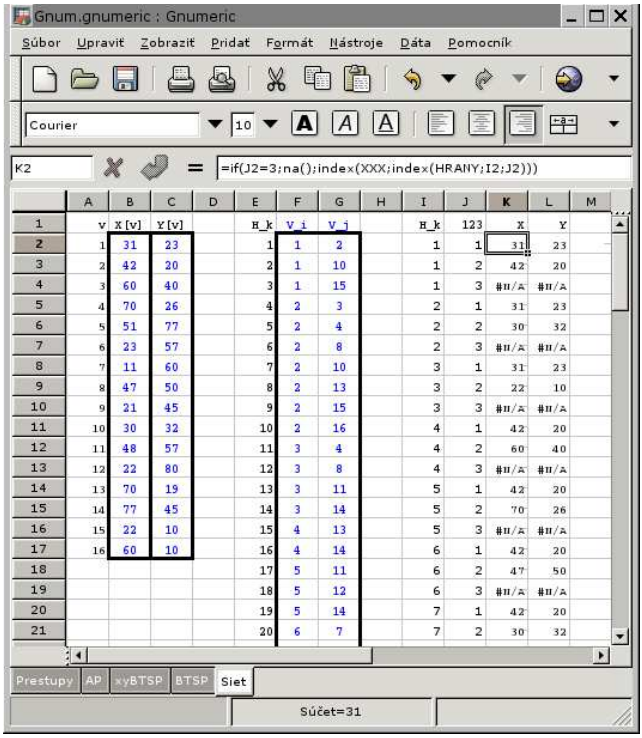


Obrázok 6: Obmedzujúce podmienky na riešenie dvojsortimentovej BTSP

5 Graf dopravnej siete

Pre potreby grafickej reprezentácie niektorých prvkov, resp. grafových štruktúr na dopravnej sieti (napr. sledov, ciest, centier, diep, atď.), nemajú tabuľkové procesory priamu grafickú podporu. Samotný graf (presnejšie diagram) dopravnej siete môžeme pohodlne reprezentovať množinou úsečiek vďaka pomerne jednoduchému triku.

Na obrázku 7 máme v Gnumericu dopravnú sieť určenú množinou súradníc $X[v], Y[v]$



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	v	X[v]	Y[v]		H_k	V_i	V_j		H_k	123	X	Y	
2	1	31	23		1	1	2		1	1	31	23	
3	2	42	20		2	1	10		1	2	42	20	
4	3	60	40		3	1	15		1	3	#N/A	#N/A	
5	4	70	26		4	2	3		2	1	31	23	
6	5	51	77		5	2	4		2	2	30	32	
7	6	23	57		6	2	8		2	3	#N/A	#N/A	
8	7	11	60		7	2	10		3	1	31	23	
9	8	47	50		8	2	13		3	2	22	10	
10	9	21	45		9	2	15		3	3	#N/A	#N/A	
11	10	30	32		10	2	16		4	1	42	20	
12	11	48	57		11	3	4		4	2	60	40	
13	12	22	80		12	3	8		4	3	#N/A	#N/A	
14	13	70	19		13	3	11		5	1	42	20	
15	14	77	45		14	3	14		5	2	70	26	
16	15	22	10		15	4	13		5	3	#N/A	#N/A	
17	16	60	10		16	4	14		6	1	42	20	
18					17	5	11		6	2	47	50	
19					18	5	12		6	3	#N/A	#N/A	
20					19	5	14		7	1	42	20	
21					20	6	7		7	2	30	32	

Obrázok 7: Vrcholy a hrany dopravnej siete

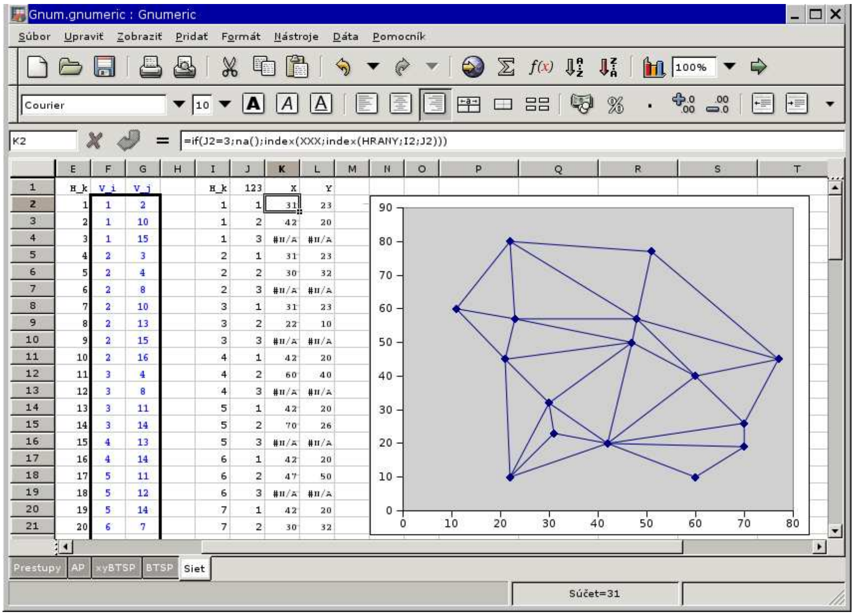
jej vrcholov v a množinou jej hrán $[v_i, v_j]$. V oblastiach $XXX = Siet!\$B\$2 : \$B\$17$ a $YYY = Siet!\$C\$2 : \$C\$17$ máme zoznam súradnic vrcholov a zoznam hrán je v oblasti $HRANY = Siet!\$F\$2 : \$G\$36$.

Definujeme oblasť $Siet!\$I\$2 : \$J\$4 = \{1,1;1,2;1,3\}$ a oblasť $Siet!I5 : J5 = \{F2 + 1, G2\}$. Formuly buniek $H2$ a $G2$ udávajú súradnice $X[v], Y[v]$ príslušného koncového vrcholu hrany $[v_i, v_j]$, alebo je to oddeľovač – nedefinovaná bunka $na()$:

$$
\begin{aligned}
H2 &= if(G2 = 3, na(), index(XXX, index(HRANY, I2, J2))),\\
G2 &= if(G2 = 3, na(), index(YYY, index(HRANY, I2, J2))).
\end{aligned}
$$

Najskôr rozkopírujeme oblasť $I5 : J5$ do oblasti $I6 : J106$ a potom aj oblasť $K2 : L2$ do oblasti $K3 : L106$, čím získame súradnice X, Y úsečiek, ktoré reprezentujú hrany siete. Zobrazením oblasti $\$K\$2 : \$L\$106$ v XY bodovom grafe dostaneme, na obrázku 8, želaný obrázok diagramu dopravnej siete.

Analogicky sa postupuje, ak chceme v dopravnej sieti farebne vyznačiť niektoré jej hrany, napr. hrany najkratšej uv-cesty, najlacnejšej kostry, páriace hrany, atď. Jediným doteraz neodstráneným nedostatkom tohto prístupu je, že sa nám do grafu nepodarilo pridať popisky, napr. ako názvy vybraných uzlov.



Obrázok 8: Graf dopravnej siete

Vyššie uvedený trik teda spočíva v nakreslení sledu úsečiek ako prerušovanej lomenej čiary. Inou, podstatne pracnejšou cestou, je implementovať Tarryho prieskumom lomenú čiaru (komponent siete), ktorá každou hranou prechádza v jednom smere práve raz. Tento postup však možno odporúčať len „veľmi hravým“ študentom so znalosťou teórie grafov.

Pri analýze cestovných poriadkov MHD Nitra [2] bola sieť MHD tvorená pätnástimi linkami. Vrcholmi tu boli zastávky liniek a hranami úseky liniek medzi zastávkami. Získali sme tak prehľad o spoločných úsekoch liniek, čo nám pomohlo pri posudzovaní zhlukov autobusov cestovných poriadkov na zastávkach týchto úsekoch.

6 Záver

Uvedené praktické príklady ukazujú, že je možné aj bez znalosti procedurálneho programovania v tabuľkových procesoroch modelovať reálne optimalizačné úlohy. Limitujúcim faktorom v prípade použitia *Riešiteľa* vo výskume je jeho kvalita a najmä rozsah reálne riešiteľných úloh. No i v prípade malých inštancií pomáha pri tvorbe alternatívnych modelov.

Oboznamovanie študentov s touto formou modelovania sa začína stretávať na katedre matematických metód s dobrou odozvou, nakoľko im umožňuje pomerne ľahko overovať korektnosť vytvorených modelov. Na strane vyučujúceho zas poskytuje veľkú variabilitu pri príprave úloh na cvičenia, čo občas vedie až k obojstranne nákazlivému potešeniu z tvorivosti.

Domnievame sa, že dostupné tabuľkové procesory by mohli zaujať dominantné postavenie ako nástroje výučby modelovania v predmetoch operačnej analýzy. Ponechávajú totiž priestor pre tvorivosť študentov a pri nápaditom vedení umožňujú sústrediť ich pozornosť hlavne na návrh a analýzu modelov.

Poďakovanie

Táto práca vznikla s podporou grantovej agentúry VEGA v rámci riešenia projektu 1/3775/06.

Literatúra

[1] LAWLER E. L., LENSTRA J. K., RINNOOY KAN A. H. G. and SHMOYS D. B.: *The traveling salesman problem, A Guided Tour of Combinatorial Optimization*. A Wiley-Interscience Publication, ISBN 0-471-90413-9, 1985

[2] ČERNÝ, J., PALÚCH, S., PEŠKO, Š.: *Odborný posudok alternatívneho návrhu liniek a cestovných poriadkov MHD v meste Nitra*. Štúdia, máj 2005

[3] *The Gnumeric Manual*. `http://www.gnome.org/projects/gnumeric/doc/gnumeric.shtml`

[4] PEŠKO, Š.: *Pohodlná optimalizácia reálnych úloh v tabuľkových procesoroch*. Slovak Society for Operations Research, 7-th international seminar, APPLICATION OF QUANTITATIVE METHODS IN RESEARCH AND PRACTICE, pp. 29–35, ISBN 80-225-2079-9, Remata, 2005

[5] PEŠKO, Š.: *Vybrané modely logistiky v EXCELi*, učebné texty k cvičeniam. Jún 2002, `http://frcatel.fri.utc.sk/~pesko/volk.zip`

Kontaktná adresa

Štefan Peško, (RNDr.,CSc.),
Katedra matematických metód, Fakulta riadenia a informatiky, Žilinská univerzita v Žiline, Univerzitná 8215/1, 010 26 Žilina, `pesko@frcatel.fri.utc.sk`

6th INTERNATIONAL CONFERENCE APLIMAT
Section Open Source Software in Research and Education
February 6–9, 2007
Bratislava, Slovakia

Organizers: Michal Kaukič and Miloš Šrámek
Reviewers: Ján Buša, Michal Kaukič, Dušan Mamrilla, Peter Mann, Andrej Petráš, Karel Šotek and Miloš Šrámek
Editors: Michal Kaukič, Miloš Šrámek, Ladislav Ševčovič and Ján Buša

ISBN 978-80-969562-7-2

Zborník bol vydaný s podporou SKOSI, n. o.
