Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach

Žilina 2.–5. júla 2009

NAJLACNEJŠIE ÚPLNÉ PÁRENIE VO VŠEOBECNÝCH GRAFOCH S POUŽITÍM OSS NÁSTROJA GLPK

PEŠKO, Štefan, (SK)

1 Úvod

V teórii grafov [3] pod *úplným párením* v grafe $G = (V,H)$ rozumieme takú podmnožinu P množiny hrán H, že každý vrchol z množiny vrcholov V inciduje práve s jednou hranou z P. Budeme sa zaoberať problémom najlacnejšieho úplného párenia v grafe (MWPM – Minimum Weight Perfect Matching), ktorý možno formulovať takto:

V danom hranovo ohodnotenom grafe $G = (V,H,c)$, kde každá hrana $h \in H$ má cenu c_h (reálne číslo) treba nájsť úplné párenie P s minimálnou sumárnou cenou $c(P) = \sum_{h \in P} c_h$.

Okrem úloh MWPM s *bipartitnými grafmi* (napr. modelujúcich priradenie pracovníkov k strojom), ktoré možno ľahko formulovať ako úlohy LP (lineárneho programovania) o najlacnejšom prípustnom toku v sieti, nie je nám vo *všeobecných grafoch* známa takáto transformácia. Medzi základné výsledky kombinatorickej optimalizácie patrí Edmondsovo riešenie problému MWPM časovo polynomiálnym *kvetovým* algoritmom, ktorý je mimoriadne náročný na implementáciu [1]. S potrebou riešiť takéto úlohy sme sa stretli v dopravnej logistike [4] pri heuristickom riešení problémov tvorby rozvrhov vozidiel a osádok alebo kapacitných okružných dopravných úloh, ale aj pri tvorbe študijných skupín v univerzitnom rozvrhu.

V tomto príspevku sa chceme podeliť nielen o alternatívne riešenie problém MWPM stredných rozmerov pomocou OSS knižníc pre zmiešané celočíselné lineárne programovanie (MILP Mixed Integer Linear Programming) – konkrétne nástrojov GLPK [2] s možnosťou využitia pohodlného modelovacieho jazyka GMPL (GNU Mathematical Programming Language), ale aj o cestu, ktorá nás priviedla k úvahám nad celočíselným jadrom MILP.

2 Edmondsov lineárny model

Edmondsov algoritmus z roku 1965 je založený na LP formulácii úlohy MWPM. Nech $G = (V, H, c)$ je hranovo ohodnotený graf s párnym počtom vrcholov a nech $\mathscr{C}$ označuje množinu všetkých aspoň trojprvkových podmnožín množiny vrcholov V s nepárnym počtom prvkov. Pre všetky podmožiny $S \subseteq V$ budeme značiť $\delta(S)$ množinu hrán, ktoré majú práve jeden vrchol v S. Pre vektor $(x_h : h \in H)$ a množinu $E \subseteq H$ nech $x(E) = \sum_{e \in E} x_e$. Teraz už môžeme formulovať Edmonsovu primárnu úlohu lineárneho programovania [1] (ELP):

$$\sum_{h \in H} c_h x_h \rightarrow \min, \tag{1}$$

$$x(\delta(\{v\})) = 1 \quad \forall v \in V, \tag{2}$$

$$x(\delta(S)) \geqq 1 \quad \forall S \in \mathscr{C}, \tag{3}$$

$$x_h \geqq 0 \qquad \forall h \in H. \tag{4}$$

Cieľová funkcia (1) minimalizuje cenu párenia. Incidenčná podmienka (2) zabezpečuje aby každý vrchol incidoval s práve jednou hranou grafu. Podmienka (3) zaručuje bivalentnosť premenných x_h od ktorých sa požaduje len obligátorná podmienka nezápornosti (4).

Významným úspechom Edmondsovej formulácie je, že sa autorovi podarilo ostrániť nutnosť požiadavky bivalentnosti premenných x_h. Jej slabým miestom je, že početnosť množiny $\mathscr{C}$ rastie exponenciálne s početnosťou množiny vrcholov V. Tento nedostatok je ale vyriešený pomocou duálnej úlohy k úlohe ELP a podmienok komplementarity týchto duálne združených úloh. Od publikovania Edmondsovho algoritmu, ktorý má zložitosť $O(|V|^2 \cdot |H|)$ uvádza Cook a kol. [1] do roku 1999 až 19 rôznych počítačových implementácií, ktoré sa líšia použitými dátovými štruktúrami a zložitosťou algoritmov. V citovanej práci boli úspešne vyriešené inštancie až s 5 000 000 vrcholmi pričom 100 000 - vrcholové geometrické inštancie boli vyriešené do troch minút na počítači 200 MHz Pentium–Pro.

Náš cieľ je oveľa skromnejší, čo do rozmeru riešených inštancií úloh i doby výpočtu. Začneme s formuláciou úlohy, ktorá je vhodná pre riešiče MILP.

3 MILP formulácie problému

Najjednoduchšou bivalentnou formuláciou úlohy MWPM, ktorá má zhodný počet premenných aj ich interpretáciu ako v úlohe ELP, je nasledujúca úloha (BLP):

$$\sum_{h \in H} c_h x_h \rightarrow \min, \tag{5}$$

$$x(\delta(\{v\})) = 1 \quad \forall v \in V, \tag{6}$$

$$x_h \in \{0, 1\} \qquad \forall h \in H. \tag{7}$$

V tejto úlohe sme len nahradili podmienky (3) a (4) úlohy ELP podmienkou (7). Výhodou tejto formulácie je len jej jednoduchosť. Jej nevýhoda, neprijateľne dlhá doba výpočtu, sa prejaví už pri riešení úloh stredných rozmerov $|V| \approx 60$.

Pozorovali sme, že jednou z príčin dlhého výpočtu je veľký počet celočíselných premenných, ktorý spôsobuje vetvenie hlboko v strome riešení aj v prípade, keď je optimálne riešenie nájdené, ale ešte nie je overené. Preto je potrebné nájsť takú MILP formuláciu úlohy, v ktorej bude čo najmenej celočíselných premenných.

Prvý model, ktorý spĺňal naše požiadavky, vychádzal z formulácie klasického priraďovacieho problému v úplnom bipartitnom grafe K_{nn} s párnym počtom n vrcholov v oboch častiach $S = \{1,2,\ldots,n\}$ a $T = \{1,2,\ldots,n\}$ grafu, ktorého dvojice (i,j) sú ohodnotené symetrickou maticou $C = (c_{ij})$ pre $(i,j) \in S \times T$, kde $c_{ii} = \infty$. Nech (x_{ij}) je hľadaná bivalentná matica neznámych, ktorá priradí hrane (i,j) hodnotu 1 ak je $\{i,j\}$ hranou hľadaného párenia P v grafe G. Dostávame tak nasledujúcu úlohu zmiešaného celočíselného lineárneho programovania (AMLP):

$$\sum_{i\in S}\sum_{j\in T} c_{ij}x_{ij} \quad \to \min, \tag{8}$$

$$\sum_{j\in S} x_{ij} = 1 \qquad \forall i \in S, \tag{9}$$

$$\sum_{i\in T} x_{ij} = 1 \qquad \forall j \in T, \tag{10}$$

$$x_{ij} \geqq 0 \qquad \forall (i,j) \in S \times T, \tag{11}$$

$$x_{ij} - x_{ji} = 0 \qquad \forall (i,j) \in S \times T : i < j, \tag{12}$$

$$y_i - \sum_{j\in T: j>i} j x_{ij} = 0 \qquad \forall i \in S - \{n\}, \tag{13}$$

$$y_i \geqq 0, \text{ celé} \qquad \forall i \in S - \{n\}. \tag{14}$$

Klasický priraďovací problém je tu formulovaný LP úlohou (8)–(11). Symetrizačná podmienka (12) zabezpečí, že hrana párenia $\{i,j\}$ je reprezentovaná dvojicou $(i,j),(j,i) \in S \times T$, pre ktoré je $x_{ij} = x_{ji}$. Dôkaz, že jednotkové hodnoty páriacich premených $x_{ij} = x_{ji} = 1$ vynucujú podmienky (13) a (14) bude podrobne vyložený v pripravovanom článku Peško [5]. Do tých čias sa uspokojíme s experimentálnym overením tohoto poznatku.

Model AMLP sme formulovali v úplnom bipartitnom grafe K_{nn}, aby sa ukázalo, či počítačové experimenty na náhodných inštanciách úplného grafu G nenájdu neceločíselné optimálne riešenie. Ukázalo sa, že celočíselné premenné y tvoria *celočíselné jadro* [5] modelu AMLP, t. j. *z celoločíselnosti premenných jadra vyplýva celočíselnosť všetkých bázických riešení uvažovanej úlohy MILP*.

Počítačové experimenty ukázali podstatné zrýchlenie výpočtu v modeli AMLP oproti modelu BLP. Naviac pre reálnu inštanciu so 100 vrcholmi, na ktorej riešič *glpsolve* modelu BLP nedopočítal riešenie ani po cca 16 hodinách výpočtu, bolo nájdené riešenie v modeli

AMLP po cca 50 minútach. To nás priviedlo k finálnej modifikácii modelu BLP, v ktorom sa naviac požaduje celočíselnosť len dodatočného vektoru $n-1$ premených $\big(y_i : i \in V - \{n\}\big)$ čo vedie k úlohe (YLP):

$$\sum_{\{i,j\}\in H} c_{\{i,j\}} x_{\{i,j\}} \quad \rightarrow \min, \tag{15}$$

$$\sum_{\{i,j\}\in \delta(\{i\})} x_{\{i,j\}} \quad = 1 \quad \forall i \in V, \tag{16}$$

$$x_{\{i,j\}} \geqq 0 \quad \forall \{i,j\} \in H. \tag{17}$$

$$\sum_{\{i,j\}\in \delta(\{i\}):j>i} j \cdot x_{\{i,j\}} - y_i = 0, \quad \forall i \in V - \{n\}, \tag{18}$$

$$y_i \geqq 0, \text{ celé} \quad \forall i \in V - \{n\} \tag{19}$$

Relaxácia modelu BLP je tu formulovaná v tvare LP úlohy (15)–(17). Podmienky celočíselného jadra (18) a (19) sú len prepisom podmienok (13) a (14) v reči hrán grafu $G = (V,H,c)$. Podarilo sa nám tak nahradiť exponenciálny počet obmedzení v Edmondsovej podmienke (3) alternatívnymi podmienkami celočíselného jadra.

Pre implementáciu modelov BLP a YLP sa ukazuje výhodným reprezentovať hrany $\{i, j\} \in H$ takými usporiadanými dvojicami (i, j), pre ktoré platí $i < j$. Potom môžeme hrany okolia vrcholu $k \in V$ definovať vzťahom

$$\delta(k) = \{(i, j) \in H : i = k \; or \; j = k\}.$$

4 Riešič glpsolve nástroja GLPK

Riešič *glpsolve* nástroja GLPK pre úlohy LP resp. MILP môže používať viacero modelovacích jazykov. My sme zvolili modelovací jazyk *GNU MathProg*, ktorého syntax je intuitívna, ako vidieť z nasledujúceho textového súboru YLP.mod. Súbor obsahuje modelové špecifikácie optimalizačnej úlohy YLP a datové špecifikácie konkrétnej inštancie úlohy.

Modelová špecifikácia definuje potrebné parametre a samotný model pomocou príkazov jazyka, pričom obmedzeniu (17) zodpovedá príkaz *s.t. match* a obmedzeniu (18) príkaz *s.t. kernel*. Datová špecifikácia obsahuje príkaz na určenie počtu vrcholov a zoznamu ocenených hrán uvažovanej inštancie – v našom prípade ilustračného grafu so 6-timi vrcholmi. Poznamenajme, že ak by sme z modelu odstránili premenné y a podmienku *s.t. kernel* dostaneme relaxovanú úlohu BLP, ktorej riešenie je neceločíselné.

```
/* MWPM Minimum Weight Perfect Matching */
# glpsol --math YLP.mod
```

```
param n, integer, >= 2;
set V := {1..n};
set H, within V cross V;
param c{(i,j) in H};

var x{(i,j) in H}, >=0;
var y{i in V: i<n}, >=0, integer;
s.t. match{k in V}: sum{(i,j) in H: i=k or j=k} x[i,j]  = 1;
s.t. kernel{i in V: i<n}: sum{(i,j) in H} j*x[i,j]  - y[i] = 0;
minimize obj: sum{(i,j) in H} c[i,j]*x[i,j];
solve;

/*========================================*/
printf "\nCena  ---> %d\n",sum{(i,j) in H} c[i,j]*x[i,j] ;
printf("  { i    j }  c[i,j] \n===================\n");
for {(i,j) in H: x[i,j] >0} printf " %4d %4d %6d\n",i,j,c[i,j];
printf "\n";

/*========================================*/
data;
param n := 6;
param : H : c :=
1   2   1
1   3   1
1   5   2
2   3   1
2   4   2
2   6   2
3   5   2
4   5   1
4   6   1
5   6   1
;
end;
```

Teraz už len stačí spustiť v prikazovom riadku riešič prikazom *glpsol –math YLP.mod* a dostaneme nasledujúci výpis:

```
Reading model section from YLP.mod...
Reading data section from YLP.mod...
49 lines were read
Generating match...
Generating kernel...
Generating obj...
Model has been successfully generated
ipp_basic_tech:  1 row(s) and 0 column(s) removed
ipp_reduce_bnds: 2 pass(es) made, 15 bound(s) reduced
```

```
ipp_basic_tech:  0 row(s) and 0 column(s) removed
ipp_reduce_coef: 1 pass(es) made, 0 coefficient(s) reduced
glp_intopt: presolved MIP has 11 rows, 15 columns, 35 non-zeros
glp_intopt: 5 integer columns, none of which are binary
Scaling...
 A: min|aij| =  1.000e+00  max|aij| =  6.000e+00  ratio =  6.000e+00
Problem data seem to be well scaled
Crashing...
Size of triangular part = 10
Solving LP relaxation...
      0: obj =   6.000000000e+00  infeas =  1.000e+00 (1)
*     2: obj =   5.000000000e+00  infeas =  0.000e+00 (0)
*     9: obj =   3.000000000e+00  infeas =  0.000e+00 (0)
OPTIMAL SOLUTION FOUND
Integer optimization begins...
+     9: mip =     not found yet >=              -inf        (1; 0)
+    21: >>>>>   4.000000000e+00 >=   3.333333333e+00  16.7% (9; 0)
+    43: mip =   4.000000000e+00 >=     tree is empty   0.0% (0; 27)
INTEGER OPTIMAL SOLUTION FOUND
Time used:   0.0 secs
Memory used: 0.2 Mb (179625 bytes)

Cena  ---> 4
  { i    j }  c[i,j]
====================
    1    3      1
    2    4      2
    5    6      1

Model has been successfully processed
```

Ak chceme experimentovať s rôznymi inštanciami úlohy, potom stačí vytvoriť textový súbor MWPM.dat, ktorý obsahuje len datovú špecifikáciu t. j. *data . . . end;* a spustiť program *glpsol –math YLP.mod –data MWPM.dat*. Riešič má k dispozícii množstvo ďalších prepínačov, pomocou ktorých môžeme meniť voľbu stratégií vetvenia, rezných nadrovín atď.

5 Záver

Naše experimenty s nástrojom GLPK riešiča pre úlohy LP/MILP nás utvrdili v dobrej skúsenosti s OSS. Študovaný problém MWPM bol pomerne jednoduchý na formuláciu, ale dostatočne obtiažny na riešenie. Pri experimentálnom hľadaní vhodného modelu sa nám podarilo, vďaka dobre zdokumentovanému a stabilnému sofvéru, získať novú formuláciu optimalizačnej úlohy, ktorá využíva pri riešení ideu celočíselného jadra úlohy MILP.

Poďakovanie

Táto práca vznikla s podporou grantovej agentúry VEGA v rámci riešenia projektu 1/0135/08 „Optimalizačné problémy v logistických a dopravných systémoch“.

Literatúra

[1] COOK, W. – ROHE, A.: 1999. *Computing Minimum–Weight Perfect Matching*. INFORMS Journal on Computing, Vol. 11, No. 2, Spring, 1999

[2] MARKHORIN, A.: 2008. *GNU Linear Programming Kit*, Moscow : Reference Manual, Verson 4.29, pp. 154, 2008

[3] PLESNÍK, J.: 1983. *Grafové algoritmy*, Bratislava : VEDA, 1983

[4] PALÚCH, S. – PEŠKO, Š.: 2007. *Kvantitatívne metódy v logistike*, Žilina : EDIS, 2007, ISBN 80-8070-636-0,

[5] PEŠKO, Š.: 2009. *Minimum Integer Kernel for Integer Linear Programming*, Mathematical Programming, in preparation

Kontaktná adresa

Štefan PEŠKO (doc., RNDr., CSc.),
Katedra matematických metód, FRI ŽU v Žiline, Univerzitná 8215/1,
010 26 Žilina,
stefan.pesko@fri.uniza.sk

Fakulta riadenia a informatiky
Žilinská univerzita

OTVORENÝ SOFTVÉR VO VZDELÁVANÍ, VÝSKUME A V IT RIEŠENIACH

Zborník príspevkov medzinárodnej konferencie
OSSConf 2009

2.–5. júla 2009
Žilina, Slovensko