FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITY KOMENSKÉHO V BRATISLAVE

Ekonomická a finančná matematika

OPTIMALIZÁCIA PLÁNOVANIA ZÁSOBOVANIA A PREPRAVY V HETEROGÉNNOM PROSTREDÍ
(Diplomová práca)

Vypracoval: **Róbert HIKL**
Vedúci diplomovej práce: **Ing. Rastislav Rázus**
Bratislava, Apríl 2003

Úprimne ďakujem svojmu diplomovému vedúcemu Ing. Rastislavovi Rázusovi a prof. RNDr. Jaroslavovi Janáčkovi, CSc. za cenné rady a podnety, ktoré výraznou mierou prispeli k skvalitneniu obsahu práce.

Obsah

OBSAH **2**

ÚVOD **2**

1 FORMULÁCIA ÚLOHY **2**

2 ZÁKLADNÉ POJMY **2**

3 KLASIFIKÁCIA PREPRAVNÝCH ÚLOH **2**

4 OPTIMÁLNE A SUBOPTIMÁLNE METÓDY RIEŠENIA **2**

4.1 KLASIFIKÁCIA METÓD TRASOVANIA 2

4.2 MATEMATICKÉ MODELY 2

4.3 METÓDY RIEŠENIA ÚLOHY OBCHODNÉHO CESTUJÚCEHO 2

4.4 METÓDY RIEŠENIA ZVOZNÝCH A ROZVOZNÝCH ÚLOH 2

4.5 METAHEURISTIKY 2

5 MATEMATICKÝ MODEL ÚLOHY **2**

6 RIEŠENIE ÚLOHY **2**

6.1 OVPLYVŇUJÚCE FAKTORY A EXTERNÝ DODÁVATEĽ 2

6.2 METÓDA OPTIMALIZÁCIE PREPRAVY 2

6.3 PRÍKLAD RIEŠENÝ ALGORITMOM 2

ZÁVER **2**

LITERATÚRA **2**

PRÍLOHA A - VÝSLEDKY OPTIMALIZÁCIE **A-2**

PRÍLOHA B – NAJDÔLEŽITEJŠIE KROKY ALGORITMU **B-2**

PRÍLOHA C – DISKETA S PROGRAMOM **C-2**

Úvod

V tejto diplomovej práci sa pokúsime riešiť praktickú úlohu, ktorú sformulujeme v kapitole 1. Skôr ako sa však pustime do konkrétneho spracovania tejto úlohy, podáme si všeobecný teoreticky základ potrebný na pochopenie jadra problému a rovnako si vysvetlíme niektoré základné súvislosti.

V 2. kapitole sa oboznámime so základnými pojmami používajúcimi sa v odbornej literatúre. Vysvetlíme si pojem ako napr. dopravný element, orientovaná dopravná sieť, alebo okružná jazda.

V 3. kapitole sformulujeme niektoré z už riešených klasických úloh, ktoré z časti využijeme pri riešení nášho problému.

V 4. kapitole klasifikujeme metódy trasovania a sformulujeme niekoľko modelov. Nezameriame sa len na jeden typ modelu, ale ukážeme aj niektoré praktické rozšírenia, alebo aj niektoré iné modely, ktoré nemusia priamo súvisieť s naším problémom, ale každopádne pomôžu pochopiť a určiť súvislosti.

V 5. kapitole sformulujeme model úlohy, ktorá bola východiskom pre túto prácu, a uvedieme možnosti použitia tohto modelu..

V 6. kapitole sa pokúsime o riešenie tejto úlohy a uvedieme ilustračný príklad a na ňom si overíme naše riešenie.

Ale ešte predtým si sformulujem problém, ktorému sa budeme venovať:

1 Formulácia úlohy

Na začiatku sme postavení pred nasledujúci problém:

Uvažujme model, ktorý bude použitý pre optimalizáciu riadenia skladových zásob v prevádzkach s vlastnými skladmi, ako sú napr. čerpacie stanice s predajom aj iného tovaru ako pohonných hmôt. Budeme sa zaoberať práve týmto tovarom bez pohonných hmôt. Tieto prevádzky sú rozmiestnené heterogénne vzhľadom na rôznorodosť krajiny. Každá prevádzka, resp. čerpacia stanica bude do systému vkladať svoje požiadavky, s ktorými budeme pracovať. Teda tieto požiadavky sú pre nás exogénne zadané. Toto vkladanie požiadaviek prevádzkou nebude vykonávané v presne určenom čase, to znamená, že pridávanie požiadaviek do systému bude mať náhodný charakter.

My sa zameriame na optimalizáciu prepráv zo skladov s prebytkom tovaru do skladov s nedostatkom určitého tovaru. Všetky vzniknuté požiadavky sa budú v systéme zhromažďovať až do okamihu optimalizácie prepravy tovarov medzi prevádzkami, resp. medzi prevádzkami a externým dodávateľom. Všetkých externých dodávateľov stotožníme s jedným dodávateľom a umiestnime ho pre zjednodušenie do depa (miesto, kde sa zhromažďujú – parkujú prepravné vozidlá) a množstvo tovaru, ktoré je možné odtiaľ doviesť, nie je ohraničené.

Teda vstupom bude určitý súbor tovarov, kapacita prepravného vozidla, náklady na prepravu tovaru medzi jednotlivými prevádzkami, ktoré môžu reprezentovať vzdialenosti medzi prevádzkami.

Úlohy sú dve:

Prvou úlohou je uspokojiť požiadavky jednotlivých prevádzok pri minimalizácii nákladov na prepravu so zahrnutím externého dodávateľa.

Druhou úlohou je vybilancovať požiadavky jednotlivých prevádzok pri minimalizácii nákladov na prepravu bez externého dodávateľa. Inými slovami presunúť tovar z prevádzok, ktoré ho majú nazvyš, do prevádzok, ktoré ho majú nedostatok.

Obe úlohy sa od seba líšia len skutočnosťou, či je externý dodávateľ zahrnutý alebo nie do procesu optimalizácie. My budeme uvažovať ekvivalentnú skutočnosť, že či bude externý dodávateľ ponúkať tovar alebo nie.

Je zrejmé, že proces uspokojovania požiadaviek skladov ovplyvňuje množstvo faktorov. A to napr. doba spotreby tovaru, veľkosť tovaru, resp. jeho hmotnosť (t.j. maximálny počet kusov tovaru, ktorý bude môcť byť prepravovaný vrámci jedného transportu) a iné. Cieľom nebude zahrnúť čo najviac ovplyvňujúcich faktorov do modelu, ale vytvoriť model tak, aby čo najvernejšie opisoval skutočnosť pri relatívne jednoduchých výpočtoch a pokúsiť sa o jeho riešenie.

2 Základné pojmy

Skôr než pristúpime k formulácii a klasifikácii jednotlivých úloh, zavedieme si niekoľko základných pojmov označujúcich objekty a operácie dopravného procesu, ktorého jednotlivé prípady budeme rozoberať. Východzím pojmom bude **dopravný element**, čím rozumieme najdrobnejší objekt premiestňovania z hľadiska rozlišovacej úrovne skúmaného dopravného systému. Ak teda popisujeme napr. prepravu kontajnerov, je dopravným elementom kontajner. Pri rozvoze mlieka, kde sú vrecká s mliekom dodávané dodávateľovi vždy len v prepravkách, považujeme za dopravný element jednu prepravku s obsahom. Ak študujeme rozvoz technického plynu, kde zákazníkovi môže byť dodaný ľubovolný počet bômb s technickým plynom, označíme ako dopravný element každú bombu. Pokiaľ však budú bomby v priebehu dopravy umiestnené na paletách a budú tak i dodávané zákazníkovi, môžeme za dopravný element považovať celú paletu. Podobne pri preprave cementu vo vreciach môže byť podľa okolností dopravným elementom vrece cementu, ale aj celá paleta naložená vrecami s cementom.

V samotnej dopravnej sieti, v ktorej bude prebiehať premiestňovanie dopravných elementov, rozlišujeme dva druhy objektov. Je to jednak **uzol** siete, v ktorom sú vykonávané operácie s dopravnými elementmi, ako je skladanie alebo nakladanie, a ďalej **úsek** siete, čo je spojenie medzi dvoma uzlami, nutné k premiestneniu elementov medzi týmito dvoma uzlami.

Za uzly dopravnej siete pri rozvoze potravín, považujeme sklady, z ktorých je rozvoz uskutočňovaný, a predajne, ktoré sú behom rozvozu zásobované. Úsekom dopravnej siete tu odpovedajú cesty spojujúce predajne a sklady. Rozlišujeme úseky **orientované** a **neorientované**. Orientované úseky umožňujú prepravu dopravných elementov iba v jednom smere. Za príklad orientovaného úseku môžeme považovať jednosmernú cestu. Neorientované úseky umožňujú prepravu v oboch možných smeroch, ako napr. dvojsmerná cesta pri úlohách rozvozu.

Orientovaný úsek je bežne zadávaný ako usporiadaná dvojica uzlov, z ktorých prvý odpovedá miestu, odkiaľ sú dopravné elementy odvážané, a druhý odpovedá miestu, kam sú po danom úseku dopravné elementy dovážané. Ak označíme prvý uzol u a druhý ako v, potom usporiadanú dvojicu budeme značiť $<u, v>$, čo v prípadoch, keď z u do v nevedie viac než jeden orientovaný úsek, môže slúžiť ako označenie orientovaného úseku.

Neorientované úseky bývajú často zadávané ako neusporiadaná dvojica uzlov, ktoré sú príslušným úsekom spojené. V tomto prípade môžeme, za predpokladu, že uzly u a v sú

spojené jedným neorientovaným úsekom, tento úsek označiť symbolom $\{u, v\}$. Úsekom i uzlom dopravnej siete môžu byť v konkrétnych prípadoch priradené reálne čísla alebo i vektor reálnych čísel, vyjadrujúcich kvantitatívne ohodnotenie veličín, ktorými úsek alebo uzol ovplyvňuje dopravné pomery v sieti. Ohodnotenie úseku môže napríklad vyjadrovať vzdialenosť koncových bodov úseku, spotrebu pohonných hmôt na prejdenie príslušnej komunikácie vozidlom istého typu, alebo môže ohodnotenie predstavovať priemerné náklady na prepravu jedného dopravného elementu po tomto úseku. Naproti tomu, ohodnotenie uzlu môže vyjadrovať počet dopravných elementov, ktoré majú byť do uzla privezené z iných uzlov siete, alebo dobu potrebnú na uskutočnenie prepravných operácií v danom uzle, prípadne, ak je uzlu priradená dvojica reálnych čísel, môžeme týmto spôsobom vyjadriť obdobie, v ktorom môžu byť uspokojené požiadavky uzlu. V prípade zásobovania môže takýto interval vyjadrovať pracovnú dobu zamestnancov predajne a súčasťou úlohy môže byť požiadavka, aby potraviny boli dodané v časovom úseku vymedzením spomenutou dvojicou čísel.

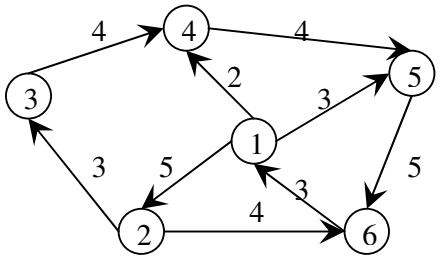
Iný dôležitý pojem, ktorý budeme používať, je **cesta** v dopravnej sieti. Cestou v dopravnej sieti rozumieme postupnosť uzlov a úsekov, začínajúcu a končiacu uzlom, v ktorej sa uzly a úseky striedajú tak, že medzi dvoma uzlami je umiestnený úsek, ktorý tieto uzly spojuje, a kde sú všetky úseky a uzly navzájom rôzne. Teraz už na základe uvedených pojmov môžeme definovať dopravnú sieť.

Dopravná sieť je daná množinou uzlov (vrcholov) V a množinou úsekov (hrán) H, kde pre každú dvojicu uzlov existuje aspoň jedna cesta, ktorej počiatok a koniec je zhodný s danými uzlami.

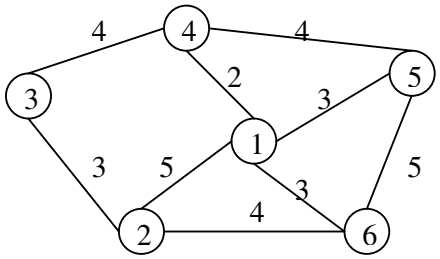
Dopravnú sieť môžeme znázorniť pomocou súvislého grafu, ktorého vrcholy budú odpovedať uzlom a hrany úsekom dopravnej siete. Ak je úsek orientovaný, znázorníme orientáciu príslušnej hrany v nákrese šípkou ukazujúcou smer orientácie. Znázornenie dopravnej siete pomocou grafu je ukázané na Obr 1, Obr 2 a Obr 3.

Podobne ako sú rozlíšené grafy, môžeme rozlišovať i dopravné siete ako orientované, neorientované a zmiešané. Dopravnú sieť nazveme orientovanou, ak sú orientované všetky jej úseky (Obr 1).

Pokiaľ všetky úseky dopravnej siete budú obojsmerné, budeme hovoriť o neorientovanej dopravnej sieti. Príklad neorientovanej dopravnej siete je uvedený na Obr 2.



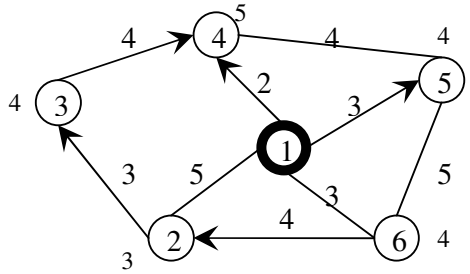
Obr 1. Orientovaná dopravná sieť. Uzly sú označené krúžkami a úseky šípkami



Obr 2. Neorientovaná dopravná sieť, v ktorej môžu byť úseky (vyznačené ako úsečky) prechádzané v oboch smeroch

Zmiešaná dopravná sieť je sieť, ktorá obsahuje orientované aj neorientované úseky (Obr 3). V istých dopravných sieťach je potrebné zdôrazniť dôležitosť niektorých uzlov v skúmanom dopravnom procese. Ako príklad takýchto uzlov uvedieme centrálny sklad pri úlohe rozvozu potravín a pod. Tieto uzly zvláštneho významu budeme nazývať **strediskami** dopravnej siete.

Na Obr 3 je znázornená dopravná sieť so strediskom v uzle 1. Znázornená sieť môže byť považovaná za ukážku dopravnej siete rozvozu, v ktorej je potrebné zo strediska 1 dopraviť odberateľom 2, 3, 4, 5 a 6 po poradí 3, 4, 5, 4 a 4 jednotiek tovaru.



Obr 3. Zmiešaná dopravná sieť so strediskom v uzle 1. Úseky siete sú ohodnotené svojou dĺžkou a ohodnotenie uzlov vyjadruje požadovaný počet dopravných elementov

Ohodnotenie uzlov na Obr 3 zodpovedá požiadavkám na množstvo tovaru alebo tiež počtu dopravných elementov, ktoré majú byť do príslušného uzla privezené; ohodnotenie úsekov môže vyjadrovať napr. dĺžky cestných úsekov medzi uzlami.

Zaoberajme sa ďalej pojmami spojenými so spôsobom premiestňovania dopravných elementov.

V máloktorej dopravnej sieti je dopravný element schopný prepravovať sa po úsekov bez nejakých prostriedkov, ako je napr. loď, automobil, vlak a iné. Okrem dopravných prostriedkov môžu byť k preprave jednotlivých prípadoch vyžadované posádka lode, vodič automobilu, strojvodca a takisto dôležité vybavenie, ako napr. poštový vak, paleta, vagón.

Aby nebolo potrebné v jednotlivých úlohách špecifikovať všetky uvedené pomocné objekty, ktoré pri riešení úloh nie je potrebné konkretizovať, zavedieme pre ne názov „**náležitosť**“. Presnejšie náležitosťou nazveme všetky prostriedky potrebné pre realizovanie prepravy elementov, ktoré sú po úsekov dopravnej siete premiestňované s elementmi.

Náležitosť spolu s dopravnými elementmi, ktoré prepravuje, budeme nazývať **kompletom**. Pokiaľ ale budeme študovať dopravnú sieť konkrétneho typu, väčšinou cestnú sieť, budeme miesto abstraktného pojmu „náležitosť“ používať prirodzenejší názov „dopravný prostriedok“ alebo „vozidlo,“ čím budeme pochopiteľne rozumieť vozidlo i s príslušnou posádkou a všetkým ostatným, čo je k preprave potrebné.

V súvislosti s prepravou elementu je potrebné špecifikovať úplne alebo čiastočne uzol, do ktorého má byť element dopravený. Ak je pre daný dopravný element určený konkrétny uzol do ktorého má byť element dopravený, hovoríme, že elementu je určená **pevná adresa**.

Pokiaľ je pre dopravný element určená iba množina uzlov, do ktorých môže byť element dopravený, hovoríme o **voľnej adrese**. Pre riešenie niektorých prepravných úloh nie je nutné poznať adresy dopravných elementov, ale miesto toho je úloha zadaná celkovými objemami, hmotnosťou alebo celkovými počtami dopravných elementov, ktoré je potrebné prepraviť medzi jednotlivými uzlami.

V takých prípadoch úlohu zadávame pomocou **množiny prepráv**, kde prepravou rozumieme usporiadanú trojicu $<u, v, q>$, v ktorej jednotlivé prvky majú nasledujúci význam; u je uzol, z ktorého treba prepraviť do uzlu v celkom q dopravných elementov.

Zatiaľ, čo miesta určenia dopravných elementov alebo jednotlivé prepravy bývajú vo väčšine úloh na dopravných sieťach súčasťou zadania úlohy, vlastný pohyb náležitostí pri prepravnej práci je treba určiť riešením príslušnej úlohy.

Zaveďme si v súvislosti s pohybom náležitostí v dopravných sieťach niekoľko potrebných pojmov.

Trasou náležitostí nazveme striedavú postupnosť uzlov a úsekov siete začínajúcej a končiacej uzlom, kde medzi dvoma uzlami stojí v postupnosti úsek, po ktorom je možné sa premiestniť do druhého uzlu. V trase náležitostí sa uzly môžu opakovať. Pomocou trasy budeme popisovať pohyb náležitostí, prípadne dopravného prostriedku v sieti. Príslušný proces tvorby alebo určenia trasy budeme ďalej nazývať **trasovaním**. V „jazyku“ teórie grafov zodpovedá pojmu trasa názov sled alebo polosled [1].

Okružnou jazdou v dopravnej sieti rozumieme trasu, v ktorej je počiatočný a koncový uzol zhodný a v ktorej sa žiadny iný uzol ani úsek neopakuje. V teórii grafov je okružná jazda nazývaná cyklom alebo polocyklom [1].

Podobne, ako bolo zavedené ohodnotenie úsekov a uzlov dopravnej siete, zavádzame ohodnotenie trasy náležitostí. Podľa formulácie úlohy môžeme za ohodnotenie trasy považovať súčet ohodnotenia všetkých úsekov alebo súčet ohodnotenia všetkých úsekov a uzlov vytvárajúcich trasu. Pomocou prvého ohodnotenia je možné vyjadriť napr. celkovú dĺžku trasy a pomocou druhého ohodnotenia dobu potrebnú na realizáciu trasy. Týmito dvoma príkladmi ani zďaleka nevyčerpávame všetky spôsoby ohodnotenia trasy. Spoločným znakom väčšiny bežne používaných ohodnotení trasy je závislosť príslušnej hodnoty na ohodnotených uzloch; je potrebné dodať, že často ide o lineárnu závislosť.

3 Klasifikácia prepravných úloh

Už aj z jednoduchších príkladov [3] vidieť, že obtiažnosť určenia optimálnej alebo aspoň suboptimálnej trasy dopravného prostriedku pri realizácii prepráv závisí na vzájomnom pomere kapacity dopravného prostriedku a na strednej veľkosti prepravovaných nákladov. Tento vzťah nám poslúži pre najhrubšie rozlíšenie prepravných úloh. Predpokladajme, že ľubovolná úloha prepráv je charakterizovaná prepravami typu <*u*, *v*, *q*> a kapacitou vozidla *K*. Ak považujeme *q* za strednú veľkosť prepravovaného nákladu, potom môžeme rozlišovať prípady [3]:

1. q > K trasovanie hromadných prepráv
2. q ~ K trasovanie kombinovaných prepráv
3. q < K trasovanie prepráv kusových zásielok

Čím je stredná veľkosť prepravovaného nákladu vzhľadom ku kapacite menšia, tým je určenie efektívnej trasy obtiažnejšie. V nasledujúcom texte sa zameriame na najmenej prepracovanú problematiku trasovania, a to na prípady trasovania prepráv kusových zásielok. Pretože ide o otvorenú problematiku, existuje už v súčasnej dobe už väčší počet metód efektívne riešiace špeciálne úlohy. Pre lepšiu orientáciu sformulujeme v nasledujúcom textu niektoré zo skôr riešených klasických úloh, ktoré neskôr využijeme.

Dopravná úloha

Táto úloha nie je typickým príkladom trasovania prepráv kusových zásielok spomenutých vyššie, ale vo väčšine príkladoch okružných jázd, pri ktorých majú elementy voľnú adresu, to znamená, že nie je im pevne priradený konečný uzol, sa dopravná úloha používa na transformáciu úlohy okružných jázd na úlohu s pevnou adresou, ktorá má už menšiu zložitosť – náročnosť výpočtu.

Sformulujme si teda dopravnú úlohu.

Nech je daná orientovaná dopravná sieť *G*, v ktorej každý úsek *ij* má predpísanú kapacitu b_{ij} a cenu c_{ij} za jednotku toku. Nech *G* má vstupy $s_1,\ldots,s_p$ a výstupy $t_1,\ldots,t_q$ s predpísanými požiadavkami $a_1,\ldots,a_p$ resp. $b_1,\ldots,b_q$. Úlohou je prepraviť dopravný element zo vstupov na výstupy v požadovaných množstvách a pri minimálnych nákladoch. V teórii

grafov to znamená [1], že v G treba nájsť tok x zo vstupov do výstupov takých, že čistý odtok $f_{s_i}^{+}(x) = a_i$ a čistý prítok $f_{t_j}^{-}(x) = b_j$ a pritom $\sum c_{ij}x_{ij}$ je minimálne.

Takáto úloha sa teda nazýva dopravná. (Iné názvy: sieťová dopravná úloha, úloha dopravy okľukou.)

Medzi najznámejšie špeciálne dopravné úlohy patrí tzv. Hitchcockova dopravná úloha (iný názov: klasická dopravná úloha). Hitchcock (1941) formuloval a riešil dopravnú úlohu, kde G je bipartitný graf (dopravná sieť) s množinou uzlov $V(G) = S \cup T$ a množinou úsekov $E(G) = \{s_i t_j \mid s_i \in S, t_j \in T\}$ a všetky kapacity úsekov sú nekonečné. Pre túto špeciálnu úlohu existuje mnoho algoritmov [1] [9].

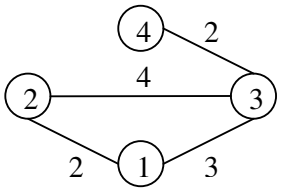
Úloha obchodného cestujúceho

Úloha obchodného cestujúceho sa vyskytuje často pri určovaní trás dopravných prostriedkov a to v prípadoch, keď kapacita náležitosti je nekonečná, alebo ako čiastkové riešenie komplikovanejšieho algoritmu. Úloha je formulovaná nasledovne:

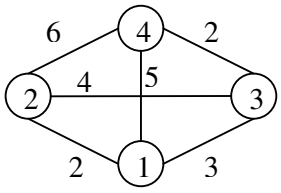
Je daná dopravná sieť s jedným strediskom a s jednou náležitosťou. V tejto dopravnej sieti je potrebné nájsť trasu náležitosti takú, aby začínala a končila v stredisku, prechádzala každým uzlom v sieti práve jedenkrát a aby jej ohodnotenie, dané súčtom ohodnotení použitých úsekov, bolo minimálne.

Zúplnenie dopravnej siete.

Pri formulácii praktických úloh existuje požiadavka, aby trasa prechádzala každým uzlom práve jedenkrát; často je nahradzovaná požiadavkou, aby sa náležitosť realizujúca trasu v každom uzle „zastavila“ práve jedenkrát. Rozdiel medzi pôvodnou a touto formuláciou demonštruje dopravná sieť na Obr 4, kde by podľa pôvodnej formulácie neexistovalo žiadne riešenie, pretože trasa, ktorá prechádza uzlom 4, musí nutne dvakrát prechádzať uzlom 3. Ak však použijeme druhú formuláciu, potom z definície dopravnej siete nejaké riešenie úlohy obchodného cestujúceho musí existovať. Uvedené doplnenie je však zbytočné v prípade, keď dopravná sieť je úplná, t.j. každé dva uzly sú spojené buď neorientovaným úsekom alebo dvoma orientovanými úsekmi s opačnou orientáciou.



Obr 4. Príklad dopravnej siete



Obr 5. Zúplnenie doprevnej sieťe z Obr 4.

Pri riešení úlohy obchodného cestujúceho a podobných úloh často používame namiesto zadanej dopravnej siete jej zúplnenie, t.j. náhradnú dopravnú sieť vytvorenú tak, že každé dva uzly pôvodnej siete spojíme úsekom, podľa druhu pôvodnej siete orientovaným alebo neorientovaným, ktorého ohodnotenie bude rovné minimálnemu ohodnoteniu cesty spojujúcej uzly v pôvodnej sieti. Zúplnenie dopravnej siete z Obr 4 je ukázané na Obr 5.

Úloha okružných jázd

Je daná dopravná sieť s jedným strediskom a jednou náležitosťou, ktorá môže súčasne prepravovať K dopravných elementov. Každý uzol i s výnimkou strediska požaduje dovezenie q_i dopravných elementov zo strediska, kde $q_i \leq K$.

Úlohou je zostaviť trasu náležitosti o minimálnom hodnotení tak, aby každým uzlom siete (okrem strediska) prechádzala práve jeden krát a aby súčet požiadaviek všetkých uzlov nachádzajúcich sa v spoločnej okružnej jazde zostavenej trasy bol nanajvýš rovný K.

V literatúre sa často ako modelový príklad uvádza príklad rozvozu sódoviek [3].

Úloha okružných jázd čiastočne pripomína úlohu viacnásobného obchodného cestujúceho. Je potrebné si ale uvedomiť, že v úlohe okružných jázd nie je počet okružných jázd zadaný. Je tu však navyše obmedzená kapacita náležitostí, ktorú formulácia úlohy viacnásobného cestujúceho nepozná. Podobne ako pri mnohých iných formuláciách i tu existujú mnohé modifikácie. Jednou z nich je zmena počtu náležitostí. Pokiaľ neexistuje pri formulácii úlohy viazanie s časovými podmienkami na dobu trvania okružnej jazdy alebo s podmienkami na to, aby do daného uzlu prišli v určenom časovom rozmedzí, je jedno, či je predpokladané zabezpečovanie všetkých okružných jázd, z ktorých sa výsledná trasa skladá, jednou alebo viacerými náležitosťami. Pokiaľ ale v úlohe uvažujeme obmedzený počet

náležitostí a maximálnu dobu, v ktorej tieto náležitosti môžu byť v prevádzke, dostávame netriviálnu modifikáciu pôvodnej úlohy.

Ďalšie možné zovšeobecnenie získame predpokladom, že každá použitá náležitosť má všeobecne inú kapacitu. Úlohu podstatne zložitejšiu dostaneme, ak budeme predpokladať zabezpečenie požiadaviek uzlov z viacerých stredísk a pod.

Zovšeobecnená úloha rozvozu

Je daná dopravná sieť s jedným strediskom a jednou alebo viacerými náležitosťami o kapacite K dopravných elementov. S dopravnou sieťou je zadaná množina prepráv $\{< u_i, v_i, q_i > | \, i = 1,2,\ldots,p\}$, kde $q_i \leq K$, $i = 1,2,\ldots,p$.

Našou úlohou je nájsť trasy náležitostí o minimálnom celkovom ohodnotení tak, aby boli realizované všetky prepravy a aby každá preprava $\{u_i, v_i, q_i\}$ bola uskutočnená pri jednej návšteve uzla u_i a v_i.

I túto úlohu je možné modifikovať použitými spôsobmi: uvedieme tu iba jednu modifikáciu, ktorá sa v skôr uvedených úlohách nevyskytla. Je to prípad, kedy časť alebo celá množina prepráv nie je určená množinou trojíc, ktoré vlastne každému elementu z dávky q_i priradzuje pevnú adresu v_i, ale pre danú množinu zameniteľných dopravných elementov sú určené niektoré uzly ako zdroje s istou kapacitou a iné uzly ako spotrebitelia s istou požiadavkou.

V tomto prípade úloha okrem určenia trás náležitostí vyžaduje aj priradenie pevných adries niektorým elementom.

4 Optimálne a suboptimálne metódy riešenia

4.1 Klasifikácia metód trasovania

V súčasnej dobe je možné metódy trasovania rozdeliť do 7 skupín podľa použitého princípu. Získavame tak nasledujúce delenie:

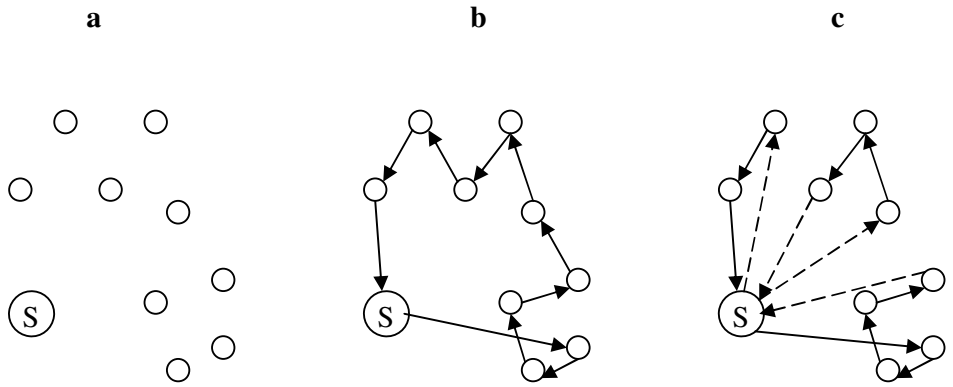
Metódy primárneho zhlukovania (cluster first – route second)

Tieto metódy poskytujú prevažne iba približné riešenie. Používajú sa pre riešenie úloh trasovania v rozsiahlych dopravných sieťach, v ktorých množinu uzlov vhodným spôsobom rozloží na podmnožiny, v ktorých potom rieši jednoduchšie úlohy trasovania, ako je napr. úloha obchodného cestujúceho.

Príkladom algoritmu z tejto skupiny je sweep algoritmus a iné dekompozičné algoritmy [9].

Metódy tvorby primárnej trasy (route first – cluster second)

Do tejto skupiny patria metódy využívajúce opačného postupu než metódy predchádzajúcej skupiny. Pri riešení úlohy je najprv vytvorená primárna trasa prechádzajúca všetkými uzlami, ktorá ale nespĺňa požiadavky prípustnosti, ako napr. požiadavku na maximálnu kapacitu dopravného prostriedku alebo na maximálnu dĺžku jednotlivých okružných jázd. Táto primárna trasa je pri ďalšom spracovávaní delená a upravovaná na okružné jazdy, spĺňajúce podmienky prípustnosti. Schématický postup znázorňujú po rade častí Obr 6.



Obr 6. Princíp metódy tvorby primárnej trasy a jej dekomponovanie na jednotlivé okružné jazdy

Metódy výhodnostných koeficientov a vkladania (savings – insertion)

Algoritmy realizujúce túto skupinu metód väčšinou vylepšujú východziu trasu pomocou operácií, ktorými vznikne nová prípustná trasa s menším celkovým ohodnotením.

Do tejto skupiny patrí Clarke-Wrightova metóda [9], [14], kde vylepšujúcou operáciou je spojenie dvoch okružných jázd, ďalej algoritmus inverzií reťazcov a algoritmus vkladania reťazcov [3], kde v prvom prípade je zlepšujúcou operáciou inverzia a v druhom premiestnenie reťazca.

Medzi metódy tejto skupiny zaraďujeme napr. algoritmy najbližšieho a najvýhodnejšieho suseda, algoritmy postupného zväčšovania a algoritmy zväčšovania o najvýhodnejší alebo najmenej výhodný uzol [1], [7].

Metódy zlepšovania výmenou (improvement and exchange)

Metódy tohto typu vylepšujú v jednotlivých krokoch doterajšie riešenie tým, že z neho vylúči niektorý z nevhodných úsekov a nahradia ho výhodnejším. Potom uskutočnia následné výmeny úsekov tak, aby nové riešenie s menším ohodnotením bolo prípustné. Predstaviteľom tejto skupiny je napr. Lin – Kernighanova metóda [15].

Metódy založené na matematickom programovaní (mathematical programming approach)

Heuristické metódy využívajúce postupov matematického programovania vychádzajú z úplného matematického modelu danej prepranej úlohy, ale neriešia ho celý, iba model

dekomponujú a riešia po častiach, ktoré nie sú príliš rozsiahle. Pri tomto prístupe najrozsiahlejšia časť podmienok modelu zodpovedá úlohe obchodného cestujúceho.

Do uvedenej skupiny môžeme zaradiť metódu riešenia úlohy trasovania vo viacstrediskovej sieti a čiastočne dekompozičnú metódu.

Interaktívne metódy

Tieto metódy sú založené na priamej spolupráce človeka s počítačom pri konštrukcii trás riešenej úlohy. Počítač „ponúka“ kooperujúcemu riešiteľovi trasu získanú obvykle niektorou z približných metód. Riešiteľ navrhuje prípadne zmeny v trase, vyplývajúce z dodatočných podmienok, na ktoré nebol dôraz v programe počítača, alebo ktoré môžu vylepšiť ohodnotenie trasy spôsobom, ktorý nie je použitým programom realizovateľný.

Týmto spôsobom je možné do riešenia vnášať prvky intuície, ktorý mnohokrát vedie k zlepšeniu, ktorého dosiahnutie prostredníctvom počítača by bolo neúnosne náročné na dobu výpočtu i na veľkosť pamäte počítača.

Presné metódy

Do tejto skupiny patria známe metódy, založené na princípe vetvenia a hraníc 36, alebo metódy využívajúce princíp rezných nadrovín, známe z celočíselného programovania. Aj napriek tomu, že vzhľadom k NP – úplnosti väčšiny úloh trasovania a vzhľadom k značnej časovej náročnosti skôr vyvinutých optimalizačných metód, tento prístup ustúpil v 70. až 80. rokoch do pozadia, nové publikácie z tejto oblasti ukazujú, že možnosti týchto metód značne prekračujú medze, ktoré boli považované za neprekonateľné.

4.2 Matematické modely

V tejto podkapitole si ukážeme postupy, ktorými je možné vytvoriť matematické modely väčšiny klasických úloh drobných prepráv a previesť ich tak na pôvodné úlohy matematického programovania.

Ako prvý sformulujeme model pre Hitchcockovu dopravnú úlohu. Nech G je bipartitný graf (dopravná sieť) s množinou uzlov $V(G) = S \cup T$ a množinou úsekov $E(G) = \{s_i t_j | s_i \in S, t_j \in T\}$ a všetky kapacity úsekov sú nekonečné Označme p počet prvkov

množiny *S* a *q* počet prvkov množiny *T*. Pre túto špeciálnu úlohu existuje mnoho algoritmov, a neskôr uvedieme simplexovu metódu, ktorú vypracoval Dantzig (1951), ako špecializáciu primárnej simplexovej metódy známej z lineárneho programovania. neskorší výklad je založený na teórii duality :

Pre primárnu a duálnu úlohu platí:

a) (Slabá veta o dualite): Pre ľubovolné prípustné riešenia *x* a *u* je $c^T x \geq b^T u$.

b) (Silná veta o dualite): Ak má jedna z úloh optimálne riešenie, tak má aj druhá a optimálne hodnoty účelovej funkcie sa rovnajú ($c^T x^* = b^T u^*$).

c) (Podmienky ortogonality): Prípustné riešenia *x* a *u* sú optimálne $\Leftrightarrow$ $(u^T a_j - c_j)x_j = 0$ pre každé *j* = 1, ..., *n*, kde a_j je *j*-tý stĺpec matice *A*.

Dôkaz tvrdenia je uvedený napr. v [1].

Ďalej nech x_{ij} označuje množstvo prepravované z s_i do t_j, potom úlohu môžeme zapísať takto:

Minimalizovať $$\sum_{j=1}^{q}\sum_{i=1}^{p} c_{ij}x_{ij} \qquad (1)$$

za podmienok $$\sum_{i=1}^{p} x_{ij} = b_j \qquad j = 1, 2, \ldots, q \qquad (2)$$

$$\sum_{j=1}^{q} x_{ij} = a_i \qquad i = 1, 2, \ldots, p \qquad (3)$$

$$x_{ij} \geq 0 \qquad \begin{matrix} i = 1, 2, \ldots, p \\ j = 1, 2, \ldots, q \end{matrix} \qquad (4)$$

K ohraničeniam (2) priraďme premenné v_j a k (3) u_i. Potom duálna úloha je takáto:

Maximalizovať $$\left(\sum_{i=1}^{p} a_i u_i + \sum_{j=1}^{q} b_j v_j\right) \qquad (5)$$

za podmienok $$u_i + v_j \leq c_{ij} \qquad \begin{matrix} i = 1, 2, \ldots, p \\ j = 1, 2, \ldots, q \end{matrix} \qquad (6)$$

Prípustné riešenia x a (u, v) sú optimálne práve vtedy, keď sú optimálne podmienky ortogonality:

$$(u_i + v_j - c_{ij})x_{ij} = 0 \qquad \begin{matrix} i = 1, 2, \ldots, p \\ j = 1, 2, \ldots, q \end{matrix} \tag{7}$$

Prejdime teraz bližšie k úlohe drobných prepráv. Základným prvkom modelu úlohy drobných prepráv v dopravnej sieti je premenná, priradená úseku dopravnej siete, vyjadrujúca, či dopravný prostriedok daným úsekom prejde alebo neprejde. Označme túto premennú x_{ij}, kde i a j sú koncové uzly úseku. Pokiaľ v úlohe uvažujeme viacej dopravných prostriedkov, je potrebné rozlíšiť ktorý z nich použije úsek z uzlu i do j. V takom prípade priradíme každému dopravnému prostriedku k a každému úseku $<i, j>$ premennú x_{ij}^k. Pretože v obidvoch uvedených prípadoch potrebujeme rozlíšiť iba prípad, kedy vozidlo daný úsek prejde od opačného prípadu, vystačíme s hodnotami 1 a 0.

Ďalej budeme predpokladať, že všetky dopravné siete v tejto podkapitole sú úplné; spôsob akým sa dá úloha na všeobecnej dopravnej sieti previesť na úlohu na úplnej sieti, bol ukázaný v kapitole 3. Uvažujeme teda úplnú dopravnú sieť so strediskom nula a ostatnými uzlami 1, 2, ..., n. Nech c_{ij} je dĺžka úseku $<i, j>$; potom môžeme úlohu obchodného cestujúceho na tejto sieti formulovať takto;

Minimalizovať
$$\sum_{j=0}^{n}\sum_{i=0}^{n} c_{ij}x_{ij} \tag{8}$$

za podmienok
$$\sum_{i=0}^{n} x_{ij} = 1 \qquad j = 0,1, 2, \ldots, n \tag{9}$$

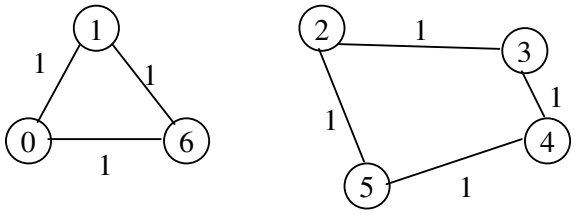
$$\sum_{j=0}^{n} x_{ij} = 1 \qquad i = 0,1, 2, \ldots, n \tag{10}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \le |S| - 1 \quad , \text{kde } S \subset \{1, 2, \ldots, n\} \tag{11}$$

a kde symbolom $|S|$ označujeme počet prvkov množiny S.

$$x_{ij} \in \{0,1\} \quad \begin{matrix} i = 0,1,\ldots,n \\ j = 0,1,\ldots,n \end{matrix} \tag{12}$$

V tomto modely výraz (8) vyjadruje celkovú dĺžku prejdených úsekov, podmienky typu (9) vyjadrujú, že do každého uzlu siete vozidlo príde práve jedenkrát, podmienky typu (10) vyjadrujú, že z každého uzlu siete vozidlo odíde práve jedenkrát a podmienky typu (11) zabezpečujú, že použité úseky budú tvoriť jedinú okružnú jazdu. V sústave podmienok (9), (10), (12) totiž vyhovuje i riešenie znázornené na Obr 7, ktoré zrejme nie je riešením úlohy obchodného cestujúceho. Podmienky typu (11) také trasy vylučujú. Riešenie z Obr 7 nebude napr. vyhovovať podmienke (11) pre $S = \{2, 3, 4, 5\}$, lebo ľavá strana nerovnosti bude mať hodnotu 4 a hodnota pravej strany bude 3.



Obr 7. Neprípustné riešenie úlohy obchodného cestujúceho, vyhovujúceho zjednodušenej sústave podmienok

Ak priradíme v dopravnej sieti každému uzlu i ($i = 1, 2, ..., n$) celočíselnú premennú u_i, môžeme podmienku (11) nahradiť podmienkou (13) a (14).

$$u_i - u_j + (n+1).x_{ij} \le n \qquad \begin{matrix} i = 1,2,...,n \\ j = 1,2,...,n \quad i \ne j \end{matrix} \tag{13}$$

u_i je celé nezáporné číslo pre $i = 1, ..., n$ (14)

Ekvivalentnosť podmienok (11) a podmienok (13) a (14) je dokázaná napr. v [16].

Pre symetrickú úlohu obchodného cestujúceho, ktorá zodpovedá úlohe obchodného cestujúceho v neorientovanej dopravnej sieti, kde je matica $\{c_{ij}\}$ ohodnotení úsekov symetrická, môžeme použiť jednoduchší model s približne polovičným počtom premenných. V pôvodnej formulácii sú totiž každému neorientovanému úseku <i, j> priradené dve premenné x_{ij} pre smer pre jazdu z i do j a x_{ji} pre opačný smer. Tento nedostatok odstránime tak, že priebežne očíslujeme úseky úplnej siete s uzlami 0, 1, ..., n číslami j = 1, 2, ...,

$(n+1)n/2$ a každému úseku priradíme premennú x_j s rovnakým významom ako predtým. Ohodnotenie úsekov budeme označovať c_j.

Ďalej pre potreby modelu definujme uzlovoúsekovú incidenčnú maticu $\{a_{ij}\}$, kde $a_{ij} = 1$, ak je uzol i jedným z koncových uzlov úseku j a $a_{ij} = 0$ v ostatných prípadoch.

Symetrickú úlohu obchodného cestujúceho môžeme sformulovať takto;

Minimalizovať $\sum_{j=1}^{m} c_j x_j$ (

za podmienok $\sum_{j=1}^{m} a_{ij} x_j = 2 \quad i = 0,1,....,n$

$\sum_{j \in H(S)} x_j \leq |S| - 1 \quad$, kde $S \subset \{1,2,...,n\}$

$x_j \in \{0,1\} \quad j = 1,...,m$

kde $m = (n + 1)n/2$ a symbolom $H(S)$ rozumieme množinu indexov tých úsekov siete, ktoré majú oba koncové uzly v množine S.

Uvažujme teraz úlohu s p dopravnými prostriedkami s kapacitami K_1, K_2, ... K_p. Nech je potrebné zo strediska 0 doviezť každému uzlu i , i = 1, 2, ..., n práve q_i dopravných elementov. Ak zavedieme okrem premenných x_{ij}^k a u_i ešte premenné y_i^k nadobúdajúce hodnoty 1 v prípade, že i-ty uzol bude navštívený k-tym dopravným prostriedkom a hodnoty 0 v iných prípadoch, môžeme úlohu okružných jázd formulovať takto:

Minimalizovať $\sum_{k=1}^{p} \sum_{j=0}^{n} \sum_{i=0}^{n} c_{ij} x_{ij}^k$ (15)

za podmienok $\sum_{i=1}^{n} q_i y_i^k \leq K_k \qquad k = 1, 2, ..., p$ (16)

$\sum_{k=1}^{p} y_i^k = 1 \qquad i = 1, 2, \ldots, n$ (17)

$$\sum_{k=1}^{p} y_0^k = p \tag{18}$$

$$\sum_{i=0}^{n} x_{ij}^k = y_j^k \qquad j = 0,1,2,\ldots,n \qquad k = 1,2,\ldots,p \tag{19}$$

$$\sum_{j=0}^{n} x_{ij}^k = y_i^k \qquad i = 0,1,2,\ldots,n \qquad k = 1,2,\ldots,p \tag{20}$$

$$u_i - u_j + (n+1).\sum_{k=1}^{p} x_{ij}^k \le n \qquad i, j = 1,2,\ldots,n \qquad i \ne j \tag{21}$$

$x_{ij}^k \in \{0,1\}; y_i^k \in \{0,1\}$; pre $k = 1, \ldots, p$, $i = 0, ..., n$

$j = 0, ..., n$ a u_i sú nezáporné čísla pre $i = 1, ..., n$. (22)

Pripomínam, že tento model zodpovedá prípadu, kedy každý dopravný prostriedok zabezpečí maximálne jednu okružnú jazdu. Uvedené vzťahy zabezpečujú splnenie nasledujúcich podmienok:

(16) zaisťuje, že súčet požiadaviek uzlov, ktoré budú obslúžené *k*-tym vozidlom, nepresiahne kapacitu tohto vozidla;

(17) zaisťuje, že požiadavky každého uzlu okrem strediska budú zabezpečené práve jedným vozidlom;

(18) vyjadruje skutočnosť, že všetky vozidlá „navštívia“ stredisko, nie je však vylúčené, že niektoré z vozidiel stredisko vôbec neopustí;

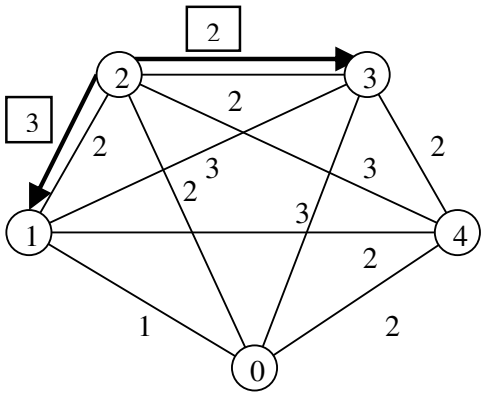
(19) zaisťuje, že do uzlu *j*, ktorý má byť navštívený *k*-tym vozidlom, toto vozidlo príde;

(20) zaisťuje, že z uzlu *i*, ktorý má byť navštívený *k*-tym vozidlom, toto vozidlo odíde;

(21) zaisťuje, že okružná jazda každého z použitých vozidiel prechádza strediskom

Nakoniec uvedieme matematický model zovšeobecnenej úlohy rozvozu, v ktorej požadujeme, aby súčet všetkých dopravných elementov, ktoré vozidlo v priebehu svojej jazdy prepraví, bol nanajvýš rovný kapacite vozidla. Poznamenávame tu, že tento predpoklad vylučuje niektoré riešenia pôvodnej úlohy, ale umožní nám zostaviť lineárny model.

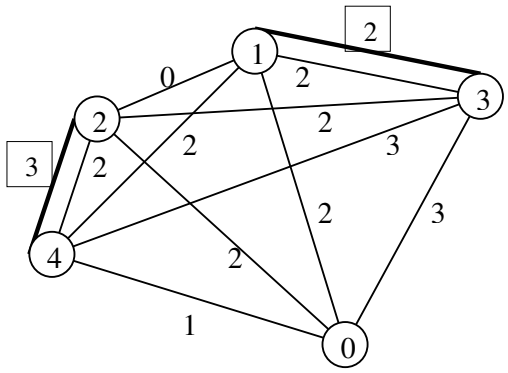
Predpokladáme teda úplnú dopravnú sieť s jedným strediskom a s r dopravnými prostriedkami s kapacitami K_1, K_2, ..., K_r. V nich je daná množina prepráv $P = \{< i_1, j_1, q_1 >, \ldots, < i_p, j_p, q_p >\}$ v tejto sieti a nech $\{c_{ij}\}$ sú dĺžky úsekov tejto siete. Príklad takejto siete je uvedený na Obr 8, kde prepravy sú vyznačené silnou čiarou a počty prepravovaných elementov sú uvedené vo štvorčekoch.



Obr 8. Zadanie zovšeobecnenej úlohy rozvozu. Čísla v štvorčekoch pri niektorých úsekoch vyjadrujú veľkosť nákladu, ktorého preprava je požadovaná

Pre ďalšiu prácu vytvoríme náhradnú dopravnú sieť so strediskom 0 a s ďalšími $2p$ uzlami, ktoré budú zodpovedať počiatočným a koncovým uzlom jednotlivých prepráv. Číslovanie uzlu zvolíme tak, aby v prípade, že počiatočnému uzlu prepravy priradíme nový index i, bol koncovému uzlu prepravy priradený index $i + p$.

Pri týchto úpravách môže nastať prípad, že v náhradnej sieti sa vyskytne niekoľko uzlov, ktorým v pôvodnej sieti zodpovedal jediný. Dĺžky úsekov spojujúcich v novej sieti uzly vytvorené z jedného pôvodného uzla položíme rovné nule. Dĺžky úsekov spojujúcich v novej sieti ostatné uzly budú rovné dĺžkam úsekov spojujúcich v pôvodnej sieti pôvodné uzly. Uzly, v ktorých v pôvodnej sieti nezačínala ani nekončila žiadna preprava do náhradnej siete s výnimkou strediska zahrnuté nebudú. Náhradnú sieť pre prepravy z Obr 8 je ukázaná na Obr 9.



Obr 9. Náhradná dopravná sieť vytvorená pre riešenie zovšeobecnenej úlohy rozvozu.

Uzol 1 je nahradený uzlom 1 a 2 novej siete; pôvodnému uzlu 3 zodpovedá uzol 3 a pôvodnému uzlu 2 zodpovedá v novej sieti uzol 4. Pôvodný uzol 4 do pôvodnej siete v žiadnej podobe neprešiel.

V náhradnej sieti s ohodnotením úsekov $\{c_{ij}\}$ teraz budeme riešiť úlohu rozvozu pre r dopravných prostriedkov a $2p$ uzlov, ktorých požiadavky na presun zo strediska sú pre $i = p + 1, ..., 2p$ nulové (ale navštívené byť tieto uzly musia) a požiadavky uzlov $i = 1, 2, ..., p$ zodpovedajú množstvám q_i, ktoré mali byť z pôvodných uzlov odvážané. Pri riešení uvedenej úlohy rozvozu musí sa naviac rešpektovať podmienka, že uzly i a $i + p$ musia byť navštívené rovnakým vozidlom, a to najprv uzol i a potom až uzol $i + p$. Ak zavedieme premenné x_{ij}^k, y_i^k a u_i rovnako ako vo formulácii úlohy okružných jázd, dostaneme nasledujúci matematický model úlohy:

Minimalizovať
$$\sum_{k=1}^{r}\sum_{j=0}^{2p}\sum_{i=0}^{2p} c_{ij}x_{ij}^k \tag{23}$$

za podmienok
$$\sum_{i=1}^{p} q_i y_i^k \le K_k \qquad k = 1, 2, ..., r \tag{24}$$

$$\sum_{k=1}^{r} y_i^k = 1 \qquad i = 1, 2, \ldots, 2p \tag{25}$$

$$\sum_{k=1}^{r} y_0^k = r \tag{26}$$

$$\sum_{i=0}^{2p} x_{ij}^k = y_j^k \qquad j = 0,1,2,\ldots,2p \qquad k = 1,2,\ldots,r \tag{27}$$

$$\sum_{j=0}^{2p} x_{ij}^k = y_i^k \qquad i = 0,1,2,\ldots,2p \qquad k = 1,2,\ldots,r \tag{28}$$

$$y_i^k = y_{i+p}^k \quad i = 1,\ldots,p \qquad k = 1,\ldots,r \tag{29}$$

$$u_i - u_j + (2p+1).\sum_{k=1}^{r} x_{ij}^k \le 2p \qquad i,j = 1,2,\ldots,2p \qquad i \ne j \tag{30}$$

$$u_i - u_{i+p} \le -1 \quad i = 1,2,\ldots,p \tag{31}$$

$x_{ij}^k \in \{0,1\}$; $y_i^k \in \{0,1\}$; pre $k = 1, \ldots, r$, $i = 0, ..., 2p$
$j = 0, ..., 2p$ a u_i sú nezáporné čísla pre $i = 1, ..., 2p$. (32)

Podmienka (24) zabezpečuje, že *k*-te vozidlo môže naložiť náklad vo všetkých uzloch, z ktorých má zabezpečovať prepravu (sú to uzly, pre ktoré $y_i^k = 1$), a zároveň nebude prekročená kapacita vozidla.

Podmienka (25) zabezpečuje, že požiadavka každého uzlu bude uspokojená jediným vozidlom.

Význam podmienok (26), (27), (28) a (30) je zhodný s významom podmienok (18), (19), (20) a (21).

Podmienky (29) zabezpečujú, že požiadavka uzlu *i* a *i* + *p* bude uspokojená rovnakým vozidlom a podmienky (31) zabezpečujú, že uzol *i* bude príslušným vozidlom navštívený skôr ako uzol *i* + *p*.

Ukázali sme teda spôsoby, ktorými sa dajú jednotlivé klasické rozvozné úlohy previesť na úlohy matematického programovania, konkrétne na úlohy celočíselného lineárneho programovania, pre ktoré existujú algoritmy na riešenie.

Tieto algoritmy sú ale nanešťastie pre väčší rozsah úloh značne náročné na dobu výpočtu, a tak ich priame využitie pre riešenie uvedených úloh celočíselného lineárneho programovania pri počte uzlov a prepráv, vyskytujúcich sa v praxi, je z technických dôvodov pri súčasnom stave dostupnej výpočtovej techniky nemožné.

Uvedené modely je však možné použiť pre dekompozíciu úloh trasovania a pre konštrukciu približných úloh, ktorých riešenie nám poskytne dolný odhad hodnoty účelovej funkcie optimálneho riešenia úlohy.

4.3 Metódy riešenia úlohy obchodného cestujúceho

Nájdenie optimálneho riešenia úlohy obchodného cestujúceho nie je vôbec jednoduché. Úloha totiž patrí do triedy NP – ťažkých problémov, teda nie je známy žiadny polynomiálny algoritmus na jej vyriešenie. Z týchto dôvodov sa pozornosť riešiteľov tejto úlohy zamerala na približné riešenie problému. Samozrejme toto riešenie by malo mať určité výhody ako kompenzáciu istej odchýlky od optimality. Touto výhodou je časová úspora a výpočtová jednoduchosť. Tieto podmienky spĺňajú aproximačné algoritmy (heuristiky) riešenia úlohy obchodného cestujúceho, ktoré zaručujú pomerne dobré výsledky v prijateľnom výpočtovom čase. A práve tieto heuristiky sú predmetom diplomovej prace Tibora Valenta [7]. Je to prehľad najznámejších heuristík v príjemnej forme, preto zdĺhavý popis týchto heuristík vynecháme.

4.4 Metódy riešenia zvozných a rozvozných úloh

V tejto kapitole si ukážeme základné princípy metód riešenia základných zvozných a rozvozných úloh, ktoré nám poskytnú istý základ pre nájdenie riešenia úlohy formulovanej v kapitole 1.

Uvedené metódy môžu byť v podstate rozdelené do dvoch skupín. V prvej budú metódy, v ktorých sú trasy dopravných prostriedkov budované priamo postupným pridávaním uzlov s priebežnou kontrolou prípustnosti príslušných zväčšovaní. V druhej skupine sa budú nachádzať metódy, ktoré úlohu zvážania a rozvážania dekomponujú na jednoduchšie podúlohy, ktorých každé riešenie spĺňa podmienky a je potrebné určiť len najlepšie riešenie podúloh z hľadiska stanoveného kritéria optimality.

Čisté zvozné alebo rozvozné úlohy, ktorých riešením sa budeme ďalej zaoberať, formulujeme v tvare:

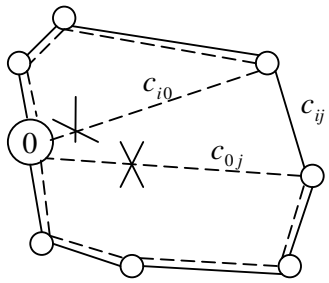
Je daná úplná neorientovaná dopravná sieť o $n + 1$ uzloch a $n(n + 1)/2$ úsekoch, ktorých ohodnotenie je dané maticou $\{c_{ij}\}$. Predpokladajme, že stredisko siete je označené číslom 0 a ostatné uzly postupne číslami 1, 2, …, n. Každý z uzlov $i = 1, ..., n$ požaduje dovezenie q_i dopravných elementov zo strediska, pričom k dispozícii je dopravný prostriedok o kapacite K.

V ďalšom texte sa obmedzíme iba na prípady, kedy platí $q_i \leq K$ pre $i = 1, ..., n$ a kde požiadavka každého prepravcu musí byť uspokojená jedinou návštevou dopravného prostriedku. Našou úlohou je nájsť takú množinu okružných jázd začínajúcich a končiacich v stredisku, aby celkové ohodnotenie týchto jázd bolo minimálne a aby bolo splnené obmedzenie dané kapacitou dopravného prostriedku, t.j. aby súčet požiadaviek uzlov prislúchajúcich k jednej okružnej jazde nepresiahol kapacitu K. Táto úloha môže byť doplnená podmienkami na maximálnu dobu trvania jednej okružnej jazdy, na maximálny počet uzlov uspokojených jednou okružnou jazdou a pod.

Za všetky algoritmy zaoberajúce sa touto problematikou uvedieme ten najznámejší. Pripomínam, že ku väčšine algoritmov, s ktorými sa stretávame v odbornej literatúre existujú mnohé modifikácie a zovšeobecnenia. Niektoré ďalšie algoritmy a ich zovšeobecnenia sú uvedené napríklad v [3].

Clarke-Wrightova metóda

Táto metóda [9], [14] je jedným z najrozšírenejších predstaviteľov prvej skupiny metód riešenia rozvozných úloh. Algoritmus Clarke-Wrightovej metódy začína spracovávať východzie prípustné, ale veľmi neefektívne riešenie, tvorené kyvadlovými jazdami tvaru stredisko – uzol – stredisko. V ďalších krokoch algoritmu z množiny okružných jázd vyberie také dve, ktoré je možné s ohľadom na kapacitu dopravného prostriedku spojiť a ich spojením sa celkové ohodnotenie celkových jázd najviac zmenší. Algoritmus končí svoju prácu vtedy, keď už žiadnym prípustným spojením dvoch jázd nie je možné dosiahnuť úsporu v ohodnotení trás.



Obr 10. Spojenie dvoch okružných jázd do jednej podľa Clarke-Wrightovej metódy

Algoritmus pripúšťa spojenie dvoch okružných jázd iba v uzloch, ktoré sú v pôvodných jazdách spojené úsekom so strediskom (Obr 10). Tým vzniká pri symetrickej dopravnej sieti nanajvýš $n(n - 1)/2$ možnosti spojenia a tento počet so zvyšovaním počtu uzlov v jednotlivých jazdách v priebehu práce algoritmu klesá. Úspora vzniknutá spojením uzlov i a j patriacich do dvoch rôznych okružných jázd, v ktorých sú uzly i a j spojené so strediskom, je vyjadrená vzorcom $c_{i0} + c_{0j} - c_{ij}$. Podrobnejšie je algoritmus popísaný nasledujúcimi bodmi:

1. Pre všetky usporiadané dvojice $\{i, j\}$ uzlov dopravnej siete, kde $i \neq j, \quad i, j \neq 0$, vypočítame koeficient $l_{ij} = c_{i0} + c_{0j} - c_{ij}$. Trasu 0, 1, 0, 2, 0, ..., 0, n, 0, ktorá je počiatočným riešením označíme symbolom T. Nakoniec definujme zobrazenie Q a F, kde zobrazenie Q je definované pre čísla r okružných jázd, že $Q(r)$ udáva súčet všetkých požiadaviek q_i, ktorých r-tá jazda uspokojuje. Zobrazenie F je definované pre všetky nestrediskové uzly dopravnej siete a priraďuje im číslo okružnej jazdy, do ktorej príslušný uzol patrí. Na začiatku definujeme $Q(r) = q_r$ a $F(i) = i$
2. Nájdeme najvyššie kladné l_{ij} a prejdeme k bodu 3. Pokiaľ také l_{ij} neexistuje, práca algoritmu končí a výsledné riešenie je určené postupnosťou T.
3. Ak platí nerovnosť $Q(F(i)) + Q(F(j)) \leq K$, položíme $l_{ij} = 0$ a uskutočníme zlúčenie jázd tak, ako je naznačené v Obr 10 Číslo novej jazdy r určíme ako $\min\{F(i), F(j)\}$. Položíme $Q(r) = Q(F(i)) + Q(F(j))$, pre všetky uzly k zo spojených jázd položíme $F(k) = r$, a pokiaľ spojením uzlov i a j niektorý z nich nebude i naďalej spojený so strediskom, položíme rovné nule všetky l_{uv}, kde buď u alebo v je rovné číslu takého uzla. Takisto vynulujeme koeficienty l_{uv}, kde uzly u aj v patria do novovzniknutej okružnej jazdy. Ak

neplatí nerovnosť, položíme iba $l_{ij} = 0$. V obidvoch prípadoch prejdeme nakoniec k bodu 2.

4.5 Metaheuristiky

Okrem uvedených metód sú v súčasnej dobe využívané heuristiky založené na metastratégiách označované ako Simulated Annealing, Tabu Search, genetické algoritmy, neurónové siete [4]. Všetky skôr uvedené minimalizačné heuristiky majú jednu nevýhodu a tou je neschopnosť opustiť lokálne minimum po tom, čo ho predchádzajúcimi krokmi dosiahnu.

Metaheuristiky sú také heuristicke prístupy, ktoré za istých okolností umožňujú opustiť lokálne minimum a prejsť postupnosťou iteračných krokov do iných častí množiny prípustných riešení, kde je nádej nájsť riešenie s lepšou hodnotou účelovej funkcie než malo nájdené lokálne minimum. Rovnako ako aj iné heuristicke metódy, metaheuristiky tiež nezaručujú nájdenie optimálneho riešenia úlohy.

Pre vysvetlenie práce metaheuristík zavedieme niekoľko pojmov. Metaheuristiky pracujú na diskrétnej množine X všetkých riešení danej úlohy. Pretože ide väčšinou o riešenie prípustné, obmedzíme sa ďalej na tento prípad. Podobne ako prosté primárne minimalizačné heuristiky i metaheuristiky prechádzajú v jednotlivých krokoch od prípustného riešenia k prípustnému riešeniu, pričom využívajú operácie využívané minimalizačnými heuristikami. Takými operáciami môže pri úlohách trasovania byť výmena alebo inverzia reťazcov a pri úlohách rajonizácie napríklad výmena dvoch zákazníkov medzi rajónmi.

Zoznam prípustných operácií je základnou charakteristikou metaheuristiky. Pomocou zoznamu prípustných operácii je definovaný pre metaheuristiku prípustný prechod medzi dvomi riešeniami úlohy. Napríklad, ak bude pri riešení úlohy obchodného cestujúceho množina povolených operácii tvorená iba inverziami reťazcov dĺžky dva, budeme prípustný prechod definovať medzi každými dvomi riešeniami, kde je možné získať jedno z druhého uskutočnením inverzie reťazcov dĺžky dva. Prípustné prechody na množine X vytvárajú topológiu danú sústavou okolí jednotlivých riešení. Okolie riešenia x^0 definujeme ako množinu $O(x^0) \subseteq X$, ktorej prvky sú práve všetky riešenia z X, ku ktorým existuje prechod z riešení x^0. Tu je potrebné si uvedomiť, že iná množina povolených operácii na množine X rovnakej úlohy vytvorí inú topológiu. Napríklad, keby medzi povolené operácie v úlohe

obchodného cestujúceho patrili nielen inverzie reťazcov dĺžky dva ale i tri, malo by okolie riešenia x^0 dvojnásobný počet prvkov než v pôvodne uvažovanom prípade.

5 Matematický model úlohy

Teraz ukážeme odvodenie matematického modelu prvej úlohy z kapitoly 1, čím prevedieme pôvodný problém na úlohu matematického programovania. Za základnú podmienku požadujeme, aby súčet všetkých dopravných elementov, ktoré vozidlo v priebehu svojej jazdy prepraví, bol nanajvýš rovný kapacite vozidla. Tento predpoklad vylučuje niektoré riešenia úlohy, ale na druhej strane nám umožní zostaviť lineárny model.

Predpokladáme teda úplnú dopravnú sieť s jedným strediskom a s r dopravnými prostriedkami s kapacitami K. V nej sa majú vykonať prepravy z množiny prepráv $P = \{< i_1, j_1, q_1 >, \ldots, < i_p, j_p, q_p >\}$ a nech $\{c_{ij}\}$ sú prepravné náklady úsekov tejto siete. Poznamenávam, že množina P nie je explicitne zadaná, ale jej prvky sú takisto ovplyvnené optimalizáciou. Teda túto množinu môžeme považovať za implicitne zadanú.

Pre ďalšiu prácu vytvoríme náhradnú dopravnú sieť so strediskom 0 a s ďalšími $2p$ uzlami, ktoré budú zodpovedať počiatočným a koncovým uzlom jednotlivých prepráv. Číslovanie uzlov zvolíme tak, aby v prípade, že počiatočnému uzlu prepravy priradíme nový index i, bol koncovému uzlu prepravy priradený index $i + p$.

Pri týchto úpravách môže nastať prípad, že v náhradnej sieti sa vyskytne niekoľko uzlov, ktorým v pôvodnej sieti zodpovedal jediný. Dĺžky úsekov spojujúcich v novej sieti uzly vytvorené z jedného pôvodného uzla položíme rovné nule. Dĺžky úsekov spojujúcich v novej sieti ostatné uzly budú rovné dĺžkam úsekov spojujúcich v pôvodnej sieti pôvodné uzly. Uzly, v ktorých v pôvodnej sieti nezačínala ani nekončila žiadna preprava, do náhradnej siete s výnimkou strediska zahrnuté nebudú.

V náhradnej sieti s ohodnotením úsekov $\{c_{ij}\}$ (pripomínam, že prepravy sú implicitne zadané) teraz budeme riešiť úlohu rozvozu pre jeden dopravný prostriedok a $2p$ uzlov, ktorých požiadavky na presun zo strediska sú pre $i = p + 1, \ldots, 2p$ nulové (ale navštívené byť tieto uzly musia) a požiadavky uzlov $i = 1, 2, \ldots, p$ zodpovedajú množstvám q_i, ktoré mali byť z pôvodných uzlov odvážané. Pri riešení uvedenej úlohy rozvozu musí sa naviac rešpektovať podmienka, že uzly i a $i + p$ musia byť navštívené v určenom poradí, a to najprv uzol i a potom až uzol $i + p$.

Zaveďme premennú, priradenú úseku dopravnej siete, vyjadrujúcu, či dopravný prostriedok daným úsekom prejde alebo neprejde. Teda priradíme každému dopravnému prostriedku k a každému úseku $<i, j>$ premennú x_{ij}^k. Pretože potrebujeme rozlíšiť iba prípad,

kedy vozidlo daný úsek prejde od opačného prípadu, bude táto premenná nadobúdať hodnoty 1 a 0. Ďalej označme y_i^k premennú nadobúdajúcu hodnoty 1 v prípade, že *i*-ty uzol bude navštívený *k*-tym dopravným prostriedkom, a hodnoty 0 v iných prípadoch a zaveďme premennú u_i rovnako ako vo formulácii úlohy okružných jázd, potom dostaneme nasledujúci matematický model úlohy:

Minimalizovať
$$\sum_{k=1}^{r}\sum_{j=0}^{2p}\sum_{i=0}^{2p} c_{ij}x_{ij}^k \tag{33}$$

za podmienok
$$\sum_{i=1}^{p} q_i y_i^k \le K \qquad k = 1, 2, ..., r \tag{34}$$

$$\sum_{k=1}^{r} y_i^k = 1 \qquad i = 1, 2, \ldots, 2p \tag{35}$$

$$\sum_{k=1}^{r} y_0^k = r \tag{36}$$

$$\sum_{i=0}^{2p} x_{ij}^k = y_j^k \qquad j = 0,1, 2, \ldots, 2p \quad k = 1, 2, \ldots, r \tag{37}$$

$$\sum_{j=0}^{2p} x_{ij}^k = y_i^k \qquad i = 0,1, 2, \ldots, 2p \quad k = 1, 2, \ldots, r \tag{38}$$

$$y_i^k = y_{i+p}^k \quad i = 1, \ldots, p \quad k = 1, \ldots, r \tag{39}$$

$$u_i - u_j + (2p+1).\sum_{k=1}^{r} x_{ij}^k \le 2p \qquad i, j = 1, 2, \ldots, 2p \quad i \ne j \tag{40}$$

$$u_i - u_{i+p} \le -1 \quad i = 1, 2, \ldots, p \tag{41}$$

$x_{ij}^k \in \{0,1\}$; $y_i^k \in \{0,1\}$; pre $k = 1, \ldots, r$, $i = 0, ..., 2p$
$j = 0, ..., 2p$ a u_i sú nezáporné čísla pre $i = 1, ..., 2p$. (42)

Podmienka (34) zabezpečuje, že *k*-te vozidlo môže naložiť náklad vo všetkých uzloch, z ktorých má zabezpečovať prepravu (sú to uzly, pre ktoré $y_i^k = 1$), a zároveň nebude prekročená kapacita vozidla.

Podmienka (35) zabezpečuje, že požiadavka každého uzlu bude uspokojená jediným vozidlom.

Podmienka (36) vyjadruje skutočnosť, že všetky vozidlá „navštívia“ stredisko, nie je však vylúčené, že niektoré z vozidiel stredisko vôbec neopustí.

Podmienka (37) zaisťuje, že do uzlu *j*, ktorý má byť navštívený *k*-tym vozidlom, toto vozidlo príde.

Podmienka (38) zaisťuje, že z uzlu *i*, ktorý má byť navštívený *k*-tym vozidlom, toto vozidlo odíde.

Podmienka (40) zaisťuje, že okružná jazda každého z použitých vozidiel prechádza strediskom.

Podmienky (39) zabezpečujú, že požiadavka uzlu *i* a *i* + *p* bude uspokojená rovnakým vozidlom a podmienky (41) zabezpečujú, že uzol *i* bude príslušným vozidlom navštívený skôr ako uzol *i* + *p*.

Sformulovali sme úlohu matematického programovania, konkrétne úlohu celočíselného lineárneho programovania, pre ktoré existujú algoritmy na riešenie.

Tieto algoritmy, podobne ako pri vzorových modeloch rozvozných úloh, sú ale nanešťastie pre väčší rozsah úloh značne náročné na dobu výpočtu, aj keď sme úlohu na začiatku zjednodušili na lineárny model, a tak ich priame využitie pre riešenie uvedených úloh celočíselného lineárneho programovania pri počte uzlov a prepráv, vyskytujúcich sa v praxi, je z technických dôvodov pri súčasnom stave výpočtovej techniky nemožné.

6 Riešenie úlohy

Pred samotnou optimalizáciu je potrebne ešte určiť, ktoré uzly budú vstupovať do optimalizácie medziprevádzkovej prepravy dopravných elementov.

6.1 Ovplyvňujúce faktory a externý dodávateľ

Do systému prepravy tovarov vstupuje množstvo faktorov, ako už sme spomenuli pri formulácii úlohy. Väčšinu ovplyvňujúcich faktorov môžeme premietnuť do ceny tovaru v tom zmysle, že ju znižujú. Typickým predstaviteľom týchto faktorov je dĺžka ostávajúcej záruky. Zrejme cena elementu s plnou zárukou nebude znížená, ale naopak, cena elementu s nulovou záručnou dobou bude nulová. Toto zníženie ceny je teda od nuly pri plnej záruke až po celú výšku hodnoty nového tovaru pri uplynutí záruky. Podobne je možné zahrnúť väčšinu faktorov do modelu. Teda črtá sa nám tu ďalší problém ako vyčísliť hodnotu tovaru, keď poznáme jeho cenu v prípade plnej záruky a uvažujeme niektoré ovplyvňujúce faktory. Na druhej strane sa ale na cenu tovaru môžeme pozerať ako na istú časť nákladov na dovezenie tovaru do uzla. Preto ďalej budeme uvažovať už s modifikovanými nákladmi prepravy c_{ij}. Preto jediný ovplyvňujúci faktor vstupujúci do algoritmu na optimalizáciu prepravy budú náklady c_{ij}.

V kapitole 1 sme boli postavení pre dve úlohy. Prvou úlohou bolo uspokojiť požiadavky jednotlivých prevádzok pri minimalizácii nákladov so zahrnutím externého dodávateľa a druhou úlohou bolo vybilancovať požiadavky jednotlivých prevádzok pri minimalizácii nákladov bez externého dodávateľa. Tieto dve úlohy sa veľmi podobajú až na prítomnosť externého dodávateľa. V nasledujúcom texte sa budeme zaoberať len úlohou s externým dodávateľom, ktorého stotožníme s depom dopravného prostriedku. Pri tejto úlohe stanovujeme ponuku externého dodávateľa pre všetky dopravné elementy na nekonečno. Jednoduchým nastavením ponuky externého dodávateľa na 0 pre všetky dopravné elementy prechádzame k druhej úlohe. Teda celkový algoritmus až na uvedenú skutočnosť bude ten istý.

Pred optimalizáciou je ešte potrebné „vyčistiť“ dopravnú sieť od uzlov, ktoré nemajú žiadne požiadavky a nie sú strediskom. Čiže v dopravnej sieti, s ktorou budeme ďalej uvažovať, sa nachádzajú len uzly s požiadavkami a stredisko.

6.2 Metóda optimalizácie prepravy

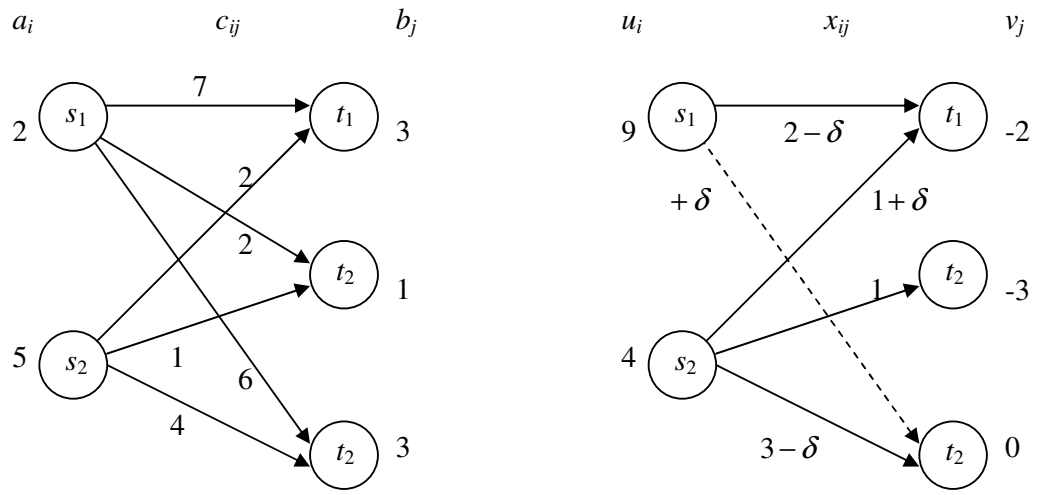
Keďže model úlohy nie je reálne možné riešiť ako úlohu celočíselného lineárneho programovania, použijeme na zistenie približného riešenia heuristicke metódy. Úloha formulovaná v kapitole 1 je pre elementy s voľnou adresou. Preto dekomponujeme úlohu na dve podúlohy. Z toho dôvodu celá optimalizácia prebehne v dvoch hlavných krokoch. Najprv sa priradia pevné adresy elementov. Takto vznikne niekoľko jázd typu stredisko – uzol i – uzol j – stredisko. V ďalšom kroku vytvoríme z týchto jázd jazdy okružne.

Krok 0: Vstupy

Vstupom nám budú požiadavky jednotlivých stredísk, kapacita K prepravného vozidla a náklady c_{ij} na prepravu tovaru medzi jednotlivými prevádzkami zahŕňajúce aj obstarávaciu cenu tovaru. Tieto náklady budeme považovať za exogénne zadané, i keď ich nie je jednoduché vypočítať. Určenie týchto nákladov sa robí štatisticky alebo pomocou istých odhadov. Existuje niekoľko prác zaoberajúcich sa touto problematikou a aj preto sa určovaniu týchto nákladov nebudeme ďalej venovať.

Krok 1: Dopravná úloha

Pre každý tovar zvlášť riešime Hitchcockovú dopravnú úlohu. Jej formulácia je uvedená v kapitole 3.



Obr 11. Príklad Hitchcockovej dopravnej úlohy; kostra a zmena riešenia

$\left((\hat{i},\hat{j})=(1,3),\delta=2,(\bar{i},\bar{j})=(1,1)\right)$.

Nasledujúci postup opakujeme pre každý tovar $k=1,\ldots,w$. Na začiatku určíme množiny S a T nasledovným spôsobom. Do množiny S priradíme všetky uzly, ktoré dali požiadavku na odvoz tovaru a do množiny T dáme uzly, ktoré majú požiadavku na dovezenie daného typu dopravného elementu. Ďalej uvažujme všetky úseky vychádzajúce z množiny S a vchádzajúce do množiny T. Ostatné úseky zanedbáme. Všetky zahrnuté úseky ešte orientujeme. Orientácie bude smerom z množiny S do množiny T. Týmto spôsobom získame bipartitný digraf G. Nech x_{ij} označuje množstvo prepravované z s_i do t_j, kde $s_i \in S, t_j \in T$. Ďalší postup bude takýto:

1) Nájdeme prípustné riešenie, ktoré bude zodpovedať nejakej kostre digrafu G: na úsekoch kostry je $x_{ij} \geq 0$, ale mimo kostry je vždy $x_{ij} = 0$. Takéto riešenie x sa nazýva bázické riešenie a možno ho nájsť postupným nasycovaním uzlov: zvolíme dvojicu (i, j) a položíme $x_{ij} = \min\{a_i, b_j\}$ a do kostry berieme úsek $s_i t_j$. Tým sa niektorý z uzlov s_i, t_j nasýti, a preto ho z digrafu vynecháme. Druhý z uzlov vždy ponecháme a jeho požiadavku zmenšíme o x_{ij}. Potom postup opakujeme na redukovanom digrafe atď., až pokiaľ má digraf ešte hrany. Takto sme získali kostru a príslušné bázické riešenie. (Prvý vynechaný vrchol bude koncový a teda indukciou vidíme, že získame kostru.)

2) Ak máme nejakú kostru T a príslušné bázické riešenie x, tak nájdeme u_i a v_j spĺňajúce podmienky ortogonality (7). Keďže na hranách kostry môžu byť všetky $x_{ij} > 0$ musíme určiť u_i a v_j tak, aby $u_i + v_j - c_{ij} = 0$. To sa docieli tým, že zvolíme napr. $v_q = 0$ a ostatné

hodnoty u_i a v_j už vypočítame postupne cez hrany kostry T. (Výpočet je jednoznačný, lebo v T existuje jediná $t_q - s_i$, resp. $t_q - t_j$ polcesta.)

3) Ak pre všetky i a j máme $u_i + v_j \le c_{ij}$, t.j. ak (u, v) je prípustné riešenie duálnej úlohy, tak (keďže podmienky (7) sú splnené) x je optimálnym riešením primárnej a (u, v) duálnej úlohy.

4) Ak existuje $(\hat{i}, \hat{j})$ tak, že $u_i + v_j > c_{ij}$, zmeníme riešenie x a (v súlade so simplexovou metódou) utvoríme novú kostru T' obsahujúcu hranu $s_{\hat{i}}t_{\hat{j}}$. Nech C je (jediný) polocyklus v $T + s_{\hat{i}}t_{\hat{j}}$ uvažovaný tak, že šíp $s_{\hat{i}}t_{\hat{j}}$ je v ňom súhlasný. Vidíme, že ak hodnoty x_{ij} na súhlasných šípoch polocyklu zväčšíme o $\delta \ge 0$ a na nesúhlasných šípoch zmenšíme o δ, tak všetky požiadavky a_i, b_j budú splnené. Naviac, ak zvolíme δ maximálne možné a také, aby $x_{ij} - \delta \ge 0$ na všetkých nesúhlasných šípoch, získame nové prípustné riešenie $x' : x'_{ij} = x_{ij} + \delta$ na súhlasných šípoch, $x'_{ij} = x_{ij} - \delta$ na nesúhlasných šípoch polocyklu C a $x'_{ij} = x_{ij}$ inak. Týmto aspoň na jednom nesúhlasnom šípe $(\bar{i}, \bar{j}) \in C$ máme $x'_{\bar{i}\bar{j}} = 0$. Položiac $T' = T + s_{\hat{i}}t_{\hat{j}} - s_{\bar{i}}t_{\bar{j}}$, získame novú kostru T', takže x' je nové bázické riešenie a ideme na krok 2).

Dôvod na zavedenie úseku s_it_j do kostry je ten, že cena riešenia poklesne o $\delta(u_{\hat{i}} + v_{\hat{j}} - c_{\hat{i}\hat{j}})$, čo pri $\delta > 0$ je skutočný pokles účelovej funkcie v dopravnej úlohe.

Každej kostre zodpovedá jediné bázické riešenie. Skutočne: na každej koncovej hrane kostry je bázické riešenie jednoznačne určené (je rovné požiadavke koncového uzlu) a tvrdenie vyplýva indukciou. Preto ak vždy bude $\delta > 0$, tak žiadne bázické riešenie a teda ani žiadna kostra sa nemôže zopakovať. Keďže kostier je len konečne mnoho, tak z toho vyplýva konečnosť algoritmu. Žiaľ, ak sa vyskytne $\delta = 0$, tak môže vzniknúť zacyklenie. Taký príklad možno nájsť v [Gass (1964)], hoci v praktických úlohách sa zacyklenie nevyskytlo. Teoreticky sa možno poistiť proti zacykleniu nepatrným pozmenením (perturbáciou) požiadaviek a_i a b_j (ak $\varepsilon > 0$ je dostatočne malé, tak v úlohe s požiadavkami $a_1 + \varepsilon, \ldots, a_p + \varepsilon, b_1, b_2, \ldots, b_{q-1}, b_{q+1} + p\varepsilon$ budú vždy úsekom kostry zodpovedať kladné hodnoty riešenia a preto sa zacyklenie nevyskytne).

Hoci sa všeobecná dopravná úloha zdá byť omnoho zložitejšia, predsa ju možno polynomiálne pretransformovať na Hitchcockovu dopravnú úlohu.

Teraz je daná dopravná sieť s jedným strediskom a jednou náležitosťou s kapacitou K dopravných elementov. S dopravnou sieťou je zadaná množina prepráv $\{< u_i, v_i, q_i^k > | \, i = 1,2,\ldots, p, \, k = 1, \ldots, w\}$, kde $q_i^k \leq K$, $i = 1,2,\ldots, p$, $k = 1, \ldots, w$. (V tejto množine už premenné u_i a v_i neznamenajú duálne premenné z dopravnej úlohy, ale reprezentujú uzly v dopravnej sieti.)

Krok 2: Úloha okružných jázd

Našou úlohou je nájsť trasy náležitostí o minimálnom celkovom ohodnotení tak, aby boli realizované všetky prepravy a aby každá preprava $\{u_i, v_i, q_i^k\}$ bola uskutočnená. Ukážeme si dva algoritmy výpočtu. Ten prvý algoritmus je náročnejší na dobu výpočtu, avšak poskytuje presnejšie riešenie. Druhý algoritmus je odvodený od prvého, pričom je upravený tak, aby sa zmenšila výpočtová náročnosť daného algoritmu, avšak bude to na úkor presnosti riešenia. Algoritmy popíšeme v niekoľkých krokoch, avšak použijeme skôr slovné vyjadrenia ako matematické zápisy, nakoľko v prílohe C na priloženej diskete sa nachádza konkrétne popísaný algoritmus ako súčasť programu riešiaceho úlohu formulovanú v kapitole 1. Časť tohto algoritmu sa nachádza aj v prílohe. Označme stredisko 0. V nasledujúcich bodoch popíšeme algoritmus na hľadanie okružných jázd:

1) Za základ okružnej jazdy zvolíme obslúženie prepravy s najväčšími nákladmi, teda prepravu, ktorú je najdrahšie vykonať. Bude to jedna z prepráv, pre ktorú je $(c_{0i} + c_{ij} + c_{j0})$ maximálne. Týmto vznikne úvodná trasa v tvare $0 - u_i - v_i - 0$.

2) Upravíme trasu tak, že ak prvým, druhým, predposledným a posledným uzlom nie je stredisko, tak potom vsunieme do trasy toto stredisko, aby bola táto podmienka splnená. Všetky nezaradené prepravy do trasy usporiadame do vektora prepráv. Ak už neexistuje žiadna preprava, ktorú by bolo možne pridať do zoznamu, tak práca algoritmu končí. V opačnom prípade prechádzame na bod 3).

3) Pre každú prepravu zo zoznamu prepráv nájdeme pomocou bodu 4) najlepšie umiestnenie v doterajšej trase. Tú prepravu, ktorá najmenej predraží doterajšiu trasu pridáme do trasy na miesto určené bodom 4) a prejdeme na bod 2).

4) Chceme nájsť najvhodnejšie pridanie prepravy $\{u_i, v_i, q_i^k\}$ pre nejaké k do doterajšej trasy tak, aby umiestnenie tejto trasy zabezpečovalo minimálne zvýšenie nákladov na okružnú jazdu, pričom nesmie byť prekročená kapacita a musí byť dodržané poradie

navštívenia uzlu u_i pred uzlov v_i. Tento krok sa uskutoční pomocou porovnania všetkých možností a vybratím toho najlepšieho.

Týmto spôsobom sme dostali trasu, ktorú je ale ešte potrebné „kozmeticky“ upraviť, lebo v takejto trase sa môžu vyskytovať trasy z uzla *i* do uzla *i*. Takéto prípady odstránime jednoduchým zlúčením uzlov a ich operácii do jediného uzlu.

V bode 2) algoritmu na začiatku upravujeme trasu na taký tvar, aby prvé a posledné dva uzly bolo stredisko. Takto zabezpečíme priame vytváranie viacerých okružných jázd naraz. Tento postup je výpočtovo náročnejší ako postup, ktorý následne opíšeme, ale poskytuje riešenie „bližšie“ k optimálnemu riešeniu.

Modifikovaný postup je popísaný nasledovnými krokmi:

1) Za základ okružnej jazdy zvolíme obslúženie prepravy s najväčšími nákladmi, teda prepravu, ktorú je najdrahšie vykonať. Bude to jedna z prepráv, pre ktorú je $(c_{0i} + c_{ij} + c_{j0})$ maximálne. Týmto vznikne úvodná trasa v tvare $0 - i - j - 0$. Ak už žiadna preprava neexistuje, tak algoritmus končí.

2) Upravíme trasu tak, že ak sa v trase opakuje bezprostredne po sebe ten istý uzol, tak ich zlúčime do jedného uzla. Všetky nezaradené prepravy do trasy usporiadame do vektora prepráv. Ak už neexistuje žiadna preprava, ktorú by bolo možne pridať do vektora, tak prejdeme na bod 1). V opačnom prípade prechádzame na bod 3).

3) Pre každú prepravu zo zoznamu prepráv nájdeme pomocou bodu 4) najlepšie umiestnenie v doterajšej trase. Tú prepravy, ktorá najmenej predraží doterajšiu trasu pridáme do trasy na miesto určené bodom 4) a prejdeme na bod 2). Ak takéto prepravy neexistujú, dostali sme okružnú jazdu, ktorú odstránime z dopravnej siete (t.j. odstránime všetky prepravy z dopravnej siete), a začneme vytvárať novú okružnú jazdu tým, že prejdeme priamo na bod 1).

4) Chceme nájsť najvhodnejšie pridanie prepravy $\{u_i, v_i, q_i^k\}$ pre nejaké k do doterajšej trasy tak, aby umiestnenie tejto trasy zabezpečovalo minimálne zvýšenie nákladov na okružnú jazdu, pričom nesmie byť prekročená kapacita a musí byť dodržané poradie navštívenia uzlu u_i pred uzlov v_i. Ďalej musí platiť pre prepravy zo strediska, že návšteva strediska sa koná len na začiatku okružnej jazdy, teda náležitosť sa nesmie vrátiť do strediska pred ukončením okružnej jazdy. Tento krok sa uskutoční pomocou porovnania všetkých možností a vybratím toho najlepšieho.

Algoritmus vytvorí niekoľko okružných jázd pre náležitosť s kapacitou *K*.

Toto dosiahnuté riešenie je možné ešte zlepšovať pomocou niektorých iných heuristík, resp. metaheuristík. Princípy metaheuristýk sú popísané v kapitole 4.5. Zapracovanie týchto metaheuristík do nášho problému by si vyžiadalo ďalšiu podobnú pracú ako je táto a aj z dôvodu, že výsledky algoritmu majú slúžiť len ako informácie pre človeka, ktorý rozhoduje v konečnom dôsledku o tvare okružných jázd (t.j. je tu silný faktor ľudského rozhodovania), nebudeme s týmito spresňujúcimi metódami uvažovať a vystačíme si s riešením, ktoré nám ponuka predchádzajúci algoritmu.

6.3 Príklad riešený algoritmom

Teraz ukážeme konkrétne príklad a jeho riešenie druhým algoritmom. Volíme druhý algoritmus preto, lebo je menej výpočtovo náročný a riešenie budeme uskutočňovať už na staršom počítači. Pre porovnanie uvedieme aj najdôležitejšie parametre osobného počítača, ktorý budeme používať. Teda je to Intel ® Pentium Celeron ® 500MHz so 128 MB operačnej pamäte s použitým operačným systémom Microsoft Windows ® 98se SK. Ako databáza bode použitý produkt Microsoft SQL Server ® 2000.

Algoritmus je súčasťou programu, ktorý sa nachádza na priloženej diskete v prílohe C. Je naprogramovaný v jazyku PHP ako aplikácia, ktorá sa dá spúšťať cez ľubovoľný internetový prehliadač podporujúci protokol HTTP. Na inštaláciu je potrebný HTTP server. Odporúčam HTTP server od spoločnosti *The Apache Software Foundation,* ktorá má tento server na stránke www.apache.org zadarmo k stiahnutiu. Ďalej je potrebný program na spúšťanie PHP skriptov. Tento je možne získať z internetovej adresy www.php.net. Program spolupracuje s databázou, ktorá bola spravovaná produktom Microsoft SQL Server ® 2000. Ďalej sa budeme zaoberať už len riešením zadanej úlohy.

Vo formulácii úlohy sme si určili dva ciele na optimalizáciu. Zjednodušene je to optimalizácia so zahrnutím externého dodávateľa alebo bez neho. My budeme uvažovať úlohu s externým dodávateľom, ktorý ponúka neobmedzené množstvo tovaru (pre potreby programu si nekonečno nahradíme dostatočne vysokými ponukami – konkrétne 10000 jednotiek). Úlohou bez externého dodávateľa sa nebudeme zaoberať, lebo je to len zjednodušenie úlohy, ktorú si teraz predvedieme.

Na nasledujúcich tabuľkách ukážeme vstupné dáta.

Prevádzky v systéme

Stredisko ID	Názov strediska
0	Depo
1	Bratislava
2	Trnava
3	Žilina
4	Martin
5	Prešov

Stĺpec „Stredisko ID“ symbolizuje jednoznačný identifikátor pre stredisko a v druhom stĺpci sa nachádza názov strediska. V našom príklade stotožníme názov strediska s názvami miest a len miesto s náležitosťami označíme ako „Depo“.

Tovary v systéme

Tovar ID	Názov tovaru
1	mlieko
2	cukor
3	mrkva
4	záclona
5	topánky
6	počítač
7	stolička
8	stôl
9	klávesnica
10	sedem
11	mys
12	kruh
13	hračka
14	vlasy
15	varecha
16	nôž
17	príbor
18	karty
19	cigarety
20	gauč

V prvom stĺpci sa nachádza jednoznačný identifikátor pre tovar a za ním v druhom stĺpci je popis resp. názov tovaru. Uvažujme s deviatimi tovarmi.

Náklady c_{ij} v systéme

Stredisko 1	Stredisko 2	Náklady
Depo	Bratislava	1
Depo	Trnava	50
Depo	Žilina	200
Depo	Martin	220
Depo	Prešov	400
Bratislava	Trnava	50

Bratislava	Žilina	200
Bratislava	Martin	220
Bratislava	Prešov	400
Trnava	Žilina	150
Trnava	Martin	170
Trnava	Prešov	350
Žilina	Martin	20
Žilina	Prešov	200
Martin	Prešov	180

Tabuľka popisuje jednotlivé prepravné náklady c_{ij} v systéme. V jednom riadku sú uvedené názvy prevádzok (stredísk) a za nimi sú uvedené náklady na prepravu tovaru medzi týmito strediskami.

Požiadavky v systéme

Stredisko	Tovar	Množstvo
Depo	mlieko	-1000
Depo	cukor	-1000
Depo	mrkva	-1000
Depo	záclona	-1000
Depo	počítač	-1000
Depo	stôl	-1000
Depo	klávesnica	-1000
Depo	kruh	-1000
Depo	varecha	-1000
Depo	nôž	-1000
Depo	príbor	-1000
Bratislava	mrkva	-16
Bratislava	záclona	-16
Bratislava	topánky	-16
Bratislava	počítač	-16
Bratislava	stolička	-15
Bratislava	klávesnica	-14
Bratislava	sedem	-13
Bratislava	mys	-13
Bratislava	hračka	-12
Bratislava	vlasy	-12
Bratislava	nôž	-11
Bratislava	príbor	-11
Bratislava	karty	-10
Bratislava	cigarety	-10
Trnava	cukor	-10
Trnava	mrkva	-9
Trnava	počítač	-8
Trnava	klávesnica	-7
Trnava	sedem	-7
Trnava	mys	-7
Trnava	kruh	-7
Trnava	hračka	-7
Trnava	vlasy	-6
Trnava	príbor	-5
Trnava	karty	-4
Trnava	cigarety	-4
Žilina	mlieko	-3
Žilina	cukor	-3
Žilina	mrkva	-3
Žilina	záclona	-3
Žilina	topánky	-2
Žilina	počítač	-2
Žilina	stolička	-1
Žilina	stôl	-1
Žilina	klávesnica	-1
Žilina	sedem	0
Žilina	mys	0
Žilina	kruh	1
Žilina	vlasy	1
Žilina	nôž	2
Žilina	príbor	2
Žilina	cigarety	3
Martin	mlieko	3
Martin	záclona	5
Martin	topánky	5
Martin	počítač	5
Martin	stôl	6
Martin	klávesnica	6
Martin	kruh	7
Martin	hračka	7
Martin	vlasy	8
Martin	príbor	8
Prešov	mlieko	10
Prešov	cukor	10
Prešov	mrkva	11
Prešov	klávesnica	12
Prešov	mys	14
Prešov	kruh	14
Prešov	hračka	15
Prešov	varecha	15
Prešov	príbor	16
Prešov	karty	16
Prešov	cigarety	16
Depo	topánky	-1000
Depo	stolička	-1000
Depo	sedem	-1000
Depo	mys	-1000
Depo	hračka	-1000
Depo	vlasy	-1000
Depo	karty	-1000
Depo	cigarety	-1000
Bratislava	mlieko	-26
Bratislava	cukor	-18
Bratislava	stôl	-14
Bratislava	kruh	-13
Bratislava	varecha	-12
Trnava	mlieko	-10
Trnava	záclona	-9

Trnava	topánky	-8
Trnava	stolička	-8
Trnava	stôl	-8
Trnava	varecha	-6
Trnava	nôž	-5
Žilina	hračka	1
Žilina	varecha	2
Žilina	karty	3
Martin	cukor	3

Martin	mrkva	3
Martin	stolička	5
Martin	sedem	7
Martin	mys	7
Martin	varecha	8
Martin	nôž	8
Martin	karty	8
Martin	cigarety	10
Prešov	záclona	11

Prešov	topánky	11
Prešov	počítač	12
Prešov	stolička	12
Prešov	stôl	12
Prešov	sedem	13
Prešov	vlasy	15
Prešov	nôž	16

Prvý stĺpec určuje stredisko, ktoré podalo požiadavku. Druhý stĺpec názov tovaru, ktorý požaduje stredisko doviesť alebo odviesť. Tretí stĺpec určuje počet jednotiek tovaru, ktoré sa bude je potrebne odviesť alebo doviesť. Symbol „-“ pred číslom znamená, že je požiadavka na odvezenie tovaru, ináč, keď nie je uvedený symbol, sa jedná o dovezenie tovaru.

Kapacita náležitosti bola 150 jednotiek. Pripomínam, že táto kapacita musí byť väčšia než akákoľvek požiadavka mimo depa.

Výsledky optimalizácie

Výsledky optimalizácie sú uvedené v prílohe A.

Znamienka „-“ pri množstvách znamenajú, že sa jedná o operáciu vyloženie. Tam kde nie je uvedené žiadne znamienka, tak tam ide o operáciu naloženie.

Riešenie úlohy pomocou nášho algoritmu trvalo 90 sekúnd. Pomocou neho sme v tomto prípade dostali jednu okružnú jazdu popísanú tabuľkou v prílohe A. Pri použití výkonnejšieho počítača pri tých istých vstupných údajoch sa trvanie optimalizácie zredukovalo o vyše dve tretiny pôvodného času.

Z výsledkov možno vidieť, že nedostávame optimálne riešenie. To je však „daň“ za možnosť vôbec riešiť túto úlohu. Riešenie nám poskytlo suboptimálne riešenie, ktoré však ma v praxi slúžiť ako informácie pre zadávateľa okružných jázd, ktorý ešte môže zohľadniť aj iné faktory a podľa potreby upraviť tieto jazdy.

Záver

V tejto práci sme sa pokúsili o riešenie praktickej úlohy problému okružných jázd, ktorú sme si v úvode formulovali.

Na začiatku sme sa oboznámili so základnými pojmami používajúcimi sa v odbornej literatúre. Vysvetlili sme si niektoré dôležité pojmy ako napr. dopravný element, orientovaná dopravná sieť, alebo okružná jazda.

Potom sme sformulovali niektoré z už riešených klasických úloh, ktoré nám poskytli základ pre riešenie našej úlohy.

Následne sme klasifikovali metódy trasovania a sformulovali niekoľko modelov.

V 5. kapitole sme sformulovali model úlohy, ktorá bola východiskom pre túto prácu, a uviedli možnosti použitia tohto modelu, pričom sme zistili, že už i pri jednoduchších modeloch nie je vôbec jednoduché nájsť optimálne riešenie.

Potom sme sa pokúsili o riešenie úlohy a na ilustračnom príklade sme aplikovali toto riešenie úlohy s použitím vlastného programu, ktorého súčasťou je naprogramovaný algoritmus.

Literatúra

[1] Plesník, J.: Grafové algoritmy, Bratislava, VEDA 1983

[2] Cousineau-Quimet, K.: A Tabu Search Heuristic for the Inventory Routing Problem, Montreal, Canada, 2002

[3] Skýva, L., Janáček, L., Cenek, P.: Energeticky optimální řízení dopravních systémů, Praha, Nadas 1987

[4] Janáček, J.: Optimalizace na dopravních sítích, Žilina, EDIS ŽU, v tlači (2003)

[5] Černý, J., Kluvánek, P.: Základy matematickej teórie dopravy, Bratislava, Veda, 1991

[6] Chan Gin Hor, Song Yuyue: A New Approach for the One-Warehouse Multi-Retailer Distribution Systems, National University of Singapore, Singapore, www.ise.nus.edu.sg 2002

[7] Valent, T.: Problém obchodného cestujúceho, FMFI UK, Bratislava, diplomová práca, 2002

[8] Janáček, J.: Opreračná analýza I, Žilina, EDIS ŽU, 1995

[9] Bernabé Dorronsoro Díaz: http://neo.lcc.uma.es/radi-aeb/WebVRP/ , 2002

[10] Baramel, J., Simchi-Levi, D.: A Location Based Heuristic for General Routing Problems, paper, 1992

[11] Chandra, P.: A Dynamic Distribution Model with Warehouse and Customer Replenishment Requirements, paper www.decisioncraft.com, Mc Gill University, Quebee, Canada,

[12] Muslea, I.: The Very Offline k-Vehicle Routing Problem in Trees, Information Sciences Institute / University of Southern California, USA,

[13] Applegate, D., Bixby, R., Chvatal, V. and Cook, W.: Solving Traveling Salesman Problems, http://www.math.princeton.edu/tsp/ ,2002

[14] Garn, W. W. : Clarke and Wright Savings Algorithm, http://osiris.tuwien.ac.at/~wgarn/, 2002

[15] Applegate, D., Cook, W., Rohe, A.: Chained Lin-Kernighan for large traveling salesman problems, http://citeseer.nj.nec.com, 1999

[16] Korbut, A. A., Finkelštejn, J. J.: Diskrétne programovanie, Bratislava, Alfa, 1972

[17] Ďuriník, P.: Metódy riešenia úlohy okružných jázd, Žilina, Vysoká škola dopravy a spojov, Diplomová práca, 1984

Príloha A - Výsledky optimalizácie

Výpočet bežal 90 sekúnd!
Náležitosť má kapacitu:100

Poradie	Jazda	uzol	Tovar c.	Množstvo		Zmena v uzle	V aute
0.	0	**0 - Depo**				68	68
0.	0	**0 - Depo**	11 - kruh	Naložím 1	Odveziem do Žilina	68	68
0.	0	**0 - Depo**	12 - hračka	Naložím 1	Odveziem do Žilina	68	68
0.	0	**0 - Depo**	13 - vlasy	Naložím 1	Odveziem do Žilina	68	68
0.	0	**0 - Depo**	14 - varecha	Naložím 2	Odveziem do Žilina	68	68
0.	0	**0 - Depo**	15 - nôž	Naložím 2	Odveziem do Žilina	68	68
0.	0	**0 - Depo**	16 - príbor	Naložím 2	Odveziem do Žilina	68	68
0.	0	**0 - Depo**	17 - karty	Naložím 3	Odveziem do Žilina	68	68
0.	0	**0 - Depo**	18 - cigarety	Naložím 3	Odveziem do Žilina	68	68
0.	0	**0 - Depo**	10 - mys	Naložím 1	Odveziem do Martin	68	68
0.	0	**0 - Depo**	11 - kruh	Naložím 1	Odveziem do Martin	68	68
0.	0	**0 - Depo**	12 - hračka	Naložím 3	Odveziem do Martin	68	68
0.	0	**0 - Depo**	13 - vlasy	Naložím 5	Odveziem do Martin	68	68
0.	0	**0 - Depo**	14 - varecha	Naložím 5	Odveziem do Martin	68	68
0.	0	**0 - Depo**	15 - nôž	Naložím 8	Odveziem do Martin	68	68
0.	0	**0 - Depo**	16 - príbor	Naložím 8	Odveziem do Martin	68	68
0.	0	**0 - Depo**	17 - karty	Naložím 8	Odveziem do Martin	68	68
0.	0	**0 - Depo**	18 - cigarety	Naložím 10	Odveziem do Martin	68	68
0.	0	**0 - Depo**	17 - karty	Naložím 2	Odveziem do Prešov	68	68
0.	0	**0 - Depo**	18 - cigarety	Naložím 2	Odveziem do Prešov	68	68
1.	0	**1 - Bratislava**	18 - cigarety	Naložím 10	Odveziem do Prešov	10	78
2.	0	**5 - Prešov**	18 - cigarety	Vyložím -10	Pochádzajúce z Bratislava	-10	68
3.	0	**1 - Bratislava**	17 - karty	Naložím 10	Odveziem do Prešov	32	100
3.	0	**1 - Bratislava**	16 - príbor	Naložím 11	Odveziem do Prešov	32	100
3.	0	**1 - Bratislava**	15 - nôž	Naložím 11	Odveziem do Prešov	32	100
4.	0	**5 - Prešov**	17 - karty	Vyložím -10	Pochádzajúce z Bratislava	-32	68
4.	0	**5 - Prešov**	16 - príbor	Vyložím -11	Pochádzajúce z Bratislava	-32	68
4.	0	**5 - Prešov**	15 - nôž	Vyložím -11	Pochádzajúce z Bratislava	-32	68
5.	0	**1 - Bratislava**	14 - varecha	Naložím 9	Odveziem do Prešov	26	94
5.	0	**1 - Bratislava**	13 - vlasy	Naložím 9	Odveziem do Prešov	26	94
5.	0	**1 - Bratislava**	12 - hračka	Naložím 8	Odveziem do Prešov	26	94
6.	0	**5 - Prešov**	14 - varecha	Vyložím -9	Pochádzajúce z Bratislava	-26	68
6.	0	**5 - Prešov**	13 - vlasy	Vyložím -9	Pochádzajúce z Bratislava	-26	68
6.	0	**5 - Prešov**	12 - hračka	Vyložím -8	Pochádzajúce z Bratislava	-26	68
7.	0	**1 - Bratislava**	11 - kruh	Naložím 7	Odveziem do Prešov	30	98
7.	0	**1 - Bratislava**	10 - mys	Naložím 7	Odveziem do Prešov	30	98
7.	0	**1 - Bratislava**	9 - sedem	Naložím 6	Odveziem do Prešov	30	98
7.	0	**1 - Bratislava**	8 - klávesnica	Naložím 4	Odveziem do Prešov	30	98
7.	0	**1 - Bratislava**	7 - stôl	Naložím 3	Odveziem do Prešov	30	98
7.	0	**1 - Bratislava**	6 - stolička	Naložím 3	Odveziem do Prešov	30	98
8.	0	**5 - Prešov**	11 - kruh	Vyložím -7	Pochádzajúce z Bratislava	-7	91
9.	0	**2 - Trnava**	18 - cigarety	Naložím 4	Odveziem do Prešov	4	95

10.	0	**5 - Prešov**	10 - mys	Vyložím -7	Pochádzajúce z Bratislava	-27	68
10.	0	**5 - Prešov**	9 - sedem	Vyložím -6	Pochádzajúce z Bratislava	-27	68
10.	0	**5 - Prešov**	8 - klávesnica	Vyložím -4	Pochádzajúce z Bratislava	-27	68
10.	0	**5 - Prešov**	7 - stôl	Vyložím -3	Pochádzajúce z Bratislava	-27	68
10.	0	**5 - Prešov**	6 - stolička	Vyložím -3	Pochádzajúce z Bratislava	-27	68
10.	0	**5 - Prešov**	18 - cigarety	Vyložím -4	Pochádzajúce z Trnava	-27	68
11.	0	**2 - Trnava**	17 - karty	Naložím 4	Odveziem do Prešov	30	98
11.	0	**2 - Trnava**	14 - varecha	Naložím 6	Odveziem do Prešov	30	98
11.	0	**2 - Trnava**	13 - vlasy	Naložím 6	Odveziem do Prešov	30	98
11.	0	**2 - Trnava**	12 - hračka	Naložím 7	Odveziem do Prešov	30	98
11.	0	**2 - Trnava**	11 - kruh	Naložím 7	Odveziem do Prešov	30	98
12.	0	**5 - Prešov**	18 - cigarety	Vyložím -2	Pochádzajúce z Depo	-34	64
12.	0	**5 - Prešov**	17 - karty	Vyložím -4	Pochádzajúce z Trnava	-34	64
12.	0	**5 - Prešov**	17 - karty	Vyložím -2	Pochádzajúce z Depo	-34	64
12.	0	**5 - Prešov**	14 - varecha	Vyložím -6	Pochádzajúce z Trnava	-34	64
12.	0	**5 - Prešov**	13 - vlasy	Vyložím -6	Pochádzajúce z Trnava	-34	64
12.	0	**5 - Prešov**	12 - hračka	Vyložím -7	Pochádzajúce z Trnava	-34	64
12.	0	**5 - Prešov**	11 - kruh	Vyložím -7	Pochádzajúce z Trnava	-34	64
13.	0	**4 - Martin**	18 - cigarety	Vyložím -10	Pochádzajúce z Depo	-34	30
13.	0	**4 - Martin**	17 - karty	Vyložím -8	Pochádzajúce z Depo	-34	30
13.	0	**4 - Martin**	16 - príbor	Vyložím -8	Pochádzajúce z Depo	-34	30
13.	0	**4 - Martin**	15 - nôž	Vyložím -8	Pochádzajúce z Depo	-34	30
14.	0	**1 - Bratislava**	14 - varecha	Naložím 3	Odveziem do Martin	65	95
14.	0	**1 - Bratislava**	13 - vlasy	Naložím 3	Odveziem do Martin	65	95
14.	0	**1 - Bratislava**	12 - hračka	Naložím 4	Odveziem do Martin	65	95
14.	0	**1 - Bratislava**	11 - kruh	Naložím 6	Odveziem do Martin	65	95
14.	0	**1 - Bratislava**	10 - mys	Naložím 6	Odveziem do Martin	65	95
14.	0	**1 - Bratislava**	9 - sedem	Naložím 7	Odveziem do Martin	65	95
14.	0	**1 - Bratislava**	8 - klávesnica	Naložím 6	Odveziem do Martin	65	95
14.	0	**1 - Bratislava**	5 - počítač	Naložím 2	Odveziem do Prešov	65	95
14.	0	**1 - Bratislava**	7 - stôl	Naložím 6	Odveziem do Martin	65	95
14.	0	**1 - Bratislava**	6 - stolička	Naložím 5	Odveziem do Martin	65	95
14.	0	**1 - Bratislava**	5 - počítač	Naložím 5	Odveziem do Martin	65	95
14.	0	**1 - Bratislava**	4 - topánky	Naložím 5	Odveziem do Martin	65	95
14.	0	**1 - Bratislava**	3 - záclona	Naložím 4	Odveziem do Martin	65	95
14.	0	**1 - Bratislava**	2 - mrkva	Naložím 2	Odveziem do Martin	65	95
14.	0	**1 - Bratislava**	4 - topánky	Naložím 1	Odveziem do Prešov	65	95
15.	0	**2 - Trnava**	16 - príbor	Naložím 5	Odveziem do Prešov	5	100
16.	0	**5 - Prešov**	16 - príbor	Vyložím -5	Pochádzajúce z Trnava	-5	95
17.	0	**4 - Martin**	14 - varecha	Vyložím -3	Pochádzajúce z Bratislava	-48	47
17.	0	**4 - Martin**	14 - varecha	Vyložím -5	Pochádzajúce z Depo	-48	47
17.	0	**4 - Martin**	13 - vlasy	Vyložím -3	Pochádzajúce z Bratislava	-48	47
17.	0	**4 - Martin**	13 - vlasy	Vyložím -5	Pochádzajúce z Depo	-48	47
17.	0	**4 - Martin**	12 - hračka	Vyložím -4	Pochádzajúce z Bratislava	-48	47
17.	0	**4 - Martin**	12 - hračka	Vyložím -3	Pochádzajúce z Depo	-48	47
17.	0	**4 - Martin**	11 - kruh	Vyložím -6	Pochádzajúce z Bratislava	-48	47
17.	0	**4 - Martin**	10 - mys	Vyložím -6	Pochádzajúce z Bratislava	-48	47
17.	0	**4 - Martin**	9 - sedem	Vyložím -7	Pochádzajúce z Bratislava	-48	47
17.	0	**4 - Martin**	8 - klávesnica	Vyložím -6	Pochádzajúce z Bratislava	-48	47
18.	0	**2 - Trnava**	6 - stolička	Naložím 8	Odveziem do Prešov	52	99
18.	0	**2 - Trnava**	5 - počítač	Naložím 8	Odveziem do Prešov	52	99

18.	0	**2 - Trnava**	4 - topánky	Naložím 8	Odveziem do Prešov	52	99
18.	0	**2 - Trnava**	3 - záclona	Naložím 8	Odveziem do Prešov	52	99
18.	0	**2 - Trnava**	2 - mrkva	Naložím 8	Odveziem do Prešov	52	99
18.	0	**2 - Trnava**	1 - cukor	Naložím 7	Odveziem do Prešov	52	99
18.	0	**2 - Trnava**	3 - záclona	Naložím 1	Odveziem do Martin	52	99
18.	0	**2 - Trnava**	2 - mrkva	Naložím 1	Odveziem do Martin	52	99
18.	0	**2 - Trnava**	1 - cukor	Naložím 3	Odveziem do Martin	52	99
19.	0	**3 - Žilina**	18 - cigarety	Vyložím -3	Pochádzajúce z Depo	1	100
19.	0	**3 - Žilina**	17 - karty	Vyložím -3	Pochádzajúce z Depo	1	100
19.	0	**3 - Žilina**	16 - príbor	Vyložím -2	Pochádzajúce z Depo	1	100
19.	0	**3 - Žilina**	15 - nôž	Vyložím -2	Pochádzajúce z Depo	1	100
19.	0	**3 - Žilina**	14 - varecha	Vyložím -2	Pochádzajúce z Depo	1	100
19.	0	**3 - Žilina**	13 - vlasy	Vyložím -1	Pochádzajúce z Depo	1	100
19.	0	**3 - Žilina**	12 - hračka	Vyložím -1	Pochádzajúce z Depo	1	100
19.	0	**3 - Žilina**	11 - kruh	Vyložím -1	Pochádzajúce z Depo	1	100
19.	0	**3 - Žilina**	8 - klávesnica	Naložím 1	Odveziem do Prešov	1	100
19.	0	**3 - Žilina**	7 - stôl	Naložím 1	Odveziem do Prešov	1	100
19.	0	**3 - Žilina**	6 - stolička	Naložím 1	Odveziem do Prešov	1	100
19.	0	**3 - Žilina**	5 - počítač	Naložím 2	Odveziem do Prešov	1	100
19.	0	**3 - Žilina**	4 - topánky	Naložím 2	Odveziem do Prešov	1	100
19.	0	**3 - Žilina**	3 - záclona	Naložím 3	Odveziem do Prešov	1	100
19.	0	**3 - Žilina**	2 - mrkva	Naložím 3	Odveziem do Prešov	1	100
19.	0	**3 - Žilina**	1 - cukor	Naložím 3	Odveziem do Prešov	1	100
20.	0	**4 - Martin**	11 - kruh	Vyložím -1	Pochádzajúce z Depo	-34	66
20.	0	**4 - Martin**	10 - mys	Vyložím -1	Pochádzajúce z Depo	-34	66
20.	0	**4 - Martin**	7 - stôl	Vyložím -6	Pochádzajúce z Bratislava	-34	66
20.	0	**4 - Martin**	6 - stolička	Vyložím -5	Pochádzajúce z Bratislava	-34	66
20.	0	**4 - Martin**	5 - počítač	Vyložím -5	Pochádzajúce z Bratislava	-34	66
20.	0	**4 - Martin**	4 - topánky	Vyložím -5	Pochádzajúce z Bratislava	-34	66
20.	0	**4 - Martin**	3 - záclona	Vyložím -4	Pochádzajúce z Bratislava	-34	66
20.	0	**4 - Martin**	2 - mrkva	Vyložím -2	Pochádzajúce z Bratislava	-34	66
20.	0	**4 - Martin**	3 - záclona	Vyložím -1	Pochádzajúce z Trnava	-34	66
20.	0	**4 - Martin**	2 - mrkva	Vyložím -1	Pochádzajúce z Trnava	-34	66
20.	0	**4 - Martin**	1 - cukor	Vyložím -3	Pochádzajúce z Trnava	-34	66
21.	0	**2 - Trnava**	15 - nôž	Naložím 5	Odveziem do Prešov	34	100
21.	0	**2 - Trnava**	10 - mys	Naložím 7	Odveziem do Prešov	34	100
21.	0	**2 - Trnava**	9 - sedem	Naložím 7	Odveziem do Prešov	34	100
21.	0	**2 - Trnava**	8 - klávesnica	Naložím 7	Odveziem do Prešov	34	100
21.	0	**2 - Trnava**	7 - stôl	Naložím 8	Odveziem do Prešov	34	100
22.	0	**5 - Prešov**	15 - nôž	Vyložím -5	Pochádzajúce z Trnava	-100	0
22.	0	**5 - Prešov**	10 - mys	Vyložím -7	Pochádzajúce z Trnava	-100	0
22.	0	**5 - Prešov**	9 - sedem	Vyložím -7	Pochádzajúce z Trnava	-100	0
22.	0	**5 - Prešov**	8 - klávesnica	Vyložím -7	Pochádzajúce z Trnava	-100	0
22.	0	**5 - Prešov**	7 - stôl	Vyložím -8	Pochádzajúce z Trnava	-100	0
22.	0	**5 - Prešov**	5 - počítač	Vyložím -2	Pochádzajúce z Bratislava	-100	0
22.	0	**5 - Prešov**	6 - stolička	Vyložím -8	Pochádzajúce z Trnava	-100	0
22.	0	**5 - Prešov**	5 - počítač	Vyložím -8	Pochádzajúce z Trnava	-100	0
22.	0	**5 - Prešov**	4 - topánky	Vyložím -8	Pochádzajúce z Trnava	-100	0
22.	0	**5 - Prešov**	3 - záclona	Vyložím -8	Pochádzajúce z Trnava	-100	0
22.	0	**5 - Prešov**	2 - mrkva	Vyložím -8	Pochádzajúce z Trnava	-100	0
22.	0	**5 - Prešov**	1 - cukor	Vyložím -7	Pochádzajúce z Trnava	-100	0

22.	0	**5 - Prešov**	8 - klávesnica	Vyložím -1	Pochádzajúce z Žilina	-100	0
22.	0	**5 - Prešov**	7 - stôl	Vyložím -1	Pochádzajúce z Žilina	-100	0
22.	0	**5 - Prešov**	6 - stolička	Vyložím -1	Pochádzajúce z Žilina	-100	0
22.	0	**5 - Prešov**	5 - počítač	Vyložím -2	Pochádzajúce z Žilina	-100	0
22.	0	**5 - Prešov**	4 - topánky	Vyložím -2	Pochádzajúce z Žilina	-100	0
22.	0	**5 - Prešov**	3 - záclona	Vyložím -3	Pochádzajúce z Žilina	-100	0
22.	0	**5 - Prešov**	2 - mrkva	Vyložím -3	Pochádzajúce z Žilina	-100	0
22.	0	**5 - Prešov**	1 - cukor	Vyložím -3	Pochádzajúce z Žilina	-100	0
22.	0	**5 - Prešov**	4 - topánky	Vyložím -1	Pochádzajúce z Bratislava	-100	0
23.	0	**0 - Depo**				0	0

Príloha B – Najdôležitejšie kroky algoritmu

```
class Uzol
{
	// q[i] množstvo i-teho tovaru na dovezenie (ak je kladne) alebo odvezenie (ak je záporné)
	var $q;
	//id uzlu v databáze
	var $id;
}

class Graf
{
	//uzol v dopravnej sieti (nadobúda hodnoty 0, 1 - nieje v sieti, je v sieti)
	var $uzol;
	var $x;	//xkij množstvo prepravy z i do j pre tovar k
	var $c;	 // c[i][j] náklady na prepravu tovaru z i do j-teho uzlu
	var $pocetTovarov;
	var $K;	//kapacita vozidla
	var $okruzneJazdy; //pole obsahujúce okružné jazdy
	var $tovarID;	//id tovarov

	function dopravnaUloha(){
		for($k=1;$k<$this->pocetTovarov;$k++){
			//určíme množinu dodávateľov a odberateľov
			$mnozinaDod=NULL;
			$mnozinaOdb=NULL;
			$m=0;
			$n=0;
			for($i=0;$i<sizeof($this->uzol);$i++) {
				//len pre tie tovary, ktoré existujú pre uzol
				if($this->uzol[$i]->q[$k]!=NULL && $this->uzol[$i]->q[$k]!=""){
					if($this->uzol[$i]->q[$k]<0) $mnozinaDod[$m++]=$i;
					if($this->uzol[$i]->q[$k]>0) $mnozinaOdb[$n++]=$i;
				}
			}
			//krok 1 - nájdeme prípustné bázické riešenie
			$this->najdiPripustneRiesenie($mnozinaDod,$mnozinaOdb,$k);
			$splna=FALSE;
			while(!$splna) {
				// krok 2 - nájdeme všetky ui a vi
				$res= $this->najdiUiaVi($k);
				//krok 3 a 4 - overenie a upravenie v prípade neúspechu
				$splna=$this->overUiaVi($res,$k);
			}
		}
	}


	function urobOkruzneJazdy(){
		$koncim=FALSE;
		$okr=0; //počet okružných jázd - iterátor
		do{
			//premenná jazda bude obsahovať polia : uzol, zoznam tovarov, koľko (+) nakladá na auto (dodávateľ),resp. (-
) dováža a vykladá z auta (odberateľ)(pozn.: je to opačné ako pri DU), zoznam koľko pre každý tovar zvlášť, zoznam uzlov
kam ide resp. odkiaľ prišiel tovar pre každý tovar zvlast
			//vychádzame z jazdy stredisko – stredisko bez tovarov
			$jazda=array(array(0,array(0),0,array(0),array(0)), array(0,array(0),0,array(0),array(0)));
			//určenie prvej požiadavky
			$prvaPoziadavka=$this->najdiNajdrahsiuJazdu();
			//ak existuje prípustná požiadavka
			if($prvaPoziadavka){
				$jazda=$this->pridajPoziadavku (...,$jazda,$prvaPoziadavka,...,1,1);
				$jazda=$this->zlucUzly($jazda);
				do{
					// pre úplne každé xkij>0 zistím, že ktoré zväčšenie okružnej jazdy je najlacnejšie
					$exisPoziad=FALSE; //identifikátor, či existuje požiadavka, ktorá sa dá umiestniť (ak nie, tak teda už
sa nedá zväčšiť okružná jazda a končím)
					$pozHodn=NULL;
					$poziadavka=NULL;
					$r=0; //iterátor na požiadavky
					for($k=0; $k<$this->pocetTovarov;$k++)
```

```
                        for($i=0; $i<sizeof($this->uzol);$i++)
                            for($j=0; $j<sizeof($this->uzol);$j++)
                                if( $this->x[$k][$i][$j]>0){
                                    $poziadavka[$r]=$this->umiestniPoziadavku($jazda,$k,$i,$j);
                                    if($poziadavka[$r]){
                                        $pozHodn[$r]=$poziadavka[$r]["hodnota"];
                                        $poziadavka[$r]["tovar"]=$k;
                                        $poziadavka[$r]["i"]=$i;
                                        $poziadavka[$r]["j"]=$j;
                                        $r++;
                                        $exisPoziad=TRUE;
                                    }
                                }
                        if($exisPoziad){
                            $min=min($pozHodn);
                            //zistím v ktorých požiadavkách je predĺženie najmenšie
                            $ii=0;
                            for($i=0;$i<sizeof($pozHodn);$i++){
                                if($pozHodn[$i]==$min) $mnozinaI[$ii++]=$i;
                            }
                            //zo všetkých miním vyberieme to, ktoré najskôr vyloží tovar.
                            //na začiatku to nastavím na nultý prvok
                            $minI=$poziadavka[$mnozinaI[0]]["kde2"]-$poziadavka[$mnozinaI[0]]["kde1"];
                            $i=$mnozinaI[0];
                            //a potom prehľadávam cez všetko
                            for($j=0;$j<$ii;$j++){
                                if($poziadavka[$mnozinaI[$j]]["kde2"]-$poziadavka[$mnozinaI[$j]]["kde1"]<$minI){
                                    $minI=$poziadavka[$mnozinaI[$j]]["kde2"]-$poziadavka[$mnozinaI[$j]]["kde1"];
                                    $i=$mnozinaI[$j];
                                }
                            }
                            //už mam určenú požiadavku a pracujem s ňou ďalej
                            $jazda=$this->pridajPoziadavku(...,$jazda,$poziadavka,...,kde1, kde2);
                            $jazda=$this->zlucUzly($jazda);
                            $i=sizeof($pozHodn);
                        }
                    }while($exisPoziad);
                    $this->okruznaJazdy[$okr++]=$jazda;
                }else $koncim=TRUE;
            }while(!$koncim);
        }
}

function algoritmus(){
    $graf = new Graf();
    //krok 1
    $graf->dopravnaUloha();
    //krok 2
    $graf->urobOkruzneJazdy();

  $vysledok[0]=$graf->okruzneJazdy;
  $vysledok[1]=$graf->popisUzlov();
  $vysledok[2]=$graf->popisTovarov();
  $vysledok[3]=$graf->K;
  return $vysledok;
}

$jazdy=algoritmus();
```

Príloha C – Disketa s programom

Program s algoritmom, ktorý rieši úlohu formulovanú v kapitole 1, sa nachádza na priloženej diskete.