VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MATEMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF MATHEMATICS

OPTIMALIZACE TOKŮ V SÍTÍCH
OPTIMALIZATION OF NETWORK FLOWS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR — ADAM RYCHTÁŘ

VEDOUCÍ PRÁCE
SUPERVISOR — RNDr. PAVEL POPELA, Ph.D.

BRNO 2014

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav matematiky
Akademický rok: 2013/2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Adam Rychtář

který/která studuje v **bakalářském studijním programu**

obor: **Matematické inženýrství (3901R021)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Optimalizace toků v sítích

v anglickém jazyce:

Optimization of network flows

Stručná charakteristika problematiky úkolu:

Student využije své znalosti z lineární algebry (zejména maticové reprezentace grafů) a matematické statistiky doplněné o znalosti optimalizace v rozsahu matematické analýzy a na základě reálných podkladů sestaví matematický model. Jako programový nástroj bude použit vhodný modelovací matematický jazyk osvojený samostatně např. GAMS.

Cíle bakalářské práce:

Výsledný matematický model bude implementován a umožní provádění testovacích výpočtů pro problémy řešené v rámci spolupráce mezi ÚM a MUC v Molde.

Seznam odborné literatury:

Klapka, J. a kol. Metody operačního výzkumu, Brno 2000
Bazaraa M. - Jarvis R. Linear Programming and Network Flows, Wiley and Sons, 1991

Vedoucí bakalářské práce: RNDr. Pavel Popela, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2013/2014.

V Brně, dne 22.11.2013

L.S.

prof. RNDr. Josef Šlapal, CSc.	prof. RNDr. Miroslav Doupovec, CSc., dr. h. c.
Ředitel ústavu	Děkan fakulty

Abstrakt
Tato bakalářská práce se zabývá přiřazením toků odpadu hranou ke konkrétnímu producentovi a aplikací na reálný model mapující Českou republiku. Cílem je určit rozptyl nákladů každého producenta při změnách včasnosti zajištění zpracování odpadu a analyzovat dopady a popřípadě i najít rizikové uzly.

Tento problém převedeme do matematické formulace, k výpočtům použijeme programovací prostředí GAMS.

Summary
This bachelor thesis focuses on the assigment of waste flows to specific producer for each edge andaplication on real model which maps the Czech Republic and the model is applied to real data of the Czech Republic. The goal is to determine variance of the expenses for each producer subject to changes in waste processing requirements. Then, the solution impacts and bottlenecks are analysed.

The suitable optmization models is formulated and computations are realizaed in GAMS.

Klíčová slova
optimalizace, dopravní úloha, lineární programování, odpadové hospodářství

Keywords
optimalization, transportation problem, linear programming, waste-managment

RYCHTÁŘ, A. *Optimalizace toků v sítích.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014. 32 s. Vedoucí bakalářské práce RNDr. Pavel Popela, Ph.D.

Prohlašuji, že jsem bakalářskou práci *Optimalizace toků v sítích* vypracoval samostatně pod vedením RNDr. Pavla Popely, Ph.D. s použitím materiálů uvedených v seznamu literatury.

Adam Rychtář

Zde bych rád poděkoval všem, kteří mi pomohli s tvorbou této práce, zejména pak RNDr. Pavlovi Popelovi, Ph.D. za vedení práce a konzultace, dále pak Ing. Radovanu Šomplákovi za ochotu, přínosné rady a čas, který mi věnoval.

Adam Rychtář

Obsah

Úvod

Česká republika se zavázala omezit skládkování biodegradabilního materiálu obsaženého ve směsném komunálním odpadu (SKO). S tím jsou spojeny pro jeho producenty, převážně obce, nemalé obtíže a to i proto, že nakládání s komunálním odpadem není prakticky státní správou řešeno. Proto vznikají různé studie, které se zabývají problematikou výstavby nových zařízení na zpracování komunálního odpadu a jejich finanční udržitelností.

Dále je nutné vzít do úvahy, že dopady na konkrétní obce mohou zásadně ovlivnit úspěšnost snah o reálné řešení nákládání s odpadem. Obecní rozpočty nebývají vysoké a každý další výdaj je značně zatěžuje. Pokud chceme, aby nový systém mohl fungovat efektivně, nemůžeme tyto vlivy zanedbat, ba právě naopak je nutné je analyzovat.

Touto problematikou se dlouhodobě zabývají zaměstnanci na pracovišti Ústavu procesního a ekologického inženýrství (ÚPEI), VUT v Brně. V rámci svojí práce rozvíjí výpočtový nástroj NERUDA.

Nejdříve se proto s projektem NERUDA stručně seznámíme. Ten problém nakládání s komunálním odpadem pojal velmi komplexně. Svůj název odvodil podle snad nejznámějšího fejetonu Kam s ním. Jeho autor Jan Neruda se v něm zamýšlí, jak naložit se starým slamníkem, v analogii na to byl právě takto pojmenován projekt, který snad odpověď Janu Nerudovi přinést dokáže. Přiblížíme si také pojmy, které s odpadovým hospodářstvím souvisejí.

Jádro komplexního systému NERUDA je optimalizační výpočet. S jeho pomocí je možné hledat vhodná řešení zajišťující minimální dopad na producenty a zároveň zajišťuje dosažení stanovených cílů v oblasti odpadového hospodářství. V tomto směru rozšířila projekt NERUDA bakalářská práce Víta Procházy, viz [10]. V práci bude nutný teoretický základ týkající se optimalizace, ten bude uveden v kapitole 2.

Aby bylo možné řešit odpadové hospodářství v rámci celé České republiky, je nutné si tuto oblast rozdělit na na vhodné lokality menší rozlohy. V dosavadní práci se uvažovalo rozdělení na bázi obcí s rozšířenou působností (ORP). Pro umožnění přepravy mezi jednotlivými ORP (uzly) je nutné nadefinovat dopravní infrastrukturu (hrany). Těmto úlohám se věnuje matematická disciplína nazvaná teorie grafů, bližší popis najdete v kapitole 3.

Když budeme znát všechny potřebné pojmy, představíme si jednoduchou motivační úlohu a pro ni představíme možnosti řešení. Nakonec vytvořený model aplikujeme na projekt NERUDA a přímo na něm ukážeme výsledky výpočtů a jejich význam. Systém NERUDA již byl v minulosti využit k mnoha aplikacím ve spojení s průmyslem. Jeho zásadní omezení vyplývá z toho, že se toky v síti "stékají"a následně "roztékají", a tudíž není jednoznačně určen zpracovatel pro konkrétního producenta, všechna řešení jsou totiž z pohledu celkových nákladů stejně kvalitní. Tento fakt komplikuje analýzu a následná doporučení producentům odpadu.

Tato práce je součástí aktivit řešených v rámci centra kompetence Waste-to-Energy (projekt č. TE02000236) financovaného Technologickou agenturou České republiky.

1 Síťové problémy v procesním inženýrství

Podnětem pro vznik této práce je rozšíření svozové úlohy NERUDA, která je realizována ÚPEI. Zabývá se nejen distribucí odpadu z míst jeho vzniku obcí k prostředkům jeho zpracování, ale dále se snaží navrhnout optimální umístění zařízení na zpracování odpadu, jejich kapacitu, z níž jsou odvozeny poplatky za zpracování, a případně i legislativní opatření.

Proto v roce 2012 vznikl rozsáhlý model zkoumající možnosti alternativního zpracování oproti skládkování, jeho finanční náročnost, zahrnující předchozí vlivy i možný vývoj v zahraničí, viz [13]. V současném modelu ale nelze analyzovat dopady na jednotlivé obce, protože známe toky mezi jednotlivými uzly, nikoliv už mezi uzly producentskými a zpracovatelskými.

Znalost toků mezi producentem a zpracovatelem je ale důležitá, pokud potřebujeme analyzovat jednotlivé územní celky nebo dopady a finanční udržitelnost nových projektů, například výstavby zařízení pro energetické využití odpadu, pro mechanicko-biologickou úpravu, překládací stanice, a tím se přiblížit západoevropským standardům. Přínosem této práce je tyto údaje spočítat z již známých reálných dat, čímž posune celý projekt NERUDA kupředu. Nejdřív se ale musíme seznámit s některými základními pojmy.

Nejběžnějším druhem odpadu je SKO. Tento pojem je velice široký, spadá do něj velký počet materiálů, což komplikuje jeho zpracování. Dále budeme čerpat z [7].

1.1 Směsný komunální odpad

Komunální odpad je veškerý odpad vznikající na území obce při činnosti fyzických osob, s výjimkou odpadů vznikajících u právnických osob nebo fyzických osob oprávněných k podnikání. Jedna z jeho složek je SKO, který představuje netříděnou složku komunálního odpadu pocházejícího především z domácností. Tento odpad je sbírán prostřednictvím kontejnerů.

Jeho složení se mění v závislosti na typu zástavby, typu sběru nebo ročního období a právě tyto změny ve složení zásadně ovlivňují jeho následné zpracování. Analýza odpadu tvoří významnou část systému NERUDA. Možné hmotnostní složení v městské zástavbě ukazuje následující tabulka převzatá z [10].

12 %	plasty
13 %	papír a lepenka
7 %	textil
2 %	nápojové kartony
25 %	ostatní biologicky rozložitelné složky
4 %	sklo
2 %	kovy
35 %	ostatní

SKO je možno na třídicích linkách roztřídit na několik složek. Je to lehká frakce (LF) skládající se především z papíru, plastů, PET lahví, nápojových kartonů a textilu. Její podíl je asi 38 %. Kuchyňský odpad spadá do podsítné frakce (PF), jedná se především o rozložitelný odpad a jeho zastoupení je asi 35 %. Do těžké frakce (TF) s asi 21% zastoupením se řadí sklo a zbytek - zejména zemina, kameny, popel a drobné částice. Zvlášť se řadí kovy se 6 %.

Vzhledem k rostoucí produkci odpadu vznikla směrnice [12], která stanovila hierarchii s jeho nakládáním, tu znázorňuje následující obrázek.

předcházení vzniku odpadu
opětovné využití odpadu
recyklace / kompostování
energetické využití odpadu
skládkování

Obrázek 1.1: hierarchie nakládání s odpadem

Nejlepší je samozřejmě předcházení vzniku odpadu, to ale není vždy úplně možné, navíc je to závislé na vůli jednotlivých lidí. Další možností je opětovné využití, příkladem mohou být vratné lahve. Recyklace se vztahuje zejména na kovový odpad. Energeticky zpracovat pak lze odpad, který už nelze ani recyklovat, může to právě biodegradabilní materiál. Nakonec skládkování je dnes nejrozšířenější zejména díky nízkým poplatků za skládkování a nízkým veřejným povědomím. Navíc se doslova mrhá materiálovým a energetickým potenciálem odpadu.

1.2 zpracování SKO

Po sesbírání SKO může začít jeho zpracování. Nejdřív je svezen na jedno místo a pak v závislosti na použité technologii je s ním nakládáno podle jedné z následujících možností. Všechny důležité systémy, které budou níže zmíněny, jsou zahrnuty ve výpočtech nástroje NERUDA.

1.2.1 Mechanicko-biologická úprava

Mechanicko-biologická úprava (MBÚ) je alternativou ke skládkování odpadu. Hlavní myšlenkou je roztřídění SKO do jeho frakcí, s nimiž se pak nakládá samostatně. Zajímavou složkou je LF, která se díky své vysoké výhřevnosti může použít k energetickým účelům - spalování v elektrárnách spolu s uhlím.

Technologie MBÚ je kombinací různých technologických postupů, a to mechanických a biologických. Nejprve je SKO mechanicky třízeno na LF, TF nebo kovy, poté se jeho rozložitelná část (PF) upravuje v biologické části MBÚ. Podstatná část výstupu MBÚ ale stále končí na skládkách.

1.2.2 Energetické využití SKO

Jinou možností, jak naložit se SKO, je energetické využití SKO (EVO), tedy jeho termické zpracování. Je to hospodárná alternativa ke spalování fosilních paliv a spalování SKO spolu s jeho látkovým využitím je schopno zajistit minimalizaci jeho množství. Energetickým využitím se získává elektrická energie nebo teplo a rovněž dochází ke snižování množství vypouštěných skleníkových plynů.

Probíhá v zařízeních, která nazýváme spalovny. Nejdřív se SKO musí z důvodů rozdílné vyhřevnosti a chemického složení homogenizovat, následně je spalováno a uvolněná energie využita k výrobě páry. Ta následně slouží k výrobě elektrické energie nebo k vytápění obytných prostor napojením na rozvodnou síť teplárny. V současné době se v ČR nachází tři zařízení EVO (Praha, Brno, Liberec).

1.2.3 Skládky

Skládkování je nejstarší a technologicky nejméně náročný způsob nakládání se SKO. V současné době se v ČR takto nakládá s 75 % SKO.

2 Optimalizace

Cílem této práce je určení rozptylu nákladů obcí na přepravu a zpracování odpadu. Toho dosáhneme tím, že budeme požadovat minimalizaci nákladů všech obcí, přitom ale budeme simulovat, která z obcí si dřív zajistí zpracovatele v souladu s předpokladem, že bude chtít co nejvíce ušetřit. Hledáním minim za daných omezujících podmínek se zabývá právě optimalizace.

Nejdříve si představíme některé pojmy, tyto budeme čerpat z [1], [2] a [3].

Definice 1. Optimalizační úlohu, neboli úlohu matematického programování zapisujeme ve tvaru minimalizuj

$$f(\mathbf{x})$$

za podmínek

$$g_i(\mathbf{x}) \leq 0, \ i = 1, ..., m$$

kde $\mathbf{x} = (x_1, ..., x_n)^T$ je vektor dimenze n, $x_j, j = 1, ..., n$ jsou tzv. rozhodovací proměnné, $f : R^n \to R$ je účelová funkce a funkce $g_1 : R^n \to R, ..., \ g_m : R^n \to R$ nazveme omezujícími podmínkami (omezeními), souhrně je můžeme zapsat $\boldsymbol{g}(\mathbf{x})$.

Definice 2. Řekneme, že funkce $f : R^n \to R$ má v bodě $\mathbf{x_0}$ *ostré lokální minimum*, jestliže existuje ryzí okolí $\overline{O}(\mathbf{x_0})$ tak, že pro $\forall \mathbf{x} \in \mathrm{Dom} f - \{\mathbf{x_0}\}$, kde $\mathrm{Dom} f$ značí definiční obor funkce f, platí

$$f(\mathbf{x}) - f(\mathbf{x_0}) < 0.$$

Pro neostré lokální minimum nahradíme ostrou nerovnost neostrou nerovností.

Definice 3. Nechť $\boldsymbol{g} : R^n \to R$, hledáme lokální extrémy funkce na množině $M \subset R^n$ určené rovnicemi

$$g_i(\mathbf{x}) = 0, \ i = 1, ..., m, \ \text{kde} \ m < n,$$

kde $g_i(\mathbf{x})$ jsou vazebné podmínky. Ty nazveme *vázanými (lokálními) extrémy.*

Definice 4. Řekneme, že funkce $f : R^n \to R$ má v bodě $\mathbf{x_0}$ *ostré globální minimum*, jestliže pro $\forall x \in \mathrm{Dom} f - \{\mathbf{x_0}\}$ platí

$$f(\mathbf{x}) - f(\mathbf{x_0}) < 0.$$

Ostrý globální extrém hledáme jak na vnitřku $\mathrm{Dom} f$, zde se jedná o ostrý lokální extrím, tak na hranici $\mathrm{Dom} f$ jako vázaný extrém. Ze všech možných vybereme minimum. Pro neostré globální minimum nahradíme ostrou nerovnost neostrou nerovností.

Definice 5. Nutnou podmínkou lokálního i globálního minima v bodě $\mathbf{x_0}$ je splnění rovnic

$$f'_{x_1} = 0, ..., f'_{x_n} = 0,$$

kde $f'_{x_1} = 0$ značíme parciální derovaci funkce f podle proměnné x_1, analogicky pro ostatní proměnné.

Jsou-li funkce $f(\mathbf{x})$ i $\boldsymbol{g}(\mathbf{x})$ lineární, mluvíme o lineárním programování.

2.1 Lineární programování

Definice 6. Nechť a_{ij}, b_i, c_j $(i = 1, 2, ..., m; j = 1, 2, ..., n)$ jsou daná reálná čísla. Úlohu minimalizace funkce

$$\sum_{j=1}^{n} c_j \cdot x_j \tag{2.1}$$

za podmínek

$$\begin{aligned} \sum_{j=1}^{n} a_{ij} \cdot x_j - b_i \leq& \ \ 0 \\ x_j \geq& \ \ 0 \end{aligned} \tag{2.2}$$

nazveme *minimalizační úlohou lineárního programování.*

Definice 7. Množinu $M = \{\mathbf{x} \in R^n | \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0\}$ nazveme *množinou přípustných řešení*, její prvky pak *přípustnými řešeními* úlohy.

Definice 8. Přípustné řešení $\overline{\mathbf{x}} \in M$ nazveme *optimálním řešením* úlohy, jestliže

$$\mathbf{c^T}\overline{\mathbf{x}} \leq \mathbf{c^T x}, \forall \mathbf{x} \in M.$$

Definice 9. *Konvexní polyedrická množina* $M \subset R^n$ je taková množina, kterou lze vyjádřit jako průnik konečného počtu uzavřených poloprostorů. Konvexní polyedrická množina je speciálním případem konvexní množiny.

Definice 10. Množinu $S \subset R^n$ nazveme konvexní množinou, jestliže pro libovolné dva body $\mathbf{x}, \mathbf{y} \in S$ a pro libovolné $\alpha \in (0, 1)$ platí

$$\alpha\mathbf{x} + (1 - \alpha)\mathbf{y} \in S.$$

Definice 11. Nechť $S \subset R^n$ je libovolná množina. Bod $\mathbf{s} \in S$ nazveme *krajním bodem* množiny S, jestliže *neexistují* body $\mathbf{x}, \mathbf{y} \in S$ a číslo $\alpha \in (0,1)$ tak, že $\mathbf{x} \neq \mathbf{y}$ a $s = \alpha\mathbf{x} + (1-\alpha)\mathbf{y}$.

Definice 12. Konvexní polyedrická množina má konečný počet krajních bodů.

Definice 13. Přípustné řešení $\mathbf{x} \in M = \{\mathbf{x} \in R^n | \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$ nazveme *základním řešením* úlohy lineárního programování , jestliže jsou sloupce matice $\mathbf{A}$ s indexy odpovídajícími nenulovým složkám $\mathbf{x}$ lineárně nezávislé.

Věta 14. *Bod* $\mathbf{x} \in M = \{\mathbf{x} \in R^n | \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$ *je krajním bodem množiny* M *právě tehdy, je-li základním řešením.*

Simplexová metoda

Simplexová metoda je nejpoužívanější metodou lineárního programování. Základní myšlenkou této metody je, že množina přípustných řešení lineární úlohy je konvexní polyedrická množina, a tudíž optimální řešení existovat pouze v krajním bodě.

Vybereme proto jeden krajní bod $\mathbf{x_0}$ a spočítáme hodnotu účelové funkce v tomto bodě. Z tohoto krajního bodu vychází konečný počet hran, z nichž každá obsahuje jiný krajní bod nebo je neomezená. Jestliže na některé neomezené hraně existuje bod, ve kterém je účelová funkce menší, potom úloha nemá optimální řešení. V opačném případě hledáme sousední krajní bod s nižší hodnotou účelové funkce než v bodě $\mathbf{x_0}$. Pokud takový bod nenalezneme, optimální řešení je v bodě $\mathbf{x_0}$ a výpočet končí. V opačném případě vyberem bod s nejnižší hodnotou účelové funkce a postup opakujeme.

Výpočet končí po konečném počtu kroků, protože konvexní polyedrická množina má konečný počet krajních bodů, které vyšetřujeme postupně tak, aby došlo k poklesu účelové funkce, každý tedy nejvýše jednou.

2.2 Metoda minimax

Ve srovnání s úlohou lineárního programování nyní budeme uvažovat úlohu minimalizace účelové funkce

$$\max_{k \in K} \sum_{j \in J} c_{jk} \cdot x_j \tag{2.3}$$

za podmínek (2.2). V naší svozové úloze budeme minimalizovat celkové náklady. V tomto celkovém minimálním úhrnu budou některé náklady jednotlivých producentů vyšší a u jiných nižší. Proto budeme hledat takové toky, aby ty nejvyšší náklady byly co nejnižší. Vhodnou možností, jak úlohu řešit, je převést ji na úlohu lineárního programování. Metoda minimax spočívá v zahrnutí přídavné rozhodující proměnné z, která reprezentuje maximální náklady

$$z = \max_{k \in K} \sum_{j \in J} c_{jk} \cdot x_j. \tag{2.4}$$

Aby byl ustanoven tento vztah, musí se přidat následující podmínka.

$$z \geq \sum_{j \in J} c_{jk} \cdot x_j, \ \forall k \in K \tag{2.5}$$

Nyní je z minimalizováno. Doplňující podmínka zajistí, že z bude větší nebo rovno $\sum_{j \in J} c_{jk} \cdot x_j$ pro všechna k, ale zároveň nebude větší než maximum ze všech $\sum_{j \in J} c_{jk} \cdot x_j$, protože je minimalizováno. Proto optimální hodnota z bude jednak tak malá, jak jen je možno, ale i právě rovna největším nákladům přes celé K.

3 Teorie grafů

Teorie grafů je vhodným prostředkem pro popis síťových problémů. Výhodu grafického znázornění využil již v roce 1736 Leonhard Euler při řešení úlohy se sedmi mosty v Královci, dále například Gustav Kirchhoff v roce 1847, kdy právě pomocí grafů ukázal, jak najít lineárně nezávislé rovnice elektrického obvodu, které k jeho řešení postačují. Dosavadní poznatky shrnul až v roce 1936 Dénes Kőnig, čímž se teorie grafů stala matematickou disciplínou.

Nejdříve uvedeme několik základních pojmů, ty budeme čerpat například z [5] a [6].

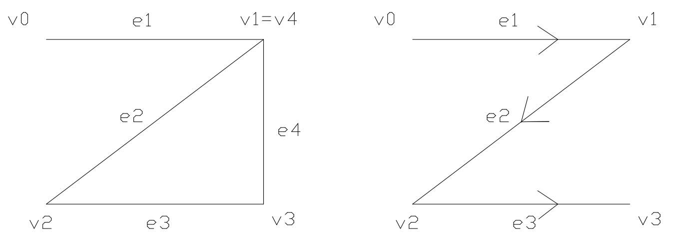
Definice 15. *Neorientovaný graf* je trojice $G = (V, E, \epsilon)$ tvořená neprázdnou konečnou množinou V, jejíž prvky nazýváme *vrcholy*, konečnou množinu E, jejíž prvky nazýváme *neorintovanými hranami* a zobrazením $\epsilon : E \to V^2$, které nazýváme *vztahem incidence*. Toto zobrazení přiřazuje každé hraně $e \in E$ jedno nebo dvouprvkovou množinu vrcholů.

Definice 16. *Orientovaný graf* je trojice $G = (V, E, \epsilon)$ tvořená neprázdnou konečnou množinou V, jejíž prvky nazýváme *vrcholy*, konečnou množinu E, jejíž prvky nazýváme *orientovanými hranami*, a zobrazení $\epsilon : E \to V^2$, které nazýváme *vztahem incidence*. Toto zobrazení přiřazuje každé hraně $e \in E$ uspořádanou množinu vrcholů (x, y). Vrchol x nazýváme *počátečním vrcholem hrany* e a vrchol y nazýváme *koncovým vrcholem hrany* e. Jestliže pro nějakou hranu e je $x = y$, pak hranu e nazýváme (orientovanou) *smyčkou*.

Definice 17. Graf, jehož hrany nebo vrcholy jsou opatřeny číselnými hodnotami, nazýváme *ohodnoceným grafem* nebo *sítí*. Zobrazení, které přiřazuje vrcholů nebo hranám hodnoty, nazýváme *ohodnocením* vrcholů nebo hran. Často se stává, že v jednom ohodnoceném grafu máme několik různých ohodnocení současně.

Definice 18. Posloupnost vrcholů a hran $v_0, e_1, v_1, e_2, ..., e_n, v_n$ nazýváme *(neorientovaný) sled*, jestliže každá hrana e_i této posloupnosti spojuje v_{i-1} a v_i. Jestliže rozlišujeme počáteční a koncový vrchol, hovoříme o *orientovaném sledu*. Ve sledu se vrcholy i hrany mohou opakovat.

Definice 19. Orientovaný (neorientovaný) sled, v němž se žádná hrana neopakuje, nazýváme *orientovaným (neorientovaným) tahem*. Tah, ve kterém se neopakuje žádný vrchol, se nazývá *orientovanou (neorientovanou) cestou*.



Obrázek 3.1: neorientovaný tah, orientovaná cesta

Definice 20. V ohodnocených sledech má často velmi dobrý smysl hovořit o součtu ohodnocení hran v nějakém sledu, tahu nebo cestě. Pro tento součet ohodnocení hran se vžil termín *délka sledu (tahu, cesty)*.

Definice 21. Nechť G je orientovaný graf bez smyček. Zvolíme-li (libovolně, ale pevně) nejen pořadí vrcholů $v_1, ..., v_n$, ale i pořadí hran $e_1, ..., e_m$, můžeme grafu G přiřadit *matici incidence* (též *incidenční matici*) A typu (n, m) předpisem

$$a_{ij} = \begin{cases} 1, & \text{pokud } v_i \text{ je koncovým vrcholem hrany } e_j \\ -1, & \text{pokud } v_i \text{ je počátečním vrcholem hrany } e_j \\ 0, & \text{v ostatních případech} \end{cases} \tag{3.1}$$

Incidenční matice grafu znázorněného na obrázku 3.1 vpravo vypadá následovně.

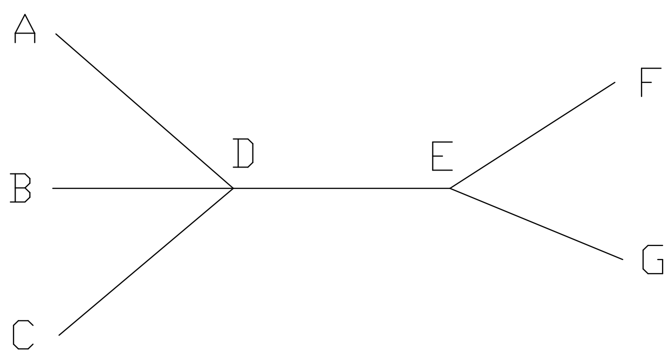
$$A_{ij} = \begin{pmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}$$

4 Modely

V této části se budeme zabývat hledáním toků mezi producentem a zpracovatelem. Vstupem nám bude optimální cena v celé síti a tomu odpovídající toky v jednotlivých hranách. Dosud ale nebylo možné spojit producenta se zpracovatelem. Vědělo se, kolik teče každou hranou, ale ne jakým dílem k celkovému toku hranou přispěl konkrétní producent.

4.1 Model I

Uveďme si motivační příklad. Tři producenti (uzly A, B, C) produkují odpad, ten je odvezen do společného uzlu D, a to každý po své hraně (A-D, B-D, C-D). Odtud se odpad veze dohromady do uzlu E po hraně D-E, kde se musí rozdělit do dvou toků (hrany E-F, E-G) mířících ke zpracovatelům (F, G).



Obrázek 4.1: síť motivačního příkladu

Před popisem motivační úlohy si nejdříve označíme množiny, které budeme dále potřebovat.

I	množina uzlů
$I_z \subset I$	podmnožina producentských uzlů
$I_k \subset I$	podmnožina zpracovatelských uzlů
J	množina hran
$J_k \subset J$	podmnožina hran ústící ke zpracovateli

Svozová úloha, která byla vytvořena v rámci projetku NERUDA, je vyřešena pouze vzhledem k minimálním celkovým nákladům. Víme pouze, v jakém poměru se tok rozdělí v uzlu E, není ale možno zjistit, který z producentů veze odpad do konkrétního místa zpracování. Pro zjištění konkrétního toku odpadu od producenta ke zpracovateli bude nutné úlohu rozšířit o další proměnné a parametry.

Pro potřeby modelu zavedeme dvě proměnné. Maximální náklady bude reprezentovat proměnná w. Dále nás bude zajímat, po které hraně každý producent odpad veze. Pro tyto hodnoty zavedeme vektor u^j, a to každému producentovi jeden. Můžeme potom dohromady psát u^j_{iz} pro $j \in J$ a $iz \in I_z$.

Uzly a hrany je nutné vzájemně provázat. Pro provázání toku odpadu mezi uzly je dána incidenční matice s prvky $A_{i,j}$, ta byla představena v kapitole 3. Dále pak matice s prvky $H_{iz,j}$ určuje, zda odpad daného producenta může vůbec téct po určité hraně. Tato skutečnost se naznačí tím, že se do matice zapíše číslo 1, pokud téct může, nebo číslo 0, pokud nikoliv.

K výpočtu je nutné znát i jisté parametry, jejich výčet je v následujícím seznamu. Parametr x_j je optimálním řešením vypočteným na principu nástroje NERUDA.

g_i	množství odpadu vstupující do modelu v tunách
b_i	kapacita zpracovatelů v tunách
d_i	poplatek za zpracování jedné tuny
c_j	vzdálenosti mezi uzly v kilometrech
f_j	cena za tunokilometr
p_{iz}	váha
x_j	celkové množstí tekoucí hranou v tunách

pro $j \in J$, $i \in I$ a $iz \in I_z$. Novou proměnnou u^j_{iz} je nutno určit pomocí výsledků z předchozí optimalizace, a to těmito omezeními:

$$x_j = \sum_{iz} H_{iz,j} \cdot u^j_{iz}, \ j \in J, \tag{4.1}$$

$$u^j_{iz} \leq H_{iz,j} \cdot x_j, \ j \in J, \tag{4.2}$$

$$g_{iz} = \sum_{j} u^j_{iz}, \ j \in J_k, \tag{4.3}$$

pro $iz \in I_z$.

Omezení (4.1) říká, že součet toků odpadu konkrétních producentů, pro které je hrana j dosažitelná, je roven celkovému toku v této hraně. Omezení (4.2) určuje, že producent může po dosažitelné hraně vézt nejvýše takové množství odpadu, které bylo dané hraně přiděleno. Nakonec, omezení (4.3) značí, že ke zpracovatelům vtéká přesně takové množství, jako bylo vyprodukováno. Navíc nejvyšší hodnotu proměnné u^j_{iz} nastavíme na hodnotu g_{iz}.

Jak bylo řečeno dříve, přiřazení toku ke konkrétnímu producentovi je optimalizační úloha. Hledáme

$$w = \max_{iz \in I_z} [\sum_j \sum_i u^j_{iz} \cdot d_i + \sum_j u^j_{iz} \cdot c_j \cdot f_j] \tag{4.4}$$

a tuto hodnotu budeme následně minimalizovat. První člen značí cenu za zpracování, druhý cenu za přepravu, jejich součtem dostaneme celkové náklady. Tímto dosáhneme toho, že producentovi s největšími možnými náklady na převoz a zpracování odpadu bude vybrána cena nižší. V souladu s popsanou metodou minimax, viz odstavec 2.2, úlohu přepíšeme do tvaru minimalizuj w za podmínek (4.1), (4.2), (4.3) a

$$w \geq \sum_j \sum_i u_{iz}^j \cdot d_i + \sum_j u_{iz}^j \cdot c_j \cdot f_j,\ \forall iz \in I_z. \tag{4.5}$$

Reálně ale není možno nařídit obcím, aby posílali odpad do námi určené spalovny, proto se zaměříme na to, zda včasný výběr zpracovatele může přinést něktrému producentovi výraznou úsporu nákladů. Zavedením vah do účelové funkce simulujeme pořadí producentů při výběru zpracovatele, díky čemuž je možné analyzovat změnu nákladů v důsledku chování okolních producentů. Opakováním výpočtu s měnícími se vahami můžeme určit variabilitu nákladů. Upravená podmínka (4.4) vypadá následovně.

$$w \geq p_{iz} \cdot [\sum_j \sum_i u_{iz}^j \cdot d_i + \sum_j u_{iz}^j \cdot c_j \cdot f_j],\ \forall iz \in I_z. \tag{4.6}$$

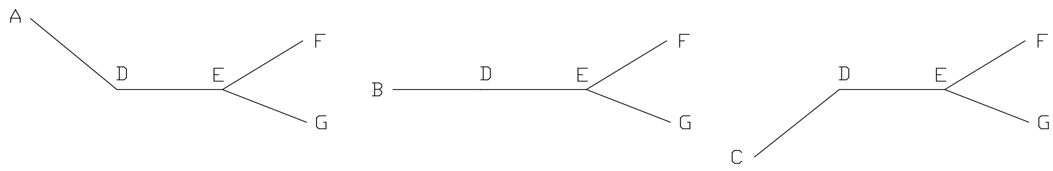
Tento model nelze použít na obecné síti. Na síti jednoduchého charakteru funguje bez obtíží. Problém nastává, pokud jsou producenti propojeni hranami. Matice $H_{iz,j}$ by byla plná jedniček, toky by se tedy opět slily. Danými omezeními navíc lze určit historii toku pouze jednu hranu za uzel, ve kterém se toky rozdělí. Další věcí je nutnost generovat další matici. V nasledující kapitole proto představíme přístup, který je možno aplikovat i na složitější sítě.

4.2 Model II

Následující model je založel na jiném principu. Předpokládá, že v celé síti je pouze jeden producent. Sítí je tedy stejný počet jako producentů, protože každý z nich má svou vlastní síť, která je ale shodná se všemi ostatními sítěmi až na polohu producenta. V dalším popisu se budeme odkazovat na síť z obázku 4.1.

Pro producenta A platí $\sum_j A_A \cdot u_A^j = v_A^i$. O vektoru v se zmíním později. Incidenční matice $A_{i,j}$ v tomto případě musí být upravena, a to vynecháním řádku s producenty B a C a hran k nim vedoucím, B-D a C-D. Analogicky se tato operace dá provést pro každého producenta.

Vektor v_A^i připadající producentovi A je zaveden jako proměnná, ačkoliv většinu jeho složek známe. Do složky, která náleží producentovi, vložíme množství odpadu, které daný producent vyprodukuje, jen se znaménkem mínus. Dáváme tím najevo, že odpad od producenta odtéká. Pro uzly, kudy odpad pouze protéká, se zadá nulová hodnota. Produceni B a C nejsou v upravené síti vůbec zahrnuti. Jediné neznámé hodnoty jsou v uzlech zpracovatelů a jsou to hodnoty námi hledané. Značí totiž, jaké množství odpadu doteklo ke každému zpracovateli právě od producenta A.



Obrázek 4.2: myšlenka modelu II

Vektor u_A^j již známe, byl představen v modelu I. Zapisováno je do něj množství odpadu producenta A, které teče určtiou hranou. Vynechány jsou v něm hrany B-D a C-D, tím je naznačena nedosažitelnost těchto hran.

Jak bylo naznačeno, je nutné upravit incidenční matici $A_{i,j}$ a to odebráním dvou hran a dvou uzlů. Druhý možný přístup by byl zachovat incidenční matici ve stejném tvaru, ale rozšířit vektory u_A^j, v_A^i tak, aby obsahovaly všechny hrany a uzly, ale aby hodnoty v těchto složkách byly nulové.

Názorně si nejdříve ukážeme na síti motivačního příkladu, jak vektory v_{iz}^i vypadají. Ve známých složkách jsou vypsány hodnoty, v ostatních ponecháme obecné označení. Pro přehlednost jsou vektory spojeny do společné matice, uzly jsou seřazeny abecedně shora dolů.

$$v_{iz}^i = \begin{pmatrix} -g_A & 0 & 0 \\ 0 & -g_B & 0 \\ 0 & 0 & -g_C \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ v_A^F & v_B^F & v_C^F \\ v_A^G & v_B^G & v_C^G \end{pmatrix}$$

Zavedeme nyní další vektor proměnných ω.

$$\omega_i = \sum_j A_{i,j} \cdot x_j. \tag{4.7}$$

Vzhledem k tomu, že incidenční matice $A_{i,j}$ i vektor toků v hranách x_j jsou známé parametry, nový vektor proměnných ω není problém určit. Ten nám ale pomůže s určením hodnot vektorů v_{iz}^i.

Vektor ω popisuje bilanci odpadu v každém uzlu sítě. Pokud v některém z uzlů odpad přibyde, tedy víc odteče než přiteklo, ω_i bude záporná, pokud naopak ubyde, bude hodnota ω_i kladná. Pokud odpad uzlem pouze proteče beze změny množství, hodnota ω_i bude nulová.

Vazba mezi ω_i a v^i_{iz} je následující:

$$\omega_i = \sum_{iz} v^i_{iz}. \tag{4.8}$$

Pro hodnoty složek vektorů v^i_{iz} platí stejná pravidla jako pro vektor ω, pouze na rozdíl od vektoru ω jsou vztaženy pouze na konkrétního producenta. Tím se přes modifikovanou soustavu

$$\sum_j A_{i,j} \cdot u^j_{iz} = v^i_{iz} \tag{4.9}$$

určí existující cesty, kudy může odpad téci. Nejdříve ale musíme ty správné složky vektoru v^i_{iz} položit rovny nule. To provedeme následující úvahou.

Budeme požadovat, aby

$$\begin{aligned} v^i_{iz} &\leq 0 \text{ pro } i \in \overline{I_k}, \\ v^i_{iz} &\geq 0 \text{ pro } i \in \overline{I_z}, \end{aligned} \tag{4.10}$$

kde $\overline{I_k}$ (popřípadě $\overline{I_z}$) znamená doplněk množiny I_k (popřípadě I_z). Tímto určíme, že uzly, které nejsou ani producentské ani zpracovatelské, budou mít hodnotu vektoru v^i_{iz} rovnu 0, tedy že v nich odpad "nevzniká"ani "nezaniká". Producenské uzly budou mít hodnotu menší než 0, což říká, že v těchto uzlech odpad může "vzniknout", analogicky pro uzly zpracovatelské.

Stačí jen určit, aby ve v^i_{iz} byly v producentských uzlech nuly až na jeden. V tuto chvíli nám stačí uvažovat vektory v^i_{iz} pouze v pozicích počátečních uzlů. Chceme, aby pro $iz, \widetilde{iz} \in I_z$ platilo

$$\begin{aligned} v^{\widetilde{iz}}_{iz} &= -g_{iz}, \text{ pokud } iz = \widetilde{iz}, \\ v^{\widetilde{iz}}_{iz} &= 0, \text{ pokud } iz \neq \widetilde{iz}. \end{aligned} \tag{4.11}$$

Ke správnému fungování modelu postačuje zadat pouze první z podmínek rovnicí

$$v^{iz}_{iz} = -g_{iz}, \tag{4.12}$$

splnění druhé podmínky zaručuje rovnice (4.8) v kombinaci s podmínkou vyjádřenou v (4.10).

Vzhledem k tomu, že se změnily některé proměnné, je třeba také upravit podmínku minimaxu. Ta je nyní ve tvaru:

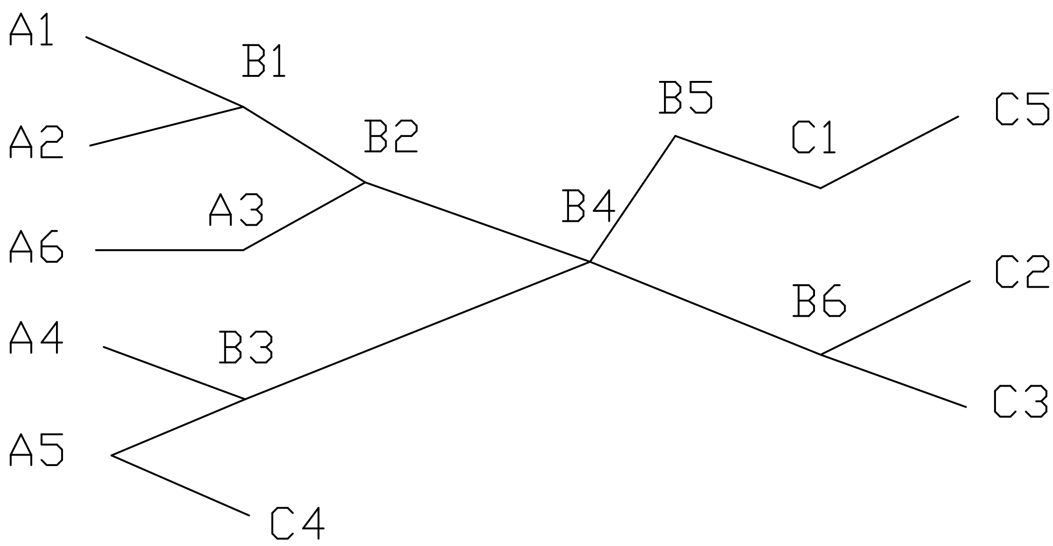
$$w \geq p_{iz} \cdot (\sum_i v^i_{iz} \cdot d_i + \sum_j u^j_{iz} \cdot c_j \cdot f_j) \tag{4.13}$$

a w je minimalizována pro $\forall iz \in I_z$. Celkové náklady pro každého producenta zvlášť dostaneme vynecháním vah.

$$z_{iz} = \sum_i v^i_{iz} \cdot d_i + \sum_j u^j_{iz} \cdot c_j \cdot f_j. \tag{4.14}$$

Závěrem dodejme, že pro výpočet tohoto modelu je potřeba pouze incidenční matice a pozice uzlů producentů a zpracovatelů, což jsou prvky zavedené ve výpočetním nástroji NERUDA. Není ani nutno dalších složitých generování jiných matic, ani určování dalších podmnožin, ty popsané v (4.10) vytvoří GAMS sám. Jediné, co se musí zadat, jsou hodnoty vektoru x_j a g_{iz}, pro výpočet účelové funkce i celkových nákladů jsou potřeba parametry c_j, f_j a d_i. Všechna tato data je možno načíst z externích souborů.

Ještě před jejich aplikací na velkou síť byly oba modely testovány na testovací síti o 17 uzlech, u které byla snaha se přiblížit síti obecné. A to zařazením dvou producentů a zpracovatelů za sebe, rozdělením toku již v uzlu zpracovatele, více míst rozdělení nebo uzly navíc mezi zpracovatelem a uzlem rozdělení toku. Na této síti byly ukázány nedostatky modelu I a správné fungování modelu II. Uzly obsahující písmeno A jsou producentské, B mezilehlé a C zpracovatelské.



Obrázek 4.3: testovací síť

5 Aplikace modelu na reálné síti

Nyní je fungující model hotov a v konečné fázi ho můžeme aplikovat na větší síťový problém. V rámci projektu NERUDA byla vytvořena síť mapující Českou republiku a je dále rozšiřována o Spolkovou republiku Německo. Tato síť obsahuje 213 uzlů - producentů a 190 uzlů představující zpracovatele - spalovny, skládky, překládací stanice nebo MBÚ. Uzly jsou spolu spojeny přes 2000 silničními hranami a 6000 hranami železničními. Navíc rozlišuje,co se po hraně přepravuje, zda ještě nezpracované SKO nebo některá z frakcí poté, co se SKO zpracuje.

Všechna vstupní data jsou zpracována projektem NERUDA. Byly zjištěny produkce odpadu ve všech zavedených producentských uzlech, dále zjištěny kapacity a poplatky za zpracování zařízení pro zpracování odpadu, popřípadě u zařízení, pro které jsou výpočty zjišťována rentabilita, tento údaj výpočtem navrhnut. Dodejme, že kapacity pro náš výpočet nepotřebujeme, jejich velikost je zjištěna z toků po hranách. Byly určeny i vzdálenosti mezi jednotlivými uzly a ceny za přepravu pro každou frakci.

Náš výpočetní nástroj ale není připraven pro vícekomoditní přepravu, proto bylo přistoupeno k zjednodušení. Znevýhodněním vozby ostatních frakcí a železniční dopravy dosáhnene toho, že budeme vozit pouze SKO a to pouze po silnici. Před samotnou aplikací ale provedeme v modelu pár změn.

První změna je pouze kosmetického rázu. Cena za přepravu odpadu je pro každou hranu stejná, takže parametr f_j se stane parametrem skalárním a budeme ho značit f.

Druhá změna je ale poněkud složitější. Při přenosu dat a jeho zpracování totiž dochází k zaokrohlovacím chybám, díky čemuž má výpočtový software problém s řešitelností modelu. Pro jejich kompenzaci zavedeme další proměnnou δ_{iz} a upravíme rovnici (4.12) do tvaru

$$v_{iz}^{iz} = -g_{iz} + \delta_{iz}, \tag{5.1}$$

díky čemuž se mírný nesoulad ve vstupních datech může vložit do δ_{iz} a umožnit tak vyřešení celého modelu. Na stranu druhou ale musíme zaručit, aby hodnota δ_{iz} byla opravdu jen v mezích zaokrouhlovacích chyb a její velikost zanedbatelná vůči hodnotě v_{iz}^{iz}. Minimalizaci jejich hodnot provedeme v účelové funkci, kde δ_{iz} bude jejím dalším členem. Její tvar zapíšeme

$$w \geq p_{iz} \cdot (\sum_i v_{iz}^i \cdot d_i + \sum_j u_{iz}^j \cdot c_j \cdot f) + P \cdot \delta_{iz},\ \forall iz, \tag{5.2}$$

kde P je číslo o několik řádů větší než hodnoty ostatních členů účelové funkce. Tímto je proměné δ_{iz} přidělena nejvyšší váha, a proto bude její hodnota nejvíce snižována.

Poslední změnou, která je opět spíše kosmetická, je využití faktu, že v síti NERUDA platí, že

$$\begin{aligned} \overline{I_k} &= I_z, \\ \overline{I_z} &= I_k, \end{aligned} \tag{5.3}$$

proto vztahy (4.10) můžeme zapsat jednodušeji ve tvaru

$$\begin{aligned} v_{iz}^i &\leq 0 \text{ pro } i \in I_z, \\ v_{iz}^i &\geq 0 \text{ pro } i \in I_k. \end{aligned} \tag{5.4}$$

5.1 Výsledky výpočtů

Výpočet modelu proběhl 100krát, v každé iteraci byly programem GAMS generovány váhy z rovnoměrného rozdělení mezi hodnotami 1 a 10 pro každého producenta. Jak bylo dříve řečeno, simulovala se tím včasnost objednávky zpracovatele, kdo se zajímal dříve, mohl si vybrat pro něj nejlevnější nabídku.

Pro prezentaci přínosu práce vybereme 2 města vzdálená od sebe asi 25 km přímou čarou. Jedná se o Klášterec nad Ohří a Žatec ležící v severních Čechách. Jejich společným znakem jsou průměrné roční náklady na svoz a zpracování odpadu ve výši okolo 10 milionů korun českých. Hlavní rozdíl je v rozpětí nákladů. Pro město Žatec vyšla směrodatná odchylka nákladů necelých 3 000 korun. Tuto hodnotu při neměnných vstupech lze připsat na vrub právě numerickým chybám a je oproti průměrné ceně zanedbatelná, variační koeficient je 0,02 %. Naproti tomu směrodatná odchylka Klášterce nad Ohří dosahuje téměř 1,5 milionů. Záleží totiž na tom, kam bylo SKO posláno, zda všechen do mostecké spalovny, v tomto případě by zaplatil přibližně 8 milionů, nebo se část SKO musí odvést do Ústí nad Labem, kdy se náklady vyšplhají až k 12 milionům. Pro město velikosti Klášterce nad Ohří by to byl výrazný zásah do rozpočtu. Naproti tomu pro Žatec byl za daných podmínek vždy vybrán pouze jediný zpracovatel, a to navrhovaná spalovna v Mostě.

Obrázek 5.1: histogram nákladů pro Klášterec nad Ohří

Nutno ale dodat, že celý výpočet probíhal pouze na jednom scénáři, předpokládaly se tedy neměnné faktory, které náklady ovlivňují. Jsou to například ceny za zpracování, produkce odpadu, výhodné nabídky ze zahraničí. Výsledky by byly jistě rozmanitější, pokud by byly zahrnuty všechny vlivy.

6 GAMS

Všechny výpočty probíhaly v programu GAMS (The General Algebraic Modeling System). Tento software je velmi pokročilý modelovací systém s vlastním programovacím jazykem vhodným k pro řešení rozsáhlých optimalizačních úloh. Obsahuje řešiče nejen pro lineární programování, ale i pro úlohy nelineární, celočíselné nebo binární. My budeme využívat pouze lineární řešič CPLEX, který využívá simplexovou metodu.

Syntax navíc velmi není příliš složitá a umožňuje i základní programovací techniky, například použití cyklů, podmínek a omezení, a není case sensitive.

Nejdříve si ukážeme prostředí GAMS a jak se v něm pracuje. Popis provedeme na modelu II aplikovaného na testovací síť. Celý kód si pak můžete prohlédnout v příloze A.

```
execseed = 1 + gmillisec(jnow);
set
s scénáře /s1*s5/
i uzly /A1*A6, B1*B6, C1*C5/
j hrany /A1-B1, A2-B1, A3-B2, B1-B2, A4-B3, A5-B3,
         B2-B4, B3-B4, B4-B5, B4-B6, B5-C1, B6-C2, B6-C3,
         A6-A3, A5-C4, C1-C5/
iz(i) producenti /A1*A6/
ik(i) zpracovatelé /C1*C5/
izc(i) vše bez producentů
ikc(i) vše bez zpracovatelů
;
```

Zcela nahoře je příkaz *execseed*, ten se vztahuje ke generování náhodných čísel. Pod ním se nachází příkaz *set*, který definuje množiny a její prvky. Nejdříve se napíše název množiny, kterým se budeme dále v programu na ni odkazovat, dále si k ní můžeme napsat libovolnou poznámku, ta je napsána modře. Mezi lomítky jsou pak vypsány všechny prvky dané množiny. Ty mohou být zapsány jednotlivě, jak jsou vypsány množiny hran, nebo ve zkráceném tvaru, jak je možno vidět u množiny uzlů. Zápis $/A1 * A6/$ je tedy ekvivalentí s $/A1, A2, A3, A4, A5, A6/$. Podmnožina se vytvoří stejně jako množina, pouze za její název se v závorce dopíše název její nadmnožiny. Prvky podmnožiny je možno také přiřadit dále v kódu logickými operacemi.

Pro méně rozsáhlé úlohy je možno zapsat maticový parametr do tabulky, zde kvůli velikosti nebude znázorněno.

```
parameter
g(i) mnozstvi na vstupu
/
```

```
         A1 10,
         A2 15,
         A3 10,
         A4 15,
         A5 15
         A6  5    /
x(j) mnozstvi na hrane
$CALL gdxxrw.exe hrany1.xlsx par=x rng=hranit Cdim=1
$GDXIN hrany1.gdx
$LOAD x
$GDXIN
;
```

Dále se zapisují parametry. Jejich zápis je velice podobný zápisu množin. K názvu parametru se zapíše, ke které množině náleží, a k jejim prvkům se následně dopíší hodnoty, kterých nabývají. Dole je ukázka načtení parametru ze souboru, o té bude řeč dále. Klíčovým slovem *scalar* se zavádí skalární parametr, pravidla jsou stejná jako u parametrů vektorových.

V dalším si ukážeme zavedení rovnic a proměnných.

```
positive variable
                 u(j,iz) množství dodavatele na hraně
                 z(iz)   celková cena
;
variable
                 v(i,iz) vektory pravé strany
                 bb(i)   využitá kapacita
;

variable
                 w       optimalizovaná hodnota
;



equation
  plneni(i)              hodnota využití kapacity(komplexně)
, vztah1(i)              v1 + v2 + v3 = bb
, vztah3(j)              u1 + u2 + u3 = x
, vztah5(i,iz)           Au=v
, spec1(iz)              zavedení hodnot vstupů
, ucelfce                účelová funkce s vahami
, ceny                   celková cena pro každého producenta
;

plneni(i)   ..  bb(i)   =e=   sum(j, A(i,j)*x(j) );

vztah1(i)   ..  bb(i)   =e=   sum(iz, v(i ,iz));
vztah3(j)   ..  x(j)    =e=   sum(iz, u(j ,iz));
vztah5(i,iz)..  v(i,iz) =e=   sum(j, A(i,j)*u(j,iz)) ;
```

```
spec1(iz)..      g(iz)   =e= - v(iz,iz);

ucelfce(iz)..    w =g=   p(iz)*(
                         sum(i, v(i,iz)*d(i) ) +
                         sum(j, u(j,iz)*c(j)*f(j) )
                         );

ceny(iz)..       z(iz) =e=   sum(i, v(i,iz)*d(i) ) +
                             sum(j, u(j,iz)*c(j)*f(j) );
```

Klíčové slovo *variables* zavede proměnnou. Ta je opět dána svým názvem a množinami. Příkazem *positive* je myšleno, že hodnoty proměnné budou nabývat nezáporných hodnot, analogicky funguje příkaz *negative*.

Rovnice, ale i nerovnice, se nejdřív všechny vypíšou pod příkaz *equation*, k jejich názvu se opět může doplnit bližší textový popis.

Dále už jen následuje příkaz pro řešení, popřípadě výpisy do souborů, zobrazení výsledných hodnot proměnných ve výpisu o řešení úlohy.

Jeho značnou výhodou je fakt, že umí nahrávat i zapisovat data z (do) tabulkových souborů MS Excel. Toho lze využít při konstrukci rozsáhlejších parametrů, které jsou nutné k výpočtu. Jejich tvorba je v prostředí k tomu určenému jednodušší a přehlednější.

6.1 Import a export dat

Jednou z výhod progamu GAMS je možnost importovat i exportovat parametry do tabulkého souboru MS Excel. To je výhodné z pohledu přenášení dat mezi původní úlohou a modelem, o kterém pojednává tato práce. Obojí můžeme provádět pomocí nástroje GDXXRW. Vyžadován je pouze nainstalovaný MS Excel na počítači, na kterém je používán. Zde budeme čerpat z [11].

Pro export je syntax následující:

$$\mathsf{execute\_unload\ 'hrany.gdx'\ \ x.L}$$
$$\mathsf{execute\ 'gdxxrw.exe\ \ hrany.gdx\ \ var = x.L'}$$

První příkaz založí soubor *hrany.gdx* (GAMS data exchange), do kterého vepíše hodnotu proměnné x_j. Přípona L značí, že načtena bude poslední - a tedy výsledná - hodnota této proměnné. Druhý příkaz spustí nástroj GDXXRW a převede GDX soubor na soubor stejného názvu ale s příponou *.xlsx*. Příkaz *var* znamená, že hodláme exportovat proměnnou, v případě parametru jde o příkaz *par*. Po otevření generovaného soubrou *hrany.xlsx* uvidíme, že do prvního řádku jsou zapsány prvky $j \in J$, v řádku druhém pak odpovídající hodnoty x_j. Dodejme, že tento proces je součástí výpočtové fáze.

Při importu parametru ze souboru *hrany.xlsx* je syntax složitější. Navíc se v importovaném souboru musí označit oblast obsahující daný parametr i s prvky, kterým odpovídají.

Tato oblast je v tomto případě nazvána *hranit*.

```
x(j) prevezene mnozstvi
$CALL GDXXRW.EXE hrany.xlsx  par = x  rng = hranit  Cdim = 1
$GDXIN hrany.gdx
$LOAD x
$GDXIN
```

V prvním řádku zavádíme parametr x_j i s popisem. Druhý řádek volá nástroj GDXXRW, který načte oblast *hranit* v souboru *hrany.xlsx* a vytvoří odpovídající soubor *hrany.gdx*. Příkaz *Cdim* se používá, pokud jsou hodnoty v řádku, v případě sloupcového zapsání se použije příkaz *Rdim*, při načítání maticového parametru se příkaz zcela vynechává. Imporotovat je možno i množinu, v tom případě se načítají pouze prvky množiny a klíčové slovo je *set* namísto *par*. V dalším řádku říkáme, že data budou načítána z nově vytvořeného GDX souboru, v dalším se přikazuje načtení. Posledním řádkem ukončíme načítání. Jen doplňme, že načítání dat je prováděno v kompilační části.

Závěr

V této práci jsem se blíže seznámili s alternativními možnostmi nakládání s odpady a i důvody, proč by skládkování nemělo být nejčastější způsob zpracování odpadu. Dále jsme sestavili matematický model, který umožňuje zjistit náklady každé obce, a nazvali ho *model II*. Jako vstupní parametry nám posloužili data shromážděná v rámci projektu NERUDA a výsledky jeho výpočtů.

Díky tomuto přídavného výpočtu k nástroji NERUDA je možno analyzovat variabilitu nákladů na svoz a zpracování odpadu obcí s rozšířenou působností jako producenty odpadu. Což může být přínosné i pro samotné producenty, a to z pohledu plánování i navazování spolupráce s ostatními producenty. Dále nástroj může sloužit jako cílená argumentace nových zařízení pro energetické využití odpadu směrem k producentům.

Literatura

[1] KLAPKA, J., DVOŘÁK, J., POPELA, P.: *Metody operačního výzkumu*, druhé vydání, Nakladatelství VUTIUM, Brno, 2001

[2] HRABEC, D.: *Optimalizace inženýrského návrhu*, bakalářská práce, Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2009, 42 s.

[3] KUREŠ, M.: *Matematická analýza II* [online]. [cit. 2014-05-16]. Dostupné z: http://mathonline.fme.vutbr.cz/Matematicka-analyzanbspII/sc-1226-sr-1-a-241/default.aspx

[4] BISSCHOP, J.: *AIMMS Optimization Modeling*, Paragon Decision Technology, 2012

[5] PLESNÍK, J.:*Grafové algoritmy*, první vydání, VEDA, Vydavatel'stvo Slovenskej akadémie veid, Bratislava, 1983

[6] DEMEL, J.: *Grafy a jejich aplikace*, první vydání, Nakladatelství Academia, Praha, 2002

[7] UCEKAJ, V.: *Analýza možností nakládání s komunálními odpady v rámci mikroregionu*, disertační práce, Vysoké učení technické v Brně, Fakulta strojního inženýrství. 2010. 153 s.

[8] Energetické využívání odpadů. [online]. [cit. 2014-05-20]. Dostupné z: http://www.evokomorany.cz/index.php/technologie/energeticke-vyuzivani-odpadu

[9] MOLLIKOVÁ, E. *Modelovací jazyky v optimalizaci*, bakalářská práce, Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2010, 54 s.

[10] PROCHÁZKA, V. *Matematický model dopravní úlohy pro oblast odpadového hospodářství.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2012. 43 s. Vedoucí bakalářské práce Ing. Martin Pavlas, Ph.D.

[11] GAMS/Excel Using GDX [online][cit. 2014-05-01].
Dostupný z http://www.gams.com/docs/excel/

[12] *Směrnice Evropského parlamentu a Rady 2008/98/ES ze dne 19. listopadu 2008 o odpadech a rušící některé směrnice.* 2008. Úřední věstník Evropské unie L312/3.

[13] ŠOMPLÁK, R.; FERDAN, T.; POPELA, P.; PAVLAS, M. Waste-to- energy facility planning under uncertain circumstances. *Applied Thermal Engineering,* 2013, roč. 61, č. 1, s. 106-114. ISSN: 1359- 4311.

Příloha A

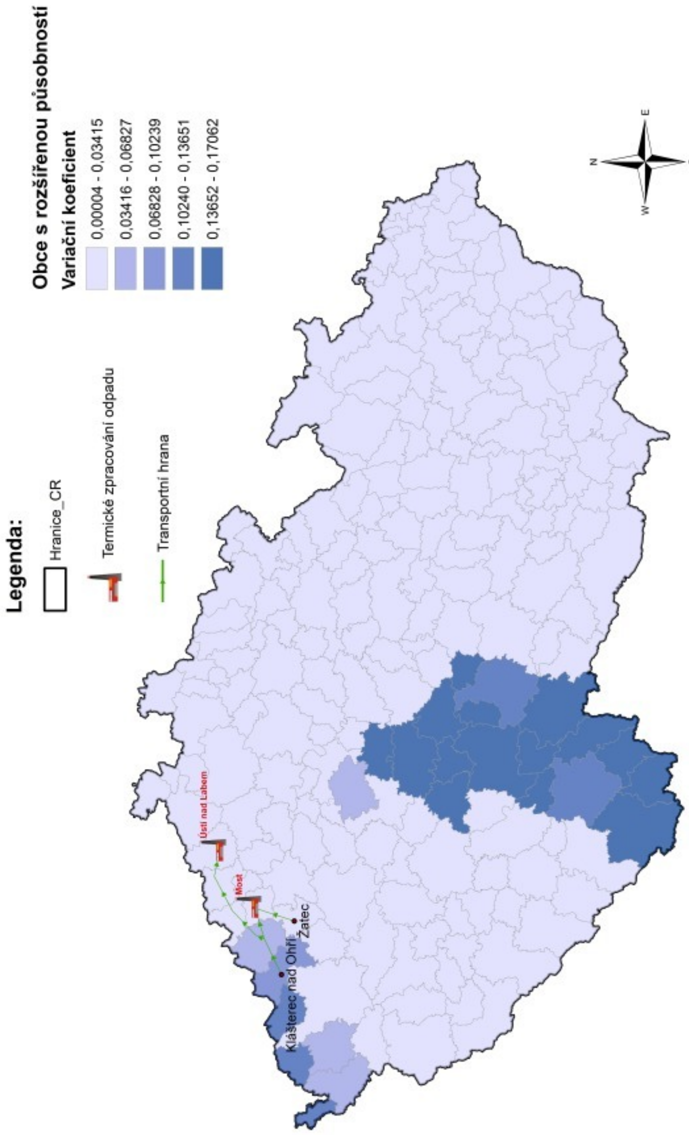
```
execseed = 1 + gmillisec(jnow);

set
s scénáře /s1*s1/
i uzly /A1*A6, B1*B6, C1*C5/
j hrany /A1-B1, A2-B1, A3-B2, B1-B2, A4-B3, A5-B3,
         B2-B4, B3-B4, B4-B5, B4-B6, B5-C1, B6-C2, B6-C3,
         A6-A3, A5-C4, C1-C5/
iz(i) producenti /A1*A6/
ik(i) zpracovatelé /C1*C5/
izc(i) vše bez producentů
ikc(i) vše bez zpracovatelů
;

table
A(i,j) incidenční matice
     A1-B1 A2-B1 A3-B2 B1-B2 A4-B3 A5-B3 B2-B4 B3-B4
A1   -1
A2         -1
A3               -1
A4                           -1
A5                                -1
A6
B1    1     1          -1
B2                1     1                -1
B3                            1    1           -1
B4                                        1     1
B5
B6
C1
C2
C3
C4
C5


+ B4-B5 B4-B6 B5-C1 B6-C2 B6-C3 A6-A3 A5-C4 C1-C5
A1
A2
A3                                  1
A4
A5                                       -1
A6                               -1
B1
B2
B3
B4 -1    -1
```

```
B5  1          -1
B6        1          -1    -1
C1              1                                   -1
C2                    1
C3                          1
C4                                      1
C5                                            1
;
parameter
g(i) mnozstvi vstupu
/
         A1 10,
         A2 15,
         A3 10,
         A4 15,
         A5 15
         A6  5    /
b(i) mnozstvi vystupu
/
         C1 20,
         C2 20,
         C3 20
         C4  5
         C5  5    /
c(j) vzdalenosti
/
         A1-B1 20,
         A2-B1 15,
         A3-B2 35,
         B1-B2 20,
         A4-B3 30,
         A5-B3 35,
         B2-B4 20,
         B3-B4 30,
         B4-B5 15,
         B4-B6 20,
         B5-C1 30,
         B6-C2 15,
         B6-C3 20,
         A6-A3  5,
         A5-C4 10,
         C1-C5 10    /
d(i) poplatek na vystupu
/
         C1 30,
         C2 20,
         C3 20
         C4 20
         C5 10     /
```

```
f(j) poplatek za vlkm
/
         A1-B1 45,
         A2-B1 45,
         A3-B2 35,
         B1-B2 45,
         A4-B3 40,
         A5-B3 35,
         B2-B4 45,
         B3-B4 45,
         B4-B5 35,
         B4-B6 45,
         B5-C1 30,
         B6-C2 40,
         B6-C3 35,
         A6-A3 30,
         A5-C4 30,
         C1-C5 30       /

x(j) mnozstvi na hrane
$CALL gdxxrw.exe hrany1.xlsx par=x rng=hranit Cdim=1
$GDXIN hrany1.gdx
$LOAD x
$GDXIN

p(iz) vahy
;

positive variable
                 u(j,iz) množství dodavatele na hraně
                 w(iz)   celková cena
;
variable
                 v(i,iz) vektory pravé strany
                 bb(i)   využitá kapacita
;

variable
                 z       optimalizovaná hodnota
;


equation
  plneni(i)              hodnota využití kapacity(komplexně)
, vztah1(i)              v1 + v2 + v3 = bb
, vztah3(j)              u1 + u2 + u3 = x
, vztah5(i,iz)           Au=v
, spec1(iz)              zavedení hodnot vstupů
```

```
, ucelfce                účelová funkce s vahami
, ceny                   celková cena pro každého producenta
;

plneni(i)   ..   bb(i)   =e=   sum(j, A(i,j)*x(j) );

vztah1(i)   ..   bb(i)   =e=   sum(iz, v(i ,iz));
vztah3(j)   ..   x(j)    =e=   sum(iz, u(j ,iz));
vztah5(i,iz)..   v(i,iz) =e=   sum(j, A(i,j)*u(j,iz)) ;

spec1(iz)..      g(iz)   =e= - v(iz,iz);

ucelfce(iz)..    z =g=   p(iz)*(
                         sum(i, v(i,iz)*d(i) ) +
                         sum(j, u(j,iz)*c(j)*f(j) )
                         );

ceny(iz)..       w(iz) =e=   sum(i, v(i,iz)*d(i) ) +
                             sum(j, u(j,iz)*c(j)*f(j) );

                         ikc(i) = not ik(i);
                         izc(i) = not iz(i);

                         v.up(ikc,iz) = 0;
                         v.lo(izc,iz) = 0;

                         u.up(j,iz) = g(iz);

model dvojka5 /all/;

loop(s,
p(iz)=uniform(1,10);
solve dvojka5 using lp minimizing z;
);
```

Příloha B



Obrázek 1: variační koefiecint ORP