



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

ÚSTAV MATEMATIKY

FACULTY OF MECHANICAL ENGINEERING

INSTITUTE OF MATHEMATICS

MODELY MATEMATICKÉHO PROGRAMOVÁNÍ PRO  
SMĚŠOVACÍ ÚLOHY

MATHEMATICAL PROGRAMS FOR BLENDING PROBLEMS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. VÍT KALENSKÝ

VEDOUCÍ PRÁCE

SUPERVISOR

RNDr. PAVEL POPELA, Ph.D.

BRNO 2018



## Zadání diplomové práce

Ústav: Ústav matematiky  
Student: **Bc. Vít Kalenský**  
Studijní program: Aplikované vědy v inženýrství  
Studijní obor: Matematické inženýrství  
Vedoucí práce: **RNDr. Pavel Popela, Ph.D.**  
Akademický rok: 2017/18

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 diplomové práce:

### **Modely matematického programování pro směšovací úlohy**

#### **Stručná charakteristika problematiky úkolu:**

Diplomant si prohloubí znalosti problematiky modelů matematického programování se zaměřením na problematiku efektivní modifikace, transformace a implementace složených modelů. Téma práce navazuje na dlouhodobě rozvíjenou problematiku optimalizačních modelů pro alokační a směšovací úlohy modelované pomocí dvojstupňových úloh stochastického programování. Dosažené teoretické a algoritmické výsledky budou aplikovány pro vybranou třídu pokročilých inženýrských problémů.

#### **Cíle diplomové práce:**

1. Prezentace vybraných typů optimalizačních úloh.
2. Vytvoření matematických optimalizačních modelů pro kombinované směšovací a logistické úlohy různé úrovně složitosti pro vybranou aplikační oblast.
3. Prostudování vlastností modelů v kontextu vybrané třídy optimalizačních úloh.
4. Modifikace a implementace vhodných algoritmů řešení s využitím vybraných modelovacích nástrojů.
5. Realizace výpočtů s využitím testovacích dat, analýza výsledků, formulace závěrů.

#### **Seznam doporučené literatury:**

BAZARAA, M. S., H. D. SHERALI and C. M. SHETTY. Nonlinear programming: theory and algorithms. 2nd ed. New York: John Wiley & Sons, 1993. ISBN 04-715-9973-5.

BIRGE, J. R. and F. LOUVEAUX. Introduction to Stochastic Programming. Springer Verlag, 1997. ISBN 978-1-4614-0236-7.

GAMS Modelling Language Manuals. GAMS Company, 2015.

NASH, S. and A. SOFER. Linear and nonlinear programming. McGraw-Hill, 1995. ISBN 978-0070460652.

WILLIAMS, H. P. Model building in mathematical programming. 5th ed. Hoboken, N.J.: John Wiley & Sons, 2013. ISBN 978-1-118-44333-0.

PARDALOS, P. M. and M. G. C. RESENDE (eds.). Handbook of applied optimization. Oxford: Oxford University Press, 2002. ISBN 01-951-2594-0.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2017/18

V Brně, dne

L. S.

---

prof. RNDr. Josef Šlapal, CSc.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty



## **Abstrakt**

Diplomová práce se zabývá optimalizačními modely s návrhem nové infrastruktury odpadového hospodářství v ČR tak, aby se spalitelný odpad, který není využíván materiálově, mohl využít energeticky. Tento úkol je zpracováván pomocí optimalizačních modelů, zahrnujících dopravní a směšovací úlohy.

Nejprve jsou v této práci uvedeny pojmy z teorie grafů a optimalizace. Následně jsou vysvětleny některé funkce programu GAMS a později i programovacího jazyka VBA užívaného pro rychlé zacházení s rozsáhlejšími daty. V hlavní části jsou vyvinuty tři postupně se rozšiřující modely, do kterých jsou na závěr implementována data z informačního systému odpadového hospodářství.

## **Summary**

This diploma thesis deals with optimization models with design of a new waste management infrastructure in the Czech Republic, such that combustible waste, which is not utilized by the material recovering, can be used by energy recovering. This task is handled by optimization models, including traffic and mixing problems.

First of all, the concepts of graph theory and optimization are presented in this paper. Subsequently, some of the GAMS functions are discussed, and later the VBA programming language used to handle the larger data quickly is presented. In the main part, three gradually expanding models are developed. At the end the data from the waste management information system are implemented into them.

## **Klíčová slova**

Energetické využití odpadu, bipartitní graf, směšovací úloha, smíšená celočíselná lineární optimalizace, GAMS

## **Keywords**

Energy recovery of waste, bipartite graph, mixing problem, mixed integer linear programming, GAMS



Prohlašuji, že jsem svoji diplomovou práci *Modely matematického programování pro směšovací úlohy* vypracoval samostatně pod vedením RNDr. Pavla Popely, Ph.D. s použitím materiálů uvedených v seznamu literatury.

Vít Kalenský



Děkuji svému vedoucímu diplomové práce RNDr. Pavlu Popelovi, Ph.D. za obětavé vedení a potřebný neustávající vyvíjený tlak. Také bych rád poděkoval za ochotu a věcné rady Ing. Radovanu Šomplákovi Ph.D., Ing. Vlastimíru Nevrlému a Ing. Františku Janošťákovi. V neposlední řadě také své přítelkyni Bc. Petře Kosové, rodině a kamarádům za podporu a trpělivost.

Vít Kalenský



# Obsah

Úvod	1
<b>1 Základní matematické pojmy</b>	<b>2</b>
1.1 Teorie grafů	2
1.1.1 Neorientovaný a orientovaný graf	2
1.1.2 Incidenční matice	3
1.1.3 Další pojmy	3
1.1.4 Typy grafů	4
1.1.5 Bipartitní graf	4
1.1.6 Úlohy o toku	5
1.1.7 Dopravní úlohy	6
1.2 Základní optimalizační úlohy	6
1.2.1 Lineární optimalizace	6
1.2.2 Smíšená celočíselná lineární optimalizace	6
1.2.3 Stochastická optimalizace	7
1.2.4 Wait-and-See přístup	7
1.2.5 Here-and-Now přístup	7
<b>2 Matematické modely směšování komodit</b>	<b>8</b>
2.1 GAMS - General Algebraic Modeling System	9
2.1.1 CPLEX řešitel	9
2.1.2 Proměnná SOS2	10
2.1.3 Import dat z Excelu (GDX soubor)	10
2.1.4 Export dat do Excelu	10
2.2 Základní úloha	12
2.2.1 Řešení základní úlohy	14
2.2.2 Zdrojový kód v GAMSu	14
2.3 Úloha celočíselného lineárního programování	16
2.3.1 Dopravní úloha s využitím bipartitní sítě	16
2.3.2 Stavba potencionálních zařízení	16
2.3.3 Řešení problému nelinearity cenových funkcí	17
2.3.4 Řešení celočíselné lineární úlohy	20
2.3.5 Zdrojový kód v GAMSu	20
2.4 Úloha MILP rozšířená o penalizační funkci	23
2.4.1 Řešení výpočtu penalizace nedostatečné výhřevnosti	23
2.4.2 Řešení úlohy s rozšířením o penalizaci	25
2.4.3 Zdrojový kód v GAMSu	25
<b>3 Implementace dat z ISOH do modelu</b>	<b>29</b>
3.1 Případová studie	29
3.1.1 Nakládání se SO v ČR v letech 2009 až 2015	29
3.1.2 Bilance mezi produkcí a zpracováním odpadu v ORP	30
3.1.3 Potenciál SO pro EVO	31
3.2 Strukturalizace dat	32

3.3	Implementovaná data . . . . .	36
3.3.1	Producenti a zpracovatelé . . . . .	36
3.3.2	Náklady za zpracování a dopravu . . . . .	38
3.4	Řešení úlohy s implementovanými daty . . . . .	40
3.4.1	Celkové náklady . . . . .	40
3.4.2	Zpracování odpadu . . . . .	41
3.4.3	Penalizace nedostatečné výhřevnosti zařízení . . . . .	43
3.4.4	Zpracování odpadu v daném ORP . . . . .	44
	<b>Závěr</b>	<b>45</b>
	<b>Seznam zkratk</b>	<b>47</b>
	<b>Seznam příloh</b>	<b>48</b>



# Úvod

Motivací k tvorbě této diplomové práce byla otázka energetického využívání spalitelných odpadů, které nejsou využívány materiálově, ani nejsou exportovány do zahraničí.

Od doby vstupu ČR do EU jsou po naší zemi vyžadovány implementace nových trendů v oblasti odpadového hospodářství. V návaznosti na to Ministerstvo životního prostředí v roce 2016 předložilo návrh změn Zákona o odpadech. Z tohoto důvodu již ČR implementovala do legislativy zákaz skládkování využitelných odpadů od roku 2024. Při plnění tohoto požadavku je snahou maximalizovat materiálové využití vzniklých odpadů. V případě, že odpad není možné využít materiálově, musí dojít alespoň k energetickému využití.

Tato diplomová práce se bude snažit pomocí znalostí principů dopravních a směšovacíh úloh navrhnout adekvátní přizpůsobení zpracovatelské infrastruktury tak, aby bylo možné využít veškerý spalitelný odpad. Při tom bude čerpat informace z Informačního systému odpadového hospodářství (ISOH) a využívat také zpracované informace pomocí nástroje pro predikci toků mezi jednotlivými územními celky, který lze nalézt v článku [1].

Nejprve budou v práci vysvětleny některé základní pojmy potřebného matematického aparátu z oblasti teorie grafů a optimalizace. V kapitole Teorie grafů budou uvedeny některé základní typy grafů. Tyto grafy budou v této práci vyjadřovat dopravní sítě. Mezi vstupy pro úlohu budou i parametry, které nebudou přesně dané, neboli bude třeba zacházet s náhodnou veličinou. Proto bude zaveden pojem stochastické optimalizace v sekci 1.2. Pro úspěšný výpočet optimálního řešení pro naši úlohu bude třeba ukázat i některé užitečné části programu GAMS, který bude diskutován v kapitole 2.1. V tomto programu bude výpočet probíhat. Zde práce také uvede, jakým způsobem jsou importována data z Excelu do GAMSu a exportována pro výpis zpět.

V hlavní části se práce pokusí vytvořit matematickou úlohu odpovídající našemu problému a aplikovat ji na vstupní data v řádu desítek uzlů. Nalezne odpovídající účelovou funkci a omezení taková, aby co nejvíce odpovídala reálné situaci z odpadového hospodářství. Určí výhody a nevýhody úloh postavených na obecné síti a na síti bipartitní. Vyřeší problém výběru vhodného způsobu optimalizace a k němu vázaného řešitele. Kromě toho také navrhne závislosti ceny za zpracování pro zařízení potencionálně určené ke stavbě. Dále navrhne také penalizační funkci za nedostatečnou výhřevnost zpracovávané směsi, která bude určena pomocí odhadu ztráty výdělků zařízení za nevyprodukovanou energii v podobě tepla a elektřiny.

Na závěr tato práce v kapitole Implementace dat z ISOH do modelu uvede případovou studii pro rok 2015, kde naznačí rozsah řešeného problému. Ukáže některé metody pro rychlou modifikaci obsáhlých dat v programu Microsoft Excel a nakonec po implementaci poskytnutých dat z ISOH do vyvinutého modelu shrne výsledky poskytnuté programem GAMS.

# 1. Základní matematické pojmy

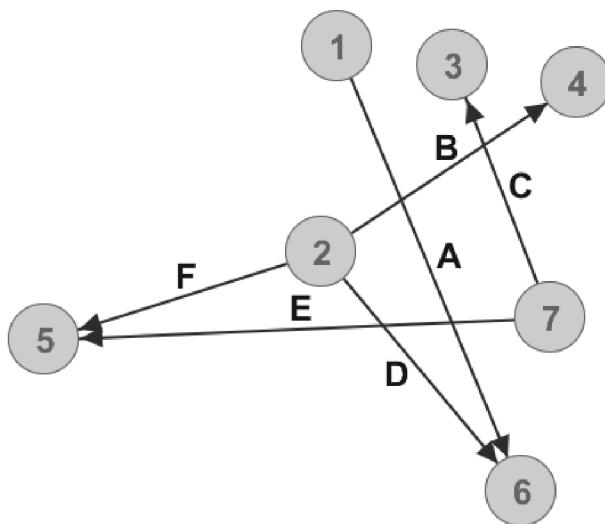
## 1.1. Teorie grafů

K pochopení následujících matematických modelů rozvozu komodit a směšovacích úloh je třeba mít základní znalosti z teorie grafů, viz [2]. Graf se skládá z uzlů a hran. Hrana spojuje dva uzly a je buď orientovaná nebo neorientovaná. Orientovaná hrana se liší od neorientované tím, že má počáteční a koncový uzel. Smyčkou nazýváme hranu, která spojuje jeden bod se sebou samým. Orientovaný graf má všechny hrany orientované a neorientovaný graf má všechny hrany neorientované.

### 1.1.1. Neorientovaný a orientovaný graf

Trojici  $G = (I^G, E^G, A^G)$ , tvořenou neprázdnou konečnou množinou  $I^G$ , jejíž prvky nazýváme *uzly*, konečnou množinou  $E^G$ , jejíž prvky nazýváme *neorientovanými hranami* a zobrazením  $A^G : E^G \rightarrow I^{G^2}$ , které nazýváme *vztahem incidence*, nazýváme *neorientovaný graf*. Na neorientovaném grafu je postavena základní úloha.

*Orientovaný graf* je trojice  $G = (I^G, E^G, A^G)$  stejně jako v případě neorientovaného grafu s tím rozdílem, že množinu  $E^G$  nazýváme *orientovanými hranami* a vztah incidence  $A^G : E^G \rightarrow I^{G^2}$  přiřazuje každé hraně  $e \in E^G$  uspořádanou dvojici uzlů  $[i, j]$ . První uzel nazýváme *počáteční uzel* hrany a značíme jej  $P_e^u$ . Druhý z nich nazýváme *koncový bod* hrany a značíme jej  $K_e^u$ . Příklad orientovaného grafu lze vidět na obrázku 1.1. Na orientovaném grafu jsou postaveny ostatní rozšířené úlohy. Vztah uzlů k hranám a k ostatním uzlům udává incidenční matice, kterou se zabývá následující text. Pro graf i incidenční matici platí,  $1, 2, 3, 4, 5, 6, 7 \in I^G$  a zároveň  $A, B, C, D, E, F \in E^G$ .



Obrázek 1.1: Příklad orientovaného grafu

### 1.1.2. Incidenční matice

Inc. M.	A	B	C	D	E	F
<b>1</b>	-1	0	0	0	0	0
<b>2</b>	0	-1	0	-1	0	-1
<b>3</b>	0	0	1	0	0	0
<b>4</b>	0	1	0	0	0	0
<b>5</b>	0	0	0	0	1	1
<b>6</b>	1	0	0	1	0	0
<b>7</b>	0	0	-1	0	-1	0

Tabulka 1.1: Incidenční matice grafu na obrázku 1.1

Všechny příslušnosti uzlů k jiným uzlům lze snadno zaznamenat pomocí incidenční matice. Pro graf na obrázku 1.1 by taková matice vypadala jako matice v tabulce 1.1. Pro příslušný uzel ve stejném řádku a příslušnou hranu ve stejném sloupci znamená číslo 0, že uzel není ani počátečním ani koncovým uzlem. Číslo  $-1$  znamená, že daná hrana má v daném uzlu počátek a číslo 1, že daná hrana má v daném uzlu konec. V této práci to ale bude vyjadřovat spíše, ze kterého uzlu na hranu komodita přichází a kudy odchází. Skýtá to jednu velkou výhodu pro matematický model.

Řekněme, že  $x_e$  je proměnná udávající množství komodity převážené po jednosměrných (orientovaných) hranách  $e \in E^G$ . Dále řekněme, že  $A_{i,e}^G$  jsou prvky incidenční matice dané sítě pro všechny uzly  $i \in I^G$  a hrany  $e \in E^G$ . Potom suma

$$\sum_e A_{i,e}^G x_e \quad \forall i \in I^G \quad (1.1)$$

vyjadřuje přírůstek, nebo úbytek množství komodit v každém uzlu  $i \in I^G$ . Tento způsob je využívám v rozšířených modelech v kapitole matematické modely směšování komodit.

### 1.1.3. Další pojmy

Množina hran incidentních s uzlem  $i$  se nazývá *okolí uzlu  $i$*  a značí se  $E_i^G$ .

*Stupeň uzlu  $i$*  je počet hran incidentních s uzlem  $i$  v grafu  $G$ , přičemž smyčky počítáme dvakrát. Značíme ho  $d_i^G$  a je roven součtu hran vstupujících do uzlu  $i$  a vystupujících z uzlu  $i$ , tedy

$$d_i^G = d_i^{G+} + d_i^{G-},$$

kde  $d_i^{G+}$  je počet hran vstupujících do uzlu  $i$  a  $d_i^{G-}$  je počet hran vystupujících z uzlu  $i$ . *Násobnost hrany* je počet hran spojujících uzly  $i$  a  $j$ . Značíme ji  $m_{i,j}^G$  a platí  $m_{i,j}^G = |E_i^G \cap E_j^G|$ .

*Řez určený množinou uzlů  $J \subset I^G$*  v grafu  $G$  je množina všech hran, jejíž jeden uzel leží v množině  $J$  a druhý v ní neleží. Značíme jej  $W_J^G$ .

Posloupnost uzlů a hran  $i_0, e_1, i_1, e_2, i_2, \dots, e_k, i_k$  nazýváme *orientovaným sledem*, jestliže pro každou hranu  $e$  z této posloupnosti platí  $P_{e_l}^u = i_{l-1}$  a  $K_{e_l}^u = i_l$ , tedy koncový uzel hrany je zároveň počátečním uzlem hrany následující.

*Orientovaný (neorientovaný) tah* je orientovaný (neorientovaný) sled, v němž se žádná hrana neopakuje.

*Orientovaná (neorientovaná) cesta* je orientovaný (neorientovaný) sled, v němž se neopakuje žádný uzel. V rozšířených úlohách, viz kapitola 2.3, jsou vypočítány parametry

některých orientovaných cest a pro zjednodušení modelu tyto cesty dosazeny jako jednotlivé hrany.

#### 1.1.4. Typy grafů

Často se k hranám či uzlům připisují hodnoty, které reprezentují dobu trvání, propustnosti nebo náklady. Graf, jehož hrany nebo uzly jsou opatřeny těmito hodnotami, nazýváme *ohodnoceným grafem* či *ohodnocenou sítí*. V našem případě jsou do těchto parametrů dosazovány délky hran.

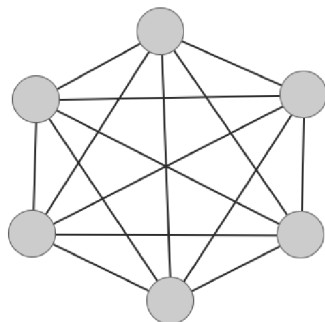
Graf  $G$ , v němž pro  $\forall [i, j] \in I^{G^2}$  platí, že  $m_{i,j}^G \leq 1$ , neboli násobnost každé existující hrany, je rovna jedné, se nazývá *prostý graf*.

Graf  $G$ , kde  $\exists [i, j] : m_{x,y}^G \geq 2$ , neboli existuje alespoň jedna hrana s násobností větší než jedna, se nazývá *multigraf*. Větší násobnost hrany je užita v případě převozu dvou různých komodit po jedné hraně, každé je totiž nutné přiřadit jinou výhřevnost a proto nelze množství komodit jen sečíst viz sekce 2.3.

Graf, který vznikl z jiného grafu odebráním nezáporného počtu hran a uzlů se nazývá *podgraf* daného grafu. Ke každé hraně musí podgraf obsahovat i její počáteční a koncový bod. Každý graf pokládáme za podgraf sama sebe.

*Úplný orientovaný graf* je prostý graf  $G = (I^G, R)$ , kde  $R = I^{G^2}$ , tedy množina všech uspořádaných dvojic uzlů  $i \in I^G$ .

*Úplný graf* je prostý neorientovaný graf bez smyček, jehož každá dvojice různých uzlů je spojena hranou. Má-li  $n$  uzlů, značíme jej  $K_n$ . Viz obr. 1.2.



Obrázek 1.2: Úplný graf  $K_6$

Úplný graf je využíván v základním modelu. Tento model využívá úplnou matici vzdáleností z každého uzlu do každého. Smyčky jsou zde zvažovány jako hrany s nulovou délkou.

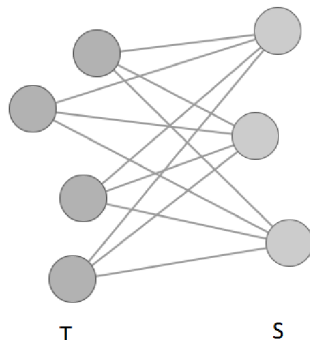
#### 1.1.5. Bipartitní graf

Graf  $G$ , jehož množina uzlů  $I^G$  je sjednocením dvou množin  $S, T$  (stran bipartitního grafu) se nazývá *bipartitní graf* a platí

$$E^G = W_S^G.$$

Jinými slovy každá hrana bipartitního grafu  $G$  má jeden uzel v  $S$  a druhý v  $T$ , tyto množiny nazýváme *stranami* bipartitního grafu. U orientovaného bipartitního grafu zpravidla požadujeme, aby všechny hrany byly orientovány souhlasně. *Úplný bipartitní graf* je

takový, ve kterém každá dvojice uzlů  $s \in S$ ,  $t \in T$  je spojena právě jednou hranou. Úplný bipartitní neorientovaný graf, jehož strany mají  $m = |S|$  a  $n = |T|$  prvků, označujeme jako  $K_{m,n}$ . Viz obr. 1.3.



Obrázek 1.3: Úplný graf  $K_{4,3}$

Bipartitní graf je využíván v rozšířených modelech, kde jednu stranu grafu tvoří producenti a druhou zpracovatelé.

### 1.1.6. Úlohy o toku

Optimalizační úlohy, které se zabývají přesunem prvků z uzlů do jiných uzlů pomocí hran, jsou jedny z nejčastějších aplikací teorie grafů.

Nechť  $G$  je orientovaný graf. Ohodnocení hran reálnými čísly  $x : E^G \rightarrow \mathbb{R}$  nazýváme *tokem v síti  $G$* , když pro každý uzel  $i$  splňuje *Kirchhoffův zákon*

$$\sum_{e \in E^+(i)} x_e = \sum_{e \in E^-(i)} x_e,$$

kde  $E_i^+$  jsou všechny hrany do uzlu  $i$  vstupující a  $E_i^-$  všechny hrany vystupující. Neboli množství toku, které do uzlu vteče, z uzlu musí zase odtéct. V tokové úloze platí Kirchhoffův zákon pro všechny uzly kromě producenta a zpracovatele, což můžeme brát jako počáteční a koncový bod toku. Zatímco v tzv. *cirkulaci* podléhají tomuto zákonu všechny uzly. Velikost toku  $x_e$  na hraně  $e$  můžeme omezit zdola hodnotou  $l_e$  a shora hodnotou  $c_e$ , takže platí  $l_e \leq x_e \leq c_e$ . Tento interval pak nazýváme *přípustným tokem*. Je-li dolní hranice toku ve všech hranách nulová a je-li dán producent a zpracovatel, pak nazýváme tuto úlohu *transportní sítí*. Označme zdrojový uzel jako  $i^p$ . Pro *velikost toku od producenta ke zpracovateli* platí

$$X_e = \sum_{e \in E_{i^p}^+} x_e - \sum_{e \in E_{i^p}^-} x_e,$$

tedy je roven vyprodukovanému toku v uzlu producenta.

#### Nejlevnější tok

Předpokládejme, že každá hrana je kromě omezení shora a zdola ohodnocena ještě *jednotkovou cenou toku*  $P_e^{trans}$ . *Cenou toku* pak nazýváme součet součinů  $x_e P_e^{trans}$ . Celková cena toku v síti je pak

$$\sum_{e \in E^G} x_e P_e^{trans}.$$

Potom je úkolem najít nejlevnější variantu toku v hranách tak, aby splňovala daná omezení. V případě naší dopravní úlohy je třeba tento vztah vynásobit ještě délkou hrany. Pak vztah vypadá takto:

$$\sum_{e \in E^G} D_e x_e P_e^{trans}.$$

### 1.1.7. Dopravní úlohy

Jedno z hlavních využití tokových úloh v reálném světě, dopravní úlohy, jsou orientované sítě s více producenty a více zpracovateli komodit. Hraný grafu si lze představit jako silnice, železnice nebo letecké a lodní dopravní linky. Za předpokladu, že je zadána jednotková cena za projetí materiálu hranou a je omezené maximální množství, které může za časový okamžik pojmout každá jednotlivá hrana, hledáme nejlevnější tok sítě.

## 1.2. Základní optimalizační úlohy

### 1.2.1. Lineární optimalizace

Úlohu lineárního programování formuloval v roce 1947 Dantzig. Lineární programování má určitou filosofii na tvoření modelů, které mají aplikaci v široké škále rozhodujících procesů vyskytujících se v odvětví státní správy, průmyslu, ekonomiky a inženýrství. Aby byl model lineární, musí splňovat předpoklad proporcionality, nezápornosti a aditivity. Úlohou lineárního programování rozumíme úlohu, jejíž účelová funkce i omezení jsou lineární, tedy

$$? \in \operatorname{argopt}_x \{ \mathbf{c}^T \mathbf{x} \mid \mathbf{A}\mathbf{x} \circ \mathbf{b} \},$$

kde  $\circ \in \{ \leq, =, \geq \}^n$  a  $\operatorname{argopt}$  vyjadřuje buď  $\operatorname{argmin}$ , nebo  $\operatorname{argmax}$ . Množinu  $C = \{ \mathbf{x} \mid \mathbf{A}\mathbf{x} \circ \mathbf{b} \}$  nazýváme *množinou přístupných řešení* a její prvky *přípustná řešení*. Následující výraz popisuje *omezení* úlohy

$$\mathbf{A}\mathbf{x} \circ \mathbf{b}$$

a rovnice

$$z = \mathbf{c}^T \mathbf{x}$$

popisuje *účelovou funkci* úlohy. Existují dva případy, pro které je úloha neřešitelná. Za prvé pokud je množina  $C$  prázdná a nebo když  $? \in \operatorname{argopt}_x \{ \mathbf{c}^T \mathbf{x} \mid \mathbf{A}\mathbf{x} \circ \mathbf{b} \} = \emptyset$ . V tomto případě nemá úloha konečné řešení a je tedy rovněž neřešitelná. Úloha má řešení pouze, pokud  $? \in \operatorname{argopt}_x \{ \mathbf{c}^T \mathbf{x} \mid \mathbf{A}\mathbf{x} \circ \mathbf{b} \} \neq \emptyset$ . Nazvěme tuto množinu  $M$ , pak prvky  $M$  jsou optimální řešení úlohy lineárního programování.

Při vývoji lineárního programování byla zavedena takzvaná *simplexová metoda*. Více o lineárním programování lze nalést v knize [3] nebo v textu [4].

### 1.2.2. Smíšená celočíselná lineární optimalizace

Pokud se v úloze matematického programování vyskytne požadavek na celočíselnost některých nebo všech rozhodovacích proměnných, nazvěme tuto úlohu *úlohou celočíselného programování*. Této úlohy může být využito pro řešení více typů reálných problémů, kdy jsou proměnnými nedělitelné prvky, jako jsou třeba osoby. Takové úlohy se nazývají *úlohy*

s nedělitelnostmi. V našem případě ovšem nebude motivací nedělitelnost, ale spíše vyskytující se nelinearita, která bude linearizována pomocí speciálně uspořádaných proměnných SOS2 popsaných v sekci GAMS, které se řeší pomocí *smíšeného celočíselného lineárního programování* (MILP - mixed integer linear programming). Pro tento případ platí, že všechna neceločíselná omezení a i účelová funkce musí být lineární. Pomocí formulace MINLP (mixed integer nonlinear programming), tedy smíšeného celočíselného nelineárního programování, bude v následujícím textu zavedeno i MILP.

### Formulace MINLP

Obecná formulace MINLP úlohy vypadá takto:

$$\min f(x_1, x_2, \dots, x_n),$$

za podmínky

$$\begin{aligned} g_l(x_1, x_2, \dots, x_n) &\leq 0, & l = 1, 2, \dots, m, \\ x_j \in M_j &\subseteq \mathbb{Z} & j \in J, \end{aligned}$$

kde  $J \neq \emptyset, J \subseteq 1, 2, \dots, n$  a  $\mathbb{Z}$  je množina celých čísel. Pokud jsou všechny funkce  $f, g_1, \dots, g_m$  lineární, pak hovoříme o MILP. Podmínka celočíselnosti je vázána pouze na některé proměnné (tj,  $J \subset \{1, 2, \dots, n\}$ ).

### 1.2.3. Stochastická optimalizace

Práce předpokládá předchozí znalosti základů deterministické optimalizace, které lze získat například z knihy [5] a dále se bude zabývat rozšířením, které bude využívat pro modelování reálné situace.

Základní stochastický model lze získat z deterministického tím způsobem, že se z některých parametrů stanou náhodné proměnné, značíme je např.  $\xi$  s realizacemi  $\xi^s$ , kde  $s$  vyjadřuje daný scénář, pro který  $\xi$  nabývá určité hodnoty.

### 1.2.4. Wait-and-See přístup

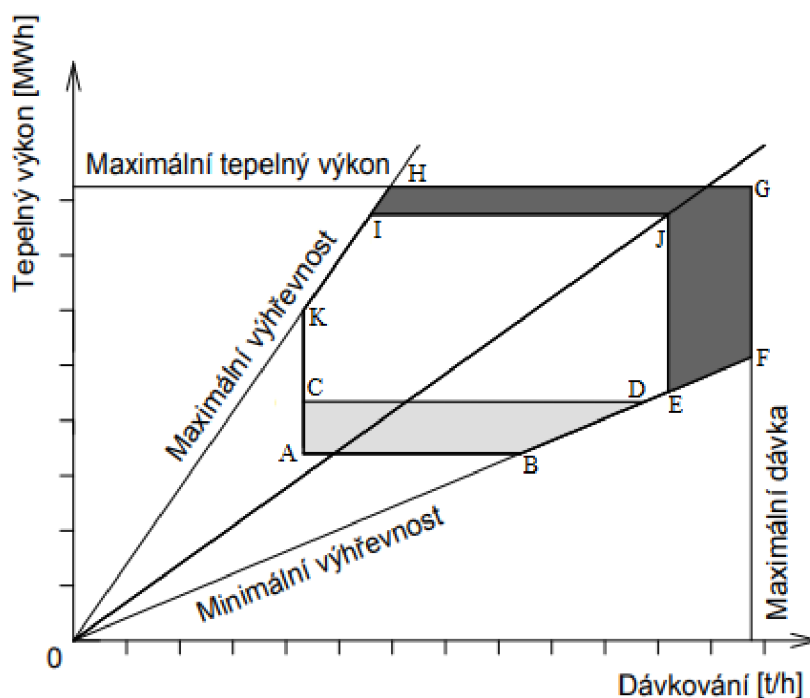
Model, který rozhoduje o proměnných až po zjištění skutečné hodnoty náhodného parametru  $\xi^s$ , se nazývá Wait-and-See. Jinak řečeno tento přístup je cenný za předpokladu, že známe realizaci  $\xi^s$  ještě předtím, než musíme rozhodovat o proměnných  $x$ , neboli je k němu perfektní informace. V tomto případě je možné stanovit funkci  $x(\xi)$  závislou na náhodném vektoru parametrů  $\xi$ . Tento přístup je využíván v dlouhodobém plánování. V našem případě chceme model připravit na to, že danou realizaci  $\xi^s$  znát nebude, proto bude nutné užívat přístupu *Here-and-Now* popsaného v následujícím odstavci.

### 1.2.5. Here-and-Now přístup

U stochastického programování je častější, že se musí zvolit proměnná  $x$  dříve, než je známá realizace náhodného vektoru parametrů  $\xi^s$ . Tato metoda se nazývá Here-and-Now přístup. Vektor  $x$  tedy musí být stejný pro všechny budoucí realizace hodnot náhodných proměnných  $\xi^s$ . Tato metoda bude aplikována v této práci. Pro produkci odpadů v daných ORP budou použita data z roku 2015 a pro výhřevnosti odpadů vygenerovány hodnoty pomocí náhodných veličin s normálním rozdělením se známou střední hodnotou a rozptylem. Viz kapitola 2.

## 2. Matematické modely směšování komodit

V této kapitole se již dostáváme k samotné praktické části diplomové práce, tedy k vývoji optimalizačních modelů pro tokovou síť spojenou se směšovací úlohou více komodit. To znamená, že před námi nestojí jen úkol co nejlevnější přepravy a zpracování, nýbrž i takzvaná úloha pro energetické využití různých komodit (paliv). Než se dostaneme k modelům, je třeba nastínit tuto úlohu pomocí obrázku 2.1.



Obrázek 2.1: Obecný spalovací rošt a jeho provozní nároky. Zdroj: [6],[7]

Každé palivo, které spalovna s tímto spalovacím roštem přijme, má svou výhřevnost znázorňovanou konkrétní šikmou osou. Spalovna může přijímat i paliva s vyšší výhřevností než je maximální a nižší než je minimální pouze v případě, že výsledná směs těchto výhřevností bude mezi těmito hranicemi. Ideální směs pro tuto spalovnu má výhřevnost, jejíž osa prochází bodem J. Bod J můžeme považovat za referenční bod, kterému se chceme blížit při provozu spalovny, protože vyprodukuje za relativně krátký čas velké množství energie a dosáhne tak nejlepšího využití paliv. Ve zmíněné úloze o energetickém využití různých komodit se snažíme maximalizovat rozdíl mezi cenou vyprodukované energie a cenou dopravy daných paliv do spaloven spolu s cenou provozu spaloven. Oblast C-D-E-J-I-K je optimální provozní stav, při kterém je rošt schopen dlouhodobě pracovat. Oblast A-B-C-D ukazuje stavy, při kterých je třeba dodávat zemní plyn, aby se spalovací rošt udržel v provozním stavu a oblast E-F-G-H-I-J ukazuje stavy, při kterých je rošt přeplněn a nebo dosahuje přílišných teplot. Ani jedna ze šedých oblastí není dlouhodobě udržitelná.



## 2.1. GAMS - General Algebraic Modeling System

Matematické modely budou v této práci počítány v programu GAMS. Je tedy nutné osvětlit některé jeho funkcionality před přístupem k modelům.

Program GAMS je modelovací systém na vysoké úrovni pro matematické programování a optimalizaci. Obsahuje jazykové překladače a vysoce výkonné řešiče. GAMS je speciálně navržen pro modelování lineárních, nelineárních a smíšených celočíselných optimalizačních problémů. Další informace lze nalézt na [8].

Tato kapitola je věnována některým doplňkům programu GAMS, které budou využity při implementaci modelů do programu.

### 2.1.1. CPLEX řešitel

Program GAMS pro výpočet používá různé programy, takzvané *řešitele*, které udávají postup výpočtu optimálních řešení. Je tedy jasné, že časová náročnost výpočtu je velmi závislá na tom, který řešitel je pro danou úlohu použit. Více o řešitelích lze najít v [9].

CPLEX je jedna z mnoha možností použitelných pro smíšené celočíselné programování (MIP), tedy Mixed Integer Programming. Optimalizátory CPLEX jsou navrženy tak, aby rychle a s minimálními zásahy uživatele řešily velké a obtížné problémy. Většinu lineárních úloh řeší pomocí duálního simplexového algoritmu.

Řešení problémů lineárního programování (LP) je ale velmi náročné na paměť. I když CPLEX zvládá využívat paměť velmi efektivně, nedostatečná fyzická paměť je jedním z nejběžnějších problémů při spouštění velkých LP. Když je paměť omezená, CPLEX automaticky provede úpravy, které ale mohou negativně ovlivnit výkon.

Úloha, kterou se zabývá tato práce, spadá do většiny problémů, které řešitel CPLEX dokáže spočítat pomocí výchozího klasického nastavení, avšak toto nastavení není jediné možné, které se dá užívat. Některé úlohy je například lepší počítat pomocí primárního simplexového algoritmu a nebo algoritmu používající bariéry. V extrémních případech, kdy je počet parametrů výrazně větší, CPLEX využívá takzvaného "sifting" algoritmu, neboli prosévacího algoritmu. Kromě jiného řešitel CPLEX také podporuje proměnné typu SOS2 (special ordered system of type 2), neboli speciálně uspořádané množiny typu 2.

**CPLEX pro MIP** Metody používané k řešení problémů čistého celočíselného a smíšeného celočíselného programování vyžadují dramaticky více matematického výpočtu, než metody pro podobné neceločíselné programy.

Řešení mnoha poměrně malých celočíselných úloh zabere obrovské množství času. Pro řešení úloh s celočíselnými proměnnými používá CPLEX takzvaný "Branch and Cut" algoritmus, neboli algoritmus větvení a řezání. Protože jediný smíšený celočíselný problém generuje mnoho podproblémů, dokonce i malé smíšené celočíselné problémy mohou být velmi náročné na výpočty a vyžadují značné množství fyzické paměti.

CPLEX může také řešit problémy typu MIQCP, tedy "Mixed Integer Quadratic Constraints Problem", neboli smíšené celočíselné problémy s kvadratickými omezeními, ale díky tomu, že podporuje SOS2 proměnné, můžeme si tato kvadratická omezení převést v řadě případů efektivně na po částech lineární.

### 2.1.2. Proměnná SOS2

Tato proměnná je diskrétní množina prvků, z nichž maximálně dva mohou nabývat jakékoliv nezáporné hodnoty a ostatní musí být rovny nule. Tato proměnná je zavedena v programu GAMS. Tyto dva prvky se sebou musí navzájem sousedit. Zpravidla je tato množina používána v závislosti na nějakém parametru. Abychom mohli používat proměnné typu SOS2, musíme používat řešič pro smíšené celočíselné modely, například CPLEX. Více o těchto proměnných na webových stránkách [10].

### 2.1.3. Import dat z Excelu (GDX soubor)

Pro práci s obsáhlými daty je Microsoft Excel velmi užitečný program a proto je v této práci užíván pro čtení vstupních dat a následně i pro zapisování výstupních dat.

GAMS řeší načítání z Excelu pomocí vytvoření GDX souboru, kam načte veškeré zadané parametry a zde s nimi potom pracuje. Soubor GDX, neboli GAMS Data EXchange se dá používat na načítání množin, parametrů, ale i rovnic.

Pro ilustraci použití tohoto nástroje lze nahlédnout do následujícího textu, který je zkopírovaný z funkčního programu jednoho z modelů. Symbol \$ označuje začátek kódu,

```
$call GDXXRW.exe Implementace.xlsm set=i rng=i rdim=1 cdim=0
$GDXIN    Implementace.gdx
$LOAD     i pi zi M C p e D A
$GDXIN
```

který není součástí GAMS jazyka popisujícího model, proto musí být zobrazen na začátku řádku. V řádku označeném \$ se nerozlišují velká a malá písmena. Slovo "call GDXXRW.exe" volá soubor GDXXRW.exe. Tento program je jeden z nástrojů GDX souboru, který slouží k načítání dat ze souboru například typu .xls (.xlsx, nebo .xlsm). Následuje název souboru, ze kterého se program GDXXRW.exe pokusí přečíst požadovaná data, v našem případě Implementace.xlsm.

Identifikace množiny nebo parametru, který chceme zapisovat, se dá provádět několika různými postupy. V tomto případě je to řešeno tak, že v daném Excel sešitu je nutno v záložce Vzorce pod ikonou Správa názvů nastavit proměnné odkazující na určitou oblast některého listu, například *i* pro sloupec s vypsányými pořadovými čísly uzlů, nebo například *A* pro tabulku incidenční matice. V kódu potom stačí jako set (množinu), nebo jako par (parametr) dát název dané množiny/parametru, v našem případě *i*, a následně rng (range) nastavit jako název proměnné, která ukazuje na vybraná data, v našem případě opět *i*. Počet rozměrů dané proměnné je dán součtem rdim a cdim, neboli raw dimension a column dimension. Příkaz "\$GDXIN" spolu s následným názvem souboru otevře soubor s tímto názvem, v našem případě Implementace.gdx.

Příkaz "\$LOAD" načte uložené množiny a parametry do souboru.

Příkaz "\$GDXIN" bez přiloženého jména zavře otevřený soubor typu gdx.

### 2.1.4. Export dat do Excelu

Aby se dalo s výstupy rychle pracovat, je užitečné si je nechat okamžitě po propočítání úlohy vypsát zpět do souboru Microsoft Excel. V případě vypisování proměnných, což

jsou hodnoty, které jsou v této práci vypisovány do Excelu, je situace o něco málo složitější. Program `gdxxrw.exe` je totiž dimenzován pouze na to, aby vypisoval množiny a proměnné, proto je třeba si na začátku při definování parametrů definovat zároveň i pomocné parametry, do kterých se bude po výpočtu zapisovat výsledná proměnná. V našem případě je touto proměnnou  $z$ , tedy celkový součet nákladů. Pomocným parametrem pro tuto proměnnou je  $varz$ . Definice parametru je zobrazena na prvním a druhém řádku kódu a zápis proměnné na třetím řádku. Doplnění proměnné  $z$  o příponu  $.l$  zajišťuje, že se zapíše hodnota proměnné, kterou má přímo v okamžiku, kdy probíhá kompilace na daném řádku.

Příkaz "EXECUTE UNLOAD" udává cílový soubor (`.gdx`), kam se mají data zapisovat. Tento soubor je nutné zapsat do uvozovek. Následující text udává, která data se mají zapsat.

Příkaz "EXECUTE" provádí samotnou operaci zapisování ze souboru (`.gdx`) do souboru (`.xls`). Samotný zápis provádí, stejně jako v minulé části textu, program `gdxxrw.exe`. Ten je již nastavený v předchozím kódu pomocí příkazu "call" na sešit Excel jménem "Implementace.xls". Je tedy třeba ještě nastavit, ze kterého `.gdx` souboru se budou data načítat, který parametr se bude kopírovat a kam se bude přesně v souboru (`.xls`) zapisovat. Všechny tyto informace je nutné zapsat do uvozovek ' '. Kód je zkopírován ze souboru GAMS, který pracuje již s daty z ISOH.

```
parameters
varz;

varz = z.l;

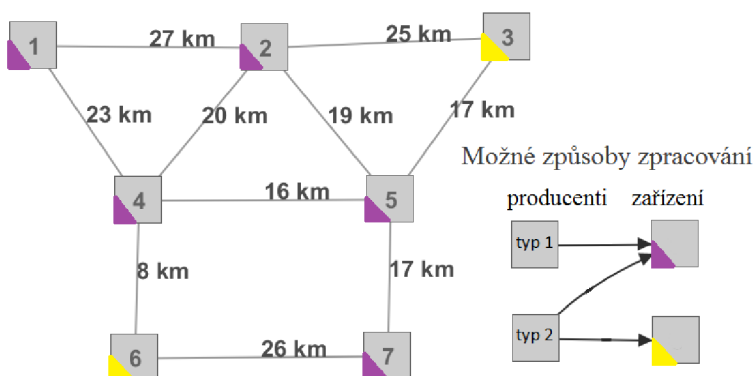
EXECUTE_UNLOAD "Implementace.gdx" varz

EXECUTE 'gdxxrw.exe Implementace.gdx par=varz rng=vysledky!A2'
```

## 2.2. Základní úloha

Úkolem základní úlohy je minimalizace nákladů na dopravu a zpracování tak, aby splňovala podmínku zpracování veškerého množství vyprodukovaných komodit a omezení spojené s výhřevností materiálu.

V základní úloze, která je zde reprezentována pomocí neorientovaného obecného grafu, se vyskytuje 7 uzlů, 10 hran neboli cest a dva druhy komodit, jak lze vidět na obrázku 2.2. Vlastnosti komodit se ovšem neliší pouze druhem materiálu, ale zároveň i tím, kde byla vyprodukována. Jestliže každý uzel produkuje komoditu, pak mísíme mezi sebou 14 různých materiálů o různé výhřevnosti. Každý uzel má danou kapacitu, která vyjadřuje maximální množství zpracovaných komodit a cenu za zpracování tuny materiálu daného typu.



Obrázek 2.2: zadaná úloha a její okrajové podmínky

Uzly označené fialovou barvou mohou zpracovávat oba typy komodit, zatímco uzly označené žlutou barvou pouze typ 2.

### Použité symboly

#### Indexové množiny

$i, j, p \in I$  indexová množina uzlů (producentů)

$o \in O$  indexová množina typů komodit (odpadů)

#### Parametry

$M_{i,o,p}$  množství vyprodukované komodity v uzlu

$C_i$  kapacita zpracovatele

$P^{trans}$  cena za km dopravy jedné tuny materiálu

$P_{i,o}^{treat}$  cena za zpracování jedné tuny materiálu

$D_{i,j}$  vzdálenost uzlů

$A_{i,o}$  binární parametr rozhodující o typech přijímaných komodit

$V_{o,p}$  výhřevnost komodity

$V^{min}$  minimální výhřevnost směsi

$V^{max}$  maximální výhřevnost směsi

### Proměnné

$z$  celkové náklady za dopravu a zpracování

$x_{i,j,o,p}$  množství komodity na dané cestě od  $i$  do  $j$

$s_{i,o,p}$  konečné zpracované množství směsi v uzlu  $i$

### Účelová funkce

$$z = \sum_{i \in I} \sum_{j \in I} \sum_{o \in O} \sum_{p \in I} D_{i,j} x_{i,j,o,p} P^{trans} + \sum_{i \in I} \sum_{o \in O} \sum_{p \in I} P_{i,o}^{treat} s_{i,o,p}$$

První část účelové funkce sčítá cenu za dopravu komodit různého typu, vyprodukovaných u různých producentů po přímých spojeních mezi všemi uzly. Tedy pro každé dva uzly  $i, j$ , daný typ odpadu  $o$  a daného producenta odpadu  $p$  existuje  $x_{i,j,o,p}$ , pomocí něhož je spočítána cena za dopravu, viz 2.2. Ve druhé části účelové funkce je přičítána cena za zpracování komodit u zpracovatelů, kde cena zpracování komodit různých typů v různých uzlech je rozdílná.

### Omezující podmínky

$$\sum_{j \in I} (x_{j,i,o,p} - x_{i,j,o,p}) + M_{i,o,p} = s_{i,o,p} \quad \forall i, p \in I, \forall o \in O, \quad (2.1)$$

$$\sum_{o \in O} \sum_{p \in I} s_{i,o,p} \leq C_i \quad \forall i \in I, \quad (2.2)$$

$$\sum_{p \in I} s_{i,o,p} \leq A_{i,o} C_i \quad \forall i \in I, \forall o \in O, \quad (2.3)$$

$$\sum_{o \in O} \sum_{p \in I} s_{i,o,p} V^{min} \leq \sum_{o \in O} \sum_{p \in I} s_{i,o,p} V_{o,p} \leq \sum_{o \in O} \sum_{p \in I} s_{i,o,p} V^{max} \quad \forall i \in I, \quad (2.4)$$

$$x_{i,j,o,p} \geq 0 \quad \forall i, j, p \in I, \forall o \in O, \quad (2.5)$$

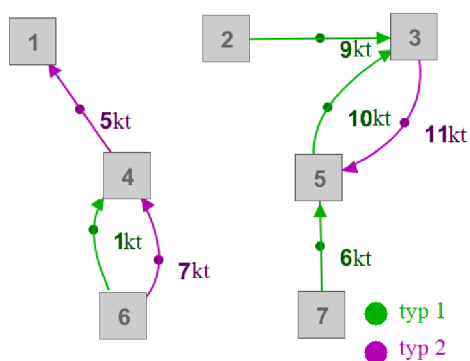
$$s_{i,o,p} \geq 0 \quad \forall i, p \in I, \forall o \in O. \quad (2.6)$$

První omezující podmínka (2.1) je výpočet konečného množství dané komodity typu  $o$  od producenta  $p$  v uzlu  $i$  dosažené do parametru  $s_{i,o,p}$ . Dále je tento parametr využit v druhém omezení (2.2) tak, že konečné množství materiálu nesmí přesáhnout danou kapacitu zařízení. To, že ne všechny typy komodit mohou do všech zařízení, zajišťuje podmínka (2.3) pomocí binárního parametru  $A_{i,o}$ , respektive pro přijímaný typ materiálu má  $A_{i,o}$  hodnotu 1 a podmínka je irelevantní. Pokud však daný typ v uzlu není přijímaný, pak  $A_{i,o} = 0$  a konečné množství komodity tohoto typu se musí rovnat 0. Podmínka (2.4) omezuje konečnou výhřevnost u zpracovatele, což vyjadřuje prostřední člen. Zdola je omezená

minimální výhřevností  $V^{min}$  a shora maximální výhřevností  $V^{max}$ . Jelikož pracujeme se součiny množství a výhřevnosti, což vyjadřuje získanou energii, nazýváme tuto podmínku energetickým omezením. Poslední dvě podmínky (2.5) a (2.6) zajišťují, aby  $x_{i,j,o,p}$  a  $s_{i,o,p}$  byly nezáporné proměnné.

### 2.2.1. Řešení základní úlohy

Na grafu 2.3 je vidět jedno z možných řešení základní úlohy. V uzlu 3 a 6 bylo vyprodukováno 11 a 7 kt materiálu. Jelikož tyto uzly neměli možnost tento typ komodity zpracovávat, museli všechno odvézt do jiných uzlů.



Obrázek 2.3: Výsledný graf

### 2.2.2. Zdrojový kód v GAMSu

Model 2.1: Základní úloha

```

1 set
2 i indexová množina uzlů (průducentů) /1,2,3,4,5,6,7/
3 o typy produkovaných komodit /o_1,o_2/
4 ;
5 alias(i,j,p)
6 ;
7 scalar
8 P_trans          fixní cena za dopravu tuny komodity jeden kilometr /10000/
9 V_min            minimální povolená výhřevnost směsi v zařízeních /10/
10 V_max           maximální povolená výhřevnost směsi v zařízeních /14/
11 ;
12 parameters
13 varz             parametr pro výpis celkových nákladů
14 varx(i,j,o,p)  parametr pro výpis dopravy komodit
15 vars(i,o,p)    parametr pro výpis zpracovaného množství komodity v zaří
                  zení
16 V(p,o)         výhřevnost komodity daného typu od daného producenta
17 M(i,p,o)       množství vyprodukované komodity určitého typu v uzlu
18 C(j)           kapacita zařízení v uzlu
19 P_treat(j,o)   cena za zpracování jedné tuny materiálu
20 D(i,j)         vzdálenost uzlů
21 A(i,o)         binární parametr rozhodující o typech přijímaných komodit
22 ;

```

```

23 $call GDXXRW.exe maly_model.xlsx par=M rng=m rdim=2 cdim=1 par=C rng=k rdim
    =1 cdim=0 par=P_treat rng=Cena rdim=1 cdim=1 par=D rng=v rdim=1 cdim=1
    par=A rng=po rdim=1 cdim=1 par=V rng=vyh rdim=1 cdim=1
24 $GDXIN    maly_model.gdx
25 $LOAD     M C P_treat D V A
26 $GDXIN

28 variable
29 z                celkové náklady
30 ;
31 positive variable
32 x(i,j,o,p)      množství dané komodity na dané cestě
33 s(i,o,p)        množství dané komodity v uzlu
34 ;
35 equations
36 ucelfce, omez1(i,o,p), omez2(i), omez3(i,o), omez4(i), omez5(i)
37 ;
38 * Účelová funkce zahrnující náklady na dopravu a na zpracování
39 ucelfce.. z =e= sum((i,j,o,p), D(i,j)*x(i,j,o,p)*P_trans) + sum((i,o,p),
    P_treat(i,o) * s(i,o,p))
40 ;
41 * Výpočet množství dané komodity v uzlu
42 omez1(i,o,p).. s(i,o,p) =e= M(i,p,o) + sum(j, x(j,i,o,p) - x(i,j,o,p))
43 ;
44 * Omezení množství zpracované směsi kapacitou zařízení
45 omez2(i).. sum((o,p), s(i,o,p)) =l= C(i)
46 ;
47 * Omezení zpracování komodit, které zařízení nepřijímá
48 omez3(i,o).. sum(p, s(i,o,p)) =l= A(i,o)*C(i)
49 ;
50 * Omezení minimální výhřevnosti zpracovávané směsi
51 omez4(i).. sum((o,p), s(i,o,p))*V_min =l= sum((o,p), s(i,o,p))*V(p,o))
52 ;
53 * Omezení maximální výhřevnosti zpracovávané směsi
54 omez5(i).. sum((o,p), s(i,o,p))*V(p,o) =l= sum((o,p), s(i,o,p))*V_max
55 ;
56 model K /all/
57 * Model K obsahuje všechny použité rovnice i nerovnice.
58 ;
59 solve K using lp minimizing z
60 * Vyřešení modelu K minimalizací z užitím lineárního programování
61 ;
62 Display M, C, P_treat, P_trans, D, V, A, x.l, z.l, s.l
63 * Zobrazení vypsanych paramertů v souboru .lst
64 ;
65 * Přepočítání proměnných do parametrů, k pozdějšímu vypsání do souboru .xls
66 varz = z.l
67 ;
68 varx(i,j,o,p) = x.l(i,j,o,p)
69 ;
70 vars(i,o,p) = s.l(i,o,p)
71 ;
72 EXECUTE_UNLOAD "vypis_maly_model.gdx" varz vars varx
73 EXECUTE 'gdxxrw.exe vypis_maly_model.gdx par=varz rng=sheet!A2'
74 EXECUTE 'gdxxrw.exe vypis_maly_model.gdx par=vars rng=sheet!B2'
75 EXECUTE 'gdxxrw.exe vypis_maly_model.gdx par=varx rng=sheet!K2'

```



## 2.3. Úloha celočíselného lineárního programování

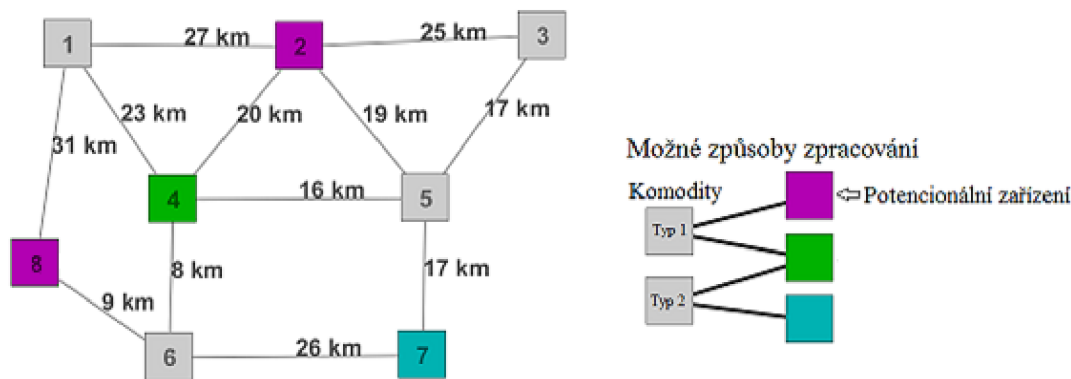
Tato úloha je celočíselná díky užití SOS2 proměnných pro linearizaci nelineární cenové funkce.

### 2.3.1. Dopravní úloha s využitím bipartitní sítě

V předchozí úloze se skrývá problém, který by se mohl projevit později při jejím použití na větší vstupní data. Jelikož je postavená na obecné a později zúplněné síti, tak v ní probíhají výpočty ceny dopravy ze všech uzlů do všech uzlů. S tímto přístupem by při zvyšování množství vstupních dat docházelo k růstu počtu rovnic a proměnných k výpočtům geometrickou řadou. Tento problém je vyřešen jednoduše postavením nové úlohy na bipartitní síť (viz podsekcce 1.1.5), kde na jedné straně sítě stojí rozdělení producenti podle typu komodity, kterou produkují a na druhé straně zpracovatelé podle typu zpracování, které nabízejí. Následně stanovíme nejkratší cesty od producentů ke zpracovatelům, kteří mohou daný typ komodity zpracovávat. Pokud bude náklad cestovat přes více uzlů než jeden, nebude již potřeba zaznamenávat veškeré množství komodity mezi uzly, ale pouze na celé cestě. Nebude již tedy třeba značit typ a producenta u každé komodity, ale vlastnosti materiálu budou dány cestou, po které se budou dopravovat ke zpracovateli.

### 2.3.2. Stavba potencionálních zařízení

Další rozšíření úlohy tkví v přidání potencionálních zařízení, o kterých model rozhodne, zda je výhodné je postavit a jakou kapacitu by měla mít. Cena investice za stavbu je zahrnuta v ceně za zpracování. Nástin situace můžete nalézt na obrázku 2.4.



Obrázek 2.4: Graf modelu celočíselné lineární úlohy. Zdroj obrázku s příjmy a výdaji: [11].

V grafu obsahujícím 12 hran a 8 uzlů se vyskytuje jedno zařízení označené modrou barvou, které představuje zpracovatele schopného zpracovávat pouze druhý typ komodit. Dále v uzlu 4 označeném zeleně jsou dvě zařízení, jedno pro zpracování prvního a druhé pro zpracování druhého typu komodit. K mísení tedy dochází pouze u stejných typů komodit. Konečně dvě fialově označená zařízení jsou schopna zpracovávat pouze první typ komodit a zároveň jsou obě potencionálními zařízeními, neboli model rozhoduje o jejich existenci. Veškeré uzly se zároveň počítají jako producenti obou typů komodit.



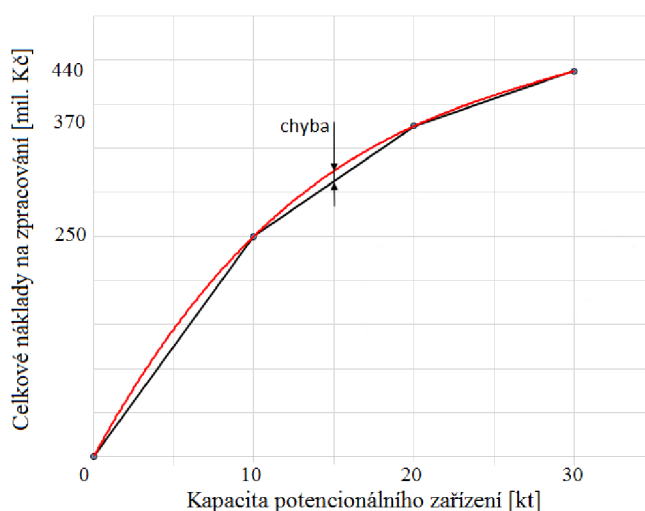
### 2.3.3. Řešení problému nelinearity cenových funkcí

Každé obecné zařízení v naší úloze má své příjmy a výdaje. Na koláčových grafech v obrázku 2.5 jsou znázorněny hlavní z nich. Do výdajů se počítají především mzdy, nákup potřebných chemikálií a nakládání s rezidui. V příjmech tvoří většinu vyrobená energie v podobě tepla a elektřiny spolu s cenou za zpracování. Všechny tyto parametry jsou přímo závislé na kapacitě zařízení a dalších modelem neovlivnitelných podmínkách. Pouze cena za zpracování je nastavitelná.



Obrázek 2.5: Příjmy a výdaje obecného zařízení na zpracování komodity, Zdroj: [12]

Suma veškerých výdajů spolu s případnou cenou za stavbu zařízení, ode které odečteme sumu peněžní odměny získané z prodeje vyrobené energie, je funkce závislá na kapacitě zařízení. Tato funkce je zobrazena na grafu 2.6. Jinak řečeno, graf zobrazuje minimální cenu za zpracování veškerého možného odpadu v kapacitě zařízení takovou, aby byla stavba zařízení výhodná vzhledem k požadavkům úlohy. Tím se dostáváme k poslednímu problému předchozího modelu základní úlohy a tím je jeho neschopnost simulovat nelinearitu cenových funkcí, kterou lze vidět například na situaci s kapacitními zařízeními.



Obrázek 2.6: Graf závislosti celkového nákladu na zpracování veškerého materiálu dovezeného do zařízení na kapacitě zařízení.

Červená křivka zobrazuje reálnou nelineární závislost, černá po částech spojitá lineární křivka je aproximace reálné křivky, kterou jsme schopni nasimulovat v celočíselném

lineárním modelu pomocí již zmiňované SOS2 proměnné v sekci GAMS. Ukázaná chyba se mění v závislosti na diskretizaci vedoucí na po částech lineární aproximační funkci.

Při snaze aproximovat nelineární funkci po částech lineární funkcí pomocí SOS2 proměnné je třeba nejprve vytvořit dva umělé diskrétní parametry, které budou nahrazovat spojité parametry na ose  $x$  a  $y$ . Nazvěme tyto diskrétní parametry  $cena = [cena_1, \dots, cena_n]$  a  $kapacita = [kapacita_1, \dots, kapacita_n]$ . K těmto parametrům bude nyní přiřazena diskrétní SOS2 proměnná  $sos = [sos_1, \dots, sos_n]$ . Oba parametry i proměnná mají stejné množství prvků  $n$  a jsou indexovány indexovou množinou  $k \in K$ . Pro stručnost nyní nazýváme kapacitu potencionálního zařízení pouze objemem a celkové náklady na zpracování pouze náklady. Dále platí dva vztahy:

$$objem = sos_1 \cdot kapacita_1 + \dots + sos_n \cdot kapacita_n, \quad (2.7)$$

$$naklady = sos_1 \cdot cena_1 + \dots + sos_n \cdot cena_n \quad (2.8)$$

Vztahy (2.7) a (2.8) je dána závislost nákladů na objemu, kterou nastavíme tak, aby odpovídala závislosti reálné křivky v daných bodech pro  $k = 1, 2, 3, \dots, n$ .

Když je na vstupu  $naklady = x_0$  mohou nastat dvě situace.

1)  $\exists k : kapacita_k = x_0 \Rightarrow sos = [sos_1, \dots, sos_{k-1}, sos_k, sos_{k+1}, \dots, sos_n] = [0, \dots, 0, 1, 0, \dots, 0] \Rightarrow naklady = 0 + \dots + 0 + sos_k \cdot cena_k + 0 + \dots + 0 = 1 \cdot cena_k$

2)  $\exists k : kapacita_k < x_0 < kapacita_{k+1} \Rightarrow$

$$\frac{sos_{k+1}}{sos_k} = \frac{x_0 - kapacita_k}{kapacita_{k+1} - x_0}$$

což spolu s podmínkou  $sos_k + sos_{k+1} = 1$ , která vyplývá z charakteru SOS2 proměnné, dává právě lineární trend každé části aproximační funkce a platí

$$naklady = sos_k \cdot cena_k + sos_{k+1} \cdot cena_{k+1}.$$

## Použité symboly

### Indexové množiny

$i \in I$  indexová množina uzlů

$e \in E$  indexová množina hran

$k \in K$  indexová množina pro SOS2 proměnnou

### Parametry

$P^{trans}$  cena za km dopravy

$V^{max}$  maximální výhřevnost směsi

$V^{min}$  minimální výhřevnost směsi

$V_e$  výhřevnost komodity putující hranou  $e$

$C_i$  kapacita zařízení  $i$

$M_i$	produkce uzlu $i$
$P_i^{treat}$	cena za zpracování v zařízení $i$
$D_e$	délka hrany $e$
$A_{i,e}$	prvky incidenční matice
$C_{i,k}^{pot}$	uměle vytvořený parametr určující kapacitu potenciaálního zařízení
$P_{i,k}^{pot}$	uměle vytvořený parametr určující ceny za zpracování v novém zařízení

### Proměnné

$\mu_{i,k}^{pot}$	SOS2 proměnná na které závisí parametry $C_{i,k}^{pot}$ a $P_{i,k}^{pot}$
$x_e$	proměnná udávající množství komodity dopravované po dané hraně $e$
$s_i$	proměnná vyjadřující celkové množství směsi zpracované v zařízení
$z$	proměnná sčítající celkové náklady na dopravu a zpracování

### Účelová funkce

$$z = \sum_{i \in I} \sum_{e \in E} A_{i,e} x_e P_i^{treat} + \sum_{e \in E} x_e D_e P^{trans} + \sum_{k \in K} \sum_{i \in I} \mu_{i,k}^{pot} P_{i,k}^{pot}$$

První část účelové funkce sčítá cenu za zpracování komodit, ale pouze v již existujících zařízeních. Druhá část sčítá cenu za dopravu komodit s tím rozdílem od předchozího modelu, že sčítá pouze přes hrany s předem vypočítanými vzdálenostmi. Jelikož je daná komodita vázaná k hraně, nemusí již řešit ani producenta ani typ komodity. Třetí část počítá cenu za zpracování komodit v potenciaálních zařízeních dle vzoru v předchozím textu. Přes všechny  $k$  a potenciaální zařízení násobí SOS2 proměnnou  $\mu_{i,k}^{cap}$  s uměle vytvořeným diskretním parametrem pro výpočet nákladů na zpracování  $P_{i,k}^{cap}$ .

### Omezující podmínky

$$M_i + \sum_{e \in E} A_{i,e} x_e = s_i \quad \forall i \in I, \quad (2.9)$$

$$s_i \leq C_i \quad \forall i \in I, \quad (2.10)$$

$$s_i = \sum_{k \in K} \mu_{i,k}^{pot} C_{i,k}^{pot} \quad \forall i \in I, \quad (2.11)$$

$$\sum_{k \in K} \mu_{i,k}^{pot} = 1 \quad \forall i \in I, \quad (2.12)$$

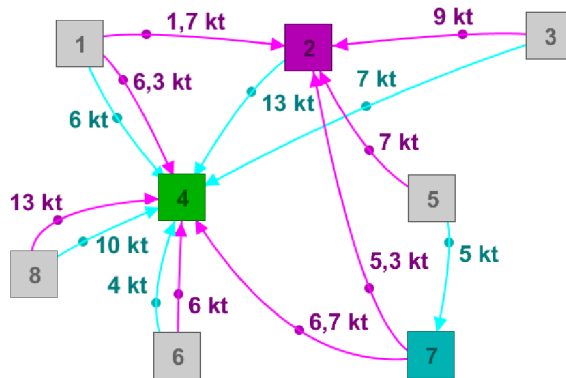
$$s_i V^{min} \leq \sum_e A_{i,e} x_e V_e \leq s_i V^{max} \quad \forall i \in I, \quad (2.13)$$

$$x_e \geq 0 \quad \forall e \in E, \quad (2.14)$$

$$s_i \geq 0 \quad \forall i \in I. \quad (2.15)$$

Kromě dvou podmínek týkajících se potencionálních zařízení jsou všechny ostatní podobné podmínkám z předchozího modelu, avšak vzhledem k novému způsobu tvorby modelu mají odlišný zápis. První podmínka (2.9) napomáhá přehlednosti modelu tím, že zobrazuje celkové množství zpracované směsi komodit v jedné nezáporné proměnné  $s_i$ . Nezápornost proměnné  $s_i$  je zajištěna v podmínce (2.15). Pro uzel  $i$  přiděluje incidenční matice  $A_{i,e}$  nulu hranám, které nemají ani jeden konec v  $i$ ,  $-1$  orientovaným hranám, které začínají v uzlu  $i$  a  $1$  hranám, které končí v  $i$ . Člen  $\sum_{e \in E} A_{i,e} x_e$  se tedy rovná rozdílu všech přivezených komodit a všech odvezených. Nezápornost proměnné  $x_e$  je zajištěna v podmínce (2.14). Podmínka (2.10) omezuje celkové množství zpracované směsi kapacitou zařízení. Energetické omezení shora a zdola je vyjádřeno v podmínce (2.13). Omezení (2.11) má stejnou funkci jako vztah (2.7) při řešení nelinearity cenových funkcí. Vyjadřuje celkové množství zpracované směsi jako skalární součin SOS2 proměnné  $\mu_{i,k}^{cap}$  a umělého parametru  $O_{i,k}$ . Z této rovnice model získává výsledný tvar proměnné  $\mu_{i,k}^{cap}$ , díky které je pak možné v účelové funkci vypočítat cenu zpracování v potencionálních zařízeních. Poslední omezení (2.12) je normování SOS2 proměnné. Tedy, že součet jejích členů pro jedno zařízení je rovno jedné.

### 2.3.4. Řešení celočíselné lineární úlohy



Obrázek 2.7: Výsledný graf celočíselné lineární úlohy

V řešení na obrázku 2.7 je zobrazeno rozhodnutí modelu o stavbě zařízení v uzlu 2 a naopak o nepostavení zařízení v uzlu 8. Dále je vidět snahu modelu o šetření nákladů na dopravě tím, že co nejvíce komodit je dopravováno pouze do nejbližšího zařízení.

### 2.3.5. Zdrojový kód v GAMSu

#### Model 2.2: Úloha MILP

```

1 set
2 i          indexová množina uzlů
3 si(i)     podmnožina indexové množiny uzlů pro zařízení pro komodity s vyšší
            výhřevností
4 pi(i)     podmnožina indexové množiny uzlů pro potencionální zařízení
5 e         indexová množina hran

```

```

6 ve(e)    podmnožina indexové množiny hran s vyšší výhřevností
7 k        index pro SOS2 proměnnou určující C_pot a P_pot
8 ;
9 scalar
10 P_trans  fixní cena za dopravu jeden kilometr jedné kilo tuny komodity /
11          10000/
12 V_min    minimální výhřevnost zpracovávané směsi v zařízení /10/
13 V_max    maximální výhřevnost zpracovávané směsi v zařízení /13/
14 ;
15 parameters
16 varz      parametr pro výpis celkových nákladů
17 varx(e)   parametr pro výpis množství komodit dopravovaných po hraná
18           ch
19 vars(i)   parametr pro výpis množství směsi zpracovávané v zařízení
20 varSOS(k,pi) parametr pro výpis SOS2 proměnné
21 A(i,e)    incidenční matice
22 D(e)      délky hran
23 M(i)      produkce komodity v uzlu
24 C(i)      kapacita zařízení v uzlu
25 P_treat(i) cena za zpracování jedné tuny směsi
26 V(ve)     Výhřevnost komodity putující na hraně
27 C_pot(k,pi) parametr kapacit potencionálních zařízení
28 P_pot(k,pi) paramet cen za zpracování v potencionálních zařízeních
29 ;
30 positive variables
31 x(e)      množství komodity dopravované po hraně
32 s(i)      množství zpracovávané směsi v daném zařízení
33 ;
34 SOS2 variable
35 SOS(k,pi) proměnná určující kapacitu potencionálních zařízení
36 ;
37 variable
38 z         Celkové náklady
39 ;
40 equations
41 ucelfce, omez1(i), omez2(i), omez3(pi), omez4(pi), omez5(si), omez6(si)
42 ;
43 * Účelová funkce obsahující náklady na zpracování v existujících zařízeních
44   , náklady na dopravu a náklady na zpracování v nově postavených zařízeních
45   ch
46 ucelfce.. z =e= sum((i,e), A(i,e)*x(e)*P_treat(i)) + sum(e, x(e)*D(e)*
47   P_trans) + sum((k,pi), SOS(k,pi)*P_pot(k,pi))
48 ;
49 * Výpočet množství zpracované směsi v zařízení v uzlu
50 omez1(i).. M(i) + sum(e, A(i,e)*x(e)) =e= s(i)
51 ;
52 * Omezení množství zpracovávané směsi kapacitou zařízení
53 omez2(i).. s(i) =l= C(i)
54 ;
55 * Rovnost kapacity nového zařízení a množství zpracovávané směsi
56 omez3(pi).. s(pi) =e= sum(k, SOS(k,pi) * C_pot(k,pi))
57 ;
58 * Součet členů SOS2 proměnné pro jedno zařízení je rovno jedné
59 omez4(pi).. sum(k, SOS(k,pi)) =e= 1
60 ;
61 * Podmínka minimální výhřevnosti zpracovávané směsi

```

```

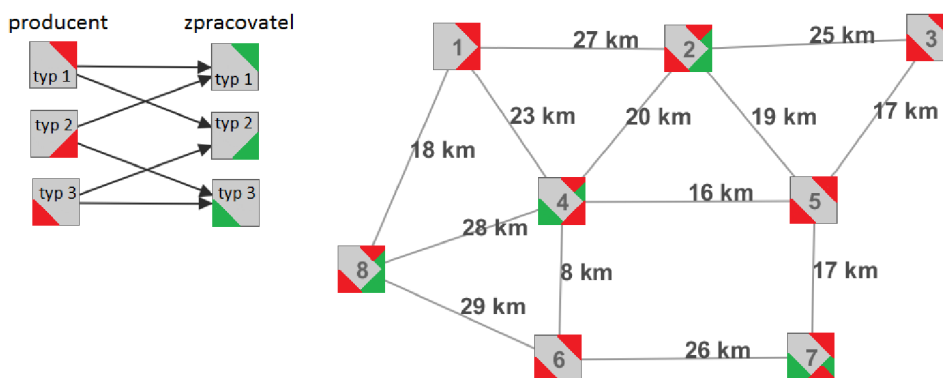
57 omez5(si).. sum(ve, A(si,ve)*x(ve))*V_min =l= sum(ve, A(si,ve)*x(ve)*V(ve))
58 ;
59 * Podmínka maximální výhřevnosti zpracovávané směsi
60 omez6(si).. sum(ve, A(si,ve)*x(ve)*V(ve)) =l= sum(ve, A(si,ve)*x(ve))*V_max
61 ;
62 model preshrany /all/
63 ;
64 $call GDXXRW.exe maly_model_bipart_SOS.xlsx set=i rng=i rdim=1 set=e rng=e
    rdim=1 set=si rng=si rdim=1 set=pi rng=pi rdim=1 set=ve rng=ve rdim=1
    set=k rng=ik rdim=1 par=A rng=a rdim=1 cdim=1 par=D rng=v rdim=1 cdim=0
    par=M rng=p rdim=1 cdim=0 par=C rng=k rdim=1 cdim=0 par=P_treat rng=Cena
    rdim=1 cdim=0 par=V rng=vyh rdim=1 cdim=0 par=C_pot rng=C_pot rdim=1
    cdim=1 par=P_pot rng=P_pot rdim=1 cdim=1
65 $GDXIN maly_model_bipart_SOS.gdx
66 $LOAD i e si pi ve k A D V M C P_treat C_pot P_pot
67 $GDXIN
68 option
69 mip=cplex
70 * Určení cplex jako řešiče pro tuto úlohu
71 solve preshrany using mip minimizing z
72 * Řešení modelu minimalizace účelové funkce pomocí smíšeného celočíselného
    programování
73 ;
74 display i, e, si, pi, ve, k, D, C, P_treat, V, M, A, C_pot, P_pot, s.l, SOS
    .l, x.l, z.l
75 ;
76 varz = z.l
77 ;
78 varx(e) = x.l(e)
79 ;
80 vars(i) = s.l(i)
81 ;
82 varSOS(k,pi) = SOS.l(k,pi)
83 ;
84 EXECUTE_UNLOAD "maly_model_bipart_SOS.gdx" varz varx vars varSOS
85 EXECUTE 'gdxxrw.exe maly_model_bipart_SOS.gdx par=varz rng=vysledky!A2'
86 EXECUTE 'gdxxrw.exe maly_model_bipart_SOS.gdx par=varx rng=vysledky!A4'
87 EXECUTE 'gdxxrw.exe maly_model_bipart_SOS.gdx par=vars rng=vysledky!A7'
88 EXECUTE 'gdxxrw.exe maly_model_bipart_SOS.gdx par=varSOS rng=vysledky!A10'

```

---

## 2.4. Úloha MILP rozšířená o penalizační funkci

V této úloze dochází ke směšování různých typů komodit a u některých typů lze rozhodovat o využití či nevyužití jejich energetického potenciálu, viz 2.8. Rozšíření o penalizaci bylo přidáno kvůli snaze vyprodukovat co nejvíce tepelné energie. Každé zařízení, ve kterém dochází k produkci energie, potřebuje, aby celková směs do něj přivezená měla určitou referenční výhřevnost. Když zařízení bude dostávat směs s referenční výhřevností, bude produkovat určité množství energie. Pokud však bude dostávat komodity, jejichž směs má výhřevnost nižší, bude také relativně k množství zpracované směsi produkovat méně energie a tím pádem budou příjmy zařízení nižší. Tento rozdíl je třeba dorovnat penalizační funkcí.



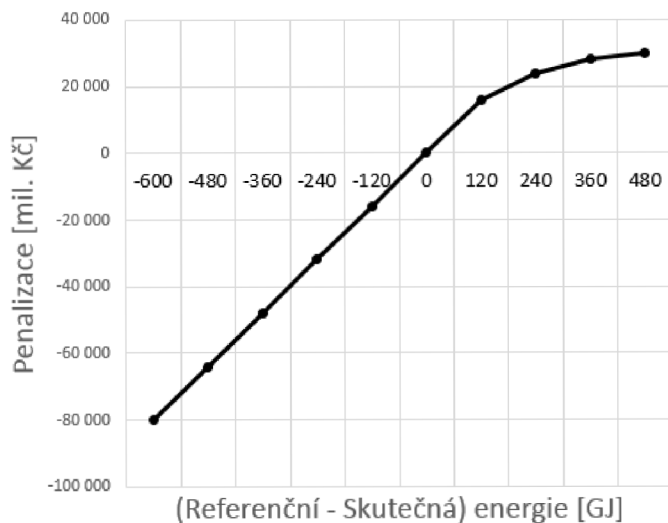
Obrázek 2.8: Neorientovaný graf úlohy celočíselného lineárního programování s rozšířením o penalizaci

Model obsahuje osm uzlů a třináct hran. Na grafu nalevo jsou zobrazené možné způsoby zpracování daných komodit, kde červené trojúhelníky označují typy producentů a zelené typy zařízení (zpracovatelů). V úloze zůstává rozhodování o budoucí existenci potenciálních zařízení. V tomto modelu jsou to zařízení typu 1 v uzlu 2 a 8 a dále zařízení typu 3 v uzlu 7.

### 2.4.1. Řešení výpočtu penalizace nedostatečné výhřevnosti

Penalizace jako část účelové funkce je, stejně jako výpočet kapacity potencionálních zařízení, nelineárně závislá na jiném parametru, v tomto případě na výhřevnosti výsledné směsi ke zpracování v daném zařízení, viz obrázek 2.9.

Přepočítání se provádí stejným způsobem jako přepočítání kapacity na cenu za zpracování v obrázku 2.6, tedy pomocí omezení (2.17) se vypočítá hodnota SOS2 proměnné  $\mu_{i,k}^{pen}$  a pomocí ní se pak dopočítá penalizace v účelové funkci.



Obrázek 2.9: Graf závislosti penalizace zařízení na rozdílu referenční energie a skutečné vyrobené energie.

## Použité symboly

### Indexové množiny

$k \in K$  indexová množina pro SOS2 proměnnou  $\mu_{i,k}^{pen}$

### Parametry

$V_i^{ref}$  referenční výhřevnost pro dané zařízení

$U_k^{pen}$  uměle vytvořený diskrétní parametr vyjadřující energii vyrobenou v zařízení

$P_k^{pen}$  uměle vytvořený diskrétní parametr vyjadřující penalizaci zařízení

### Proměnné

$\mu_{i,k}^{pen}$  SOS2 proměnná na které závisí parametry  $U_k^{pen}$  a  $P_k^{pen}$

## Účelová funkce

$$z = \sum_{i \in I} \sum_{e \in E} A_{i,e} x_e P_i^{treat} + \sum_{e \in E} x_e D_e P^{trans} + \sum_{k \in K} \sum_{i \in I} (\mu_{i,k}^{pot} P_{i,k}^{pot} + \mu_{i,k}^{pen} P_k^{pen})$$

V účelové funkci přibývá poslední člen přičítající náklady na penalizaci za míru splnění či nesplnění vyžadované vyrobené referenční energie. Uměle vytvořený diskrétní parametr penalizace  $P_k^{pen}$  je závislý na proměnné  $\mu_{i,k}^{pen}$  vypočítané pomocí následujících omezení.



## Omezení

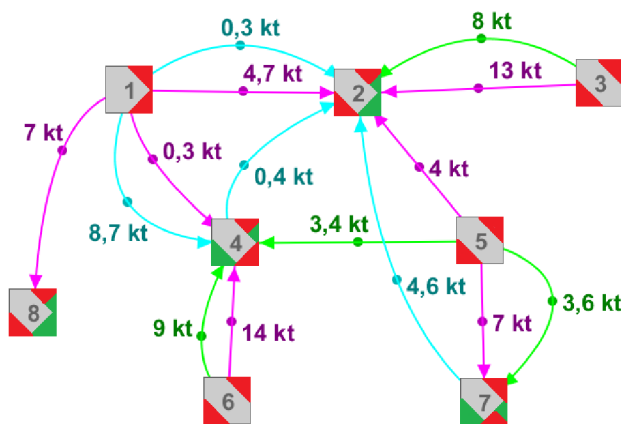
Omezení (2.9) až (2.15) jsou v této úloze odpovídající. Následující dvě podmínky souvisí s výpočtem SOS2 proměnné  $\mu_{i,k}^{pen}$ .

$$\sum_{k \in K} \mu_{i,k}^{pen} = 1 \quad \forall i \in I, \quad (2.16)$$

$$\sum_{k \in K} \mu_{i,k}^{pen} U_k^{pen} = V_i^{ref} s_i - \sum_{e \in E} A_{i,e} x_e V_e \quad \forall i \in I. \quad (2.17)$$

Pomocí podmínky (2.16) dochází k tomu, že součet oněch dvou po sobě jdoucích prvků se bude rovnat jedné. Podmínka (2.17) je výpočet sumy součinů  $\sum_{k \in K} \mu_{i,k}^{pen} U_k^{pen}$ , který udává, do jaké míry zařízení plní požadavky na výrobu energie. Tato hodnota je rovna rozdílu energie, kterou je možné vyprodukovat při referenční výhřevnosti směsi a skutečné vyprodukované energie při skutečné výhřevnosti. Na pravé straně se vyskytuje rozdíl referenční energie, vypočítané pomocí součinu vyžadované výhřevnosti na jednu kilotunu komodity s množstvím zpracované směsi  $V_i^{ref} s_i$  a skutečné energie, vypočítané stejným způsobem jen po složkách, protože po každé hraně zařízení přijímá komoditu s jinou výhřevností. Proto se množství komodity počítá jako  $A_{i,e} x_e$ .

### 2.4.2. Řešení úlohy s rozšířením o penalizaci

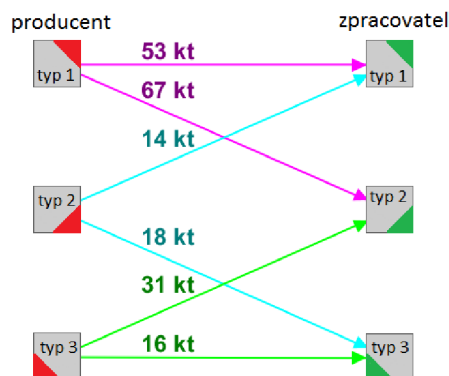


Obrázek 2.10: Výsledný graf úlohy rozšířené o penalizaci

Na grafu na obrázku 2.10 je zobrazeno jedno z možných řešení úlohy s rozšířením o penalizaci. Kromě zařízení typu 1 v uzlu 8 byla všechna ostatní zařízení vyhodnocena jako užitečná. V tomto uzlu je ovšem zároveň zařízení typu dva, proto nebylo třeba vyvážit jakoukoliv komoditu mimo uzel.

Na následujícím obrázku 2.11 je vidět, jak bylo rozděleno zpracování komodit mezi zařízení. Lze si povšimnout, že žádný typ zařízení nebyl úplně zanedbán.

### 2.4.3. Zdrojový kód v GAMSu



Obrázek 2.11: Graf výsledného rozdělení zpracovaných komodit do zařízení.

### Model 2.3: Úloha MILP s penalizací

```

1  set
2  i          indexová množina uzlů
3  vi(i)     podmnožina indexové množiny uzlů zařízení využívajících výhřevnost
             komodit
4  pi(i)     podmnožina indexové množiny uzlů potencionálních zařízení
5  e          indexová množina hran
6  ve(e)     podmnožina indexové množiny hran směřujících do zařízení využívajíc
             í výhřevnost komodit
7  k          indexová množina pro SOS2 proměnné
8  ;
9  scalar
10 P_trans   fixní cena za kilometr dopravy jedné kilotuny komodity /10000/
11 V_min     minimální výhřevnost zpracovávané směsi /12/
12 V_max     maximální výhřevnost zpracovávané směsi /19/
13 ;
14 parameters
15 varz      parametr pro výpis celkových nákladů
16 varx(e)   parametr pro výpis množství komodity dopravované po hraně
17 vars(i)   parametr pro výpis množství směsi zpracovávané v zařízení
18 varSOS(k,pi) parametr pro výpis SOS2 proměnné určující C_pot
19 varSS(k,vi) parametr pro výpis SOS2 proměnné určující U_pen
20 A(i,e)    incidenční matice
21 D(e)      délka hran
22 M(i)      produkce komodit v uzlech
23 C(i)      kapacita zařízení v uzlech
24 P_treat(i) cena za zpracování jedné komodity v zařízení
25 V(e)      výhřevnost komodity na hraně
26 RV(vi)   referenční výhřevnost pro dané zařízení
27 C_pot(k,pi) kapacita potencionálního zařízení
28 P_pot(k,pi) cena za zpracování v potencionálním zařízení
29 U_pen(k)  Míra plnění nároků na vyprodukovanou energii v zařízení
30 P_pen(k)  penalizace zařízení
31 ;
32 positive variables
33 x(e)      množství komodity dopravované po hraně
34 s(i)      množství směsi zpracovávané v zařízení
35 ;

```

```

36 sos2 variable
37 SOS(k,pi)          proměnná určující C_pot a P_pot
38 SS(k,vi)           proměnná určující U_pen a P_pen
39 ;
40 variable
41 z                  celkové náklady
42 ;
43 equations
44 ucelfce, omez1(i), omez2(i), omez3(pi), omez4(pi), omez5(vi), omez6(vi),
    omez7(vi), omez8(vi)
45 ;
46 * Účelová funkce obsahující náklady na zpracování v existujících i
    potencionálních zařízeních, dopravu komodit a penalizaci zařízení
47 ucelfce.. z =e= sum((i,e), A(i,e)*x(e)*P_treat(i) + sum(e, x(e)*D(e)*
    P_trans) + sum(k, sum(pi, SOS(k,pi)*P_pot(k,pi)) + sum(vi, SS(k,vi)*
    P_pen(k)))
48 ;
49 * Výpočet množství zpracované komodity v uzlu
50 omez1(i).. M(i) + sum(e, A(i,e)*x(e)) =e= s(i)
51 ;
52 * Omezení zpracovávané komodity kapacitou zařízení
53 omez2(i).. s(i) =l= C(i)
54 ;
55 * Rovnost kapacity nového zařízení a množství zpracovávané komodity v uzlu
56 omez3(pi).. s(pi) =e= sum(k, SOS(k,pi) * C_pot(k,pi))
57 ;
58 *Součet členů proměnné SOS pro jedno zařízení je rovno jedné
59 omez4(pi).. sum(k, SOS(k,pi)) =e= 1
60 ;
61 * Omezení minimální výhřevnosti zpracovávané směsi
62 omez5(vi).. sum(ve, A(vi,ve)*x(ve))*V_min =l= sum(ve, A(vi,ve)*x(ve))*V(ve))
63 ;
64 * Omezení maximální výhřevnosti zpracovávané směsi
65 omez6(vi).. sum(ve, A(vi,ve)*x(ve))*V(ve) =l= sum(ve, A(vi,ve)*x(ve))*V_max
66 ;
67 * Součet členů proměnné SS pro jedno zařízení je rovno jedné
68 omez7(vi).. sum(k, SS(k,vi)) =e= 1
69 ;
70 * Výpočet proměnné SS jako rozdíl referenční energie a skutečné
    vyprodukované energie zařízení
71 omez8(vi).. sum(k, U_pen(k)*SS(k,vi))=e= RV(vi)*s(vi) - sum(ve, A(vi,ve)*x(
    ve))*V(ve))
72 ;
73 model preshrany /all/
74 ;
75 $call GDXXRW.exe mm_deset.xlsx set=i rng=i rdim=1 set=vi rng=vi rdim=1 set=
    pi rng=pi rdim=1 set=e rng=e cdim=1 set=ve rng=hrdovi rdim=1 set=k rng=
    ik rdim=1 par=A rng=A rdim=1 cdim=1 par=D rng=v rdim=1 par=M rng=p rdim
    =1 par=C rng=k rdim=1 par=P_treat rng=ce rdim=1 par=V rng=vyh rdim=1 par
    =RV rng=rv rdim=1 par=C_pot rng=objem rdim=1 cdim=1 par=P_pot rng=cenabr
    rdim=1 cdim=1 par=U_pen rng=vyhspal rdim=1 par=P_pen rng=penal rdim=1
76 $GDXIN mm_deset.gdx
77 $LOAD i vi pi e ve k A D M C P_treat V RV C_pot P_pot U_pen P_pen
78 $GDXIN
79 option
80 MIP=cplex;

```

```
81 solve preshrany using MIP minimizing z;
82 display i, vi, pi, e, ve, P_trans, k, A, D, M, C, P_treat, V, RV, C_pot,
      P_pot, U_pen, P_pen, x.l, z.l, s.l, SOS.l, SS.l
83 ;
84 varz = z.l
85 ;
86 varx(e) = x.l(e)
87 ;
88 vars(i) = s.l(i)
89 ;
90 varSOS(k,pi) = SOS.l(k,pi)
91 ;
92 varSS(k,vi) = SS.l(k,vi)
93 ;
94 EXECUTE_UNLOAD "mm_deset.gdx" varz varx vars varSOS varSS
95 EXECUTE 'gdxxrw.exe mm_deset.gdx par=varz rng=vysledky!A2'
96 EXECUTE 'gdxxrw.exe mm_deset.gdx par=varx rng=vysledky!A4'
97 EXECUTE 'gdxxrw.exe mm_deset.gdx par=vars rng=vysledky!A7'
98 EXECUTE 'gdxxrw.exe mm_deset.gdx par=varSOS rng=vysledky!A10'
99 EXECUTE 'gdxxrw.exe mm_deset.gdx par=varSS rng=vysledky!A15'
```

---

## 3. Implementace dat z ISOH do modelu

Tato kapitola se zabývá implementací reálných dat z ISOH České republiky do komplexního modelu z kapitoly 2.4. Ovšem předtím je potřeba zmínit některé důležité informace z odpadového hospodářství dnešních dnů.

### 3.1. Případová studie

Se vstupem České Republiky do EU se naše země zavázala plnit určité trendy, jedním z nich je například přesměrování toku z nižších stupňů hierarchie nakládání s odpady do preferovanějších způsobů nakládání. V roce 2016 tedy Ministerstvo životního prostředí předložilo do meziresortního připomínkového řízení návrh změn Zákona o odpadech [13]. Jednou z klíčových snah tohoto návrhu je maximalizovat materiálové využití odpadu. Pokud by byl odpad z nějakého důvodu nevyužitelný tímto způsobem, byla by zde alespoň snaha o energetické využití odpadu (EVO). Změna způsobu nakládání s odpady bude vyžadovat adekvátní přizpůsobení zpracovatelské infrastruktury.

Díky článku [1] a výpočtovému nástroji, který je v článku prezentován, máme v dnešní době přesnější informace o tom, kolik která obec s rozšířenou působností (ORP) vyprodukuje odpadu daného typu.

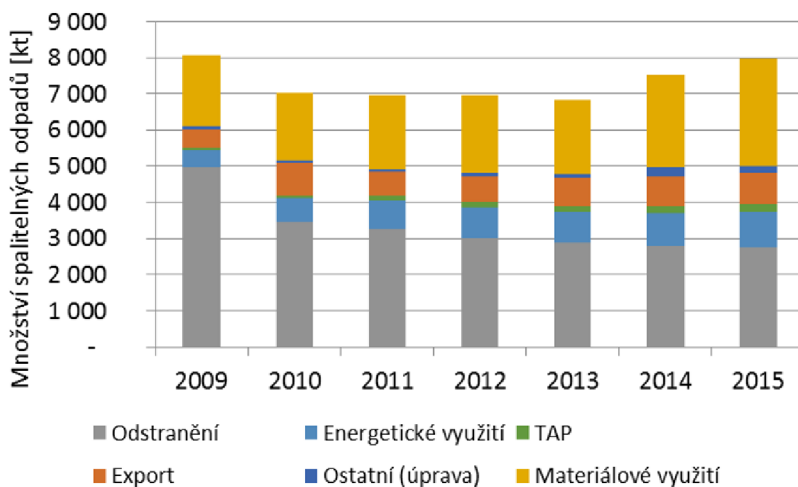
Tato práce se zabývá pouze EVO, tedy spalitelnými odpady (SO). Tyto odpady budou tedy pro účel této práce definovány jako průnik přijímaných katalogových čísel odpadů do existujících zařízení EVO (ZEVO) v ČR v roce 2015 spolu s katalogovými čísly odpadů vhodných pro výrobu paliv. V tuto dobu existovala taková zařízení tři: ZEVO Malešice, SAKO Brno a TERMIZO Liberec. Data byla čerpána z [14], [15], [16]. Zvýrazněná katalogová čísla představují odpady, které jsou potenciálně vhodné pro výrobu paliv z odpadů, katalogové číslo 191210 reprezentuje vzniklé palivo z odpadů, tzv. TAP (tuhé alternativní palivo).

#### Katalogová čísla SO:

**020103**, 020107, **020104**, 020304, 020601, **030101**, **030105**, **030301**, **030307**, **030308**, **040101**, **040108**, **040209**, **040221**, **040222**, **070213**, **090108**, **120105**, **150101**, **150102**, **150103**, **150105**, 150106, **150109**, **150203**, **160119**, 160199, 170201, **170203**, **170302**, **170604**, 180104, **191201**, **191204**, **191207**, **191208**, **191210**, 191212, **200101**, **200110**, **200111**, 200138, **200139**, 200301, 200302, 200303, 200307

#### 3.1.1. Nakládání se SO v ČR v letech 2009 až 2015

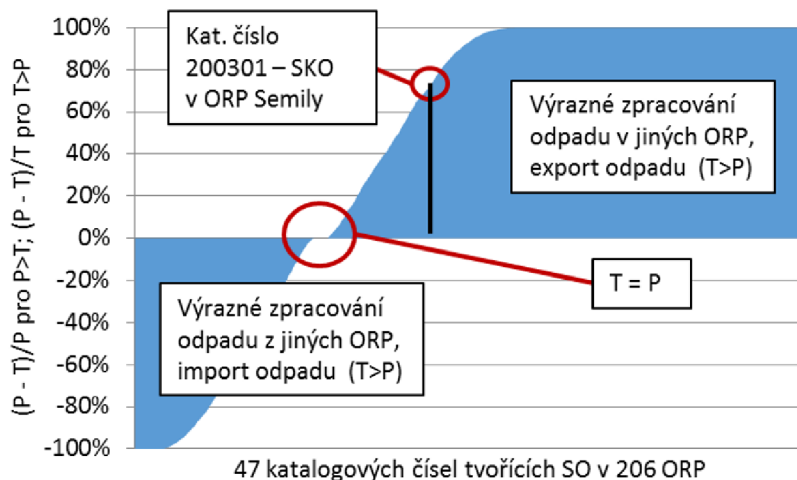
Na obrázku 3.1 lze vidět kolísavou situace v produkci SO v letech 2009 až 2015. Hlavně je ale vidět zlepšení v procentuálním zastoupení skutečně využitých odpadů, ať už materiálově a nebo energeticky, oproti bezcenné likvidaci například skládkováním. V roce 2009 to bylo zhruba 37 %, zatím co v roce 2015 už to bylo přes 63 %.



Obrázek 3.1: Nakládání se SO v ČR v letech 2009 až 2015, viz [1]

### 3.1.2. Bilance mezi produkcí a zpracováním odpadu v ORP

Z dat veřejného informačního systému odpadového hospodářství (VISOH) vyplývá, že pro ORP se větší část odpadu zpracovává jinde než v místě produkce. Graf na obrázku 3.2 je bilance mezi produkcí (P) a zpracováním (T) všech SO podle katalogových čísel ve všech ORP pro rok 2015.



Obrázek 3.2: Poměr produkce a zpracování odpadu v jednotlivých ORP, zdroj: VUT na základě zpracování dat VISOH, viz [1]

Bilance pro SKO v ORP Semily je uvedena jako příklad. Pro záporné hodnoty, kde  $T > P$  (je zde více odpadu zpracováváno než vyprodukováno), platí vzorec  $\frac{P-T}{T}$ . Hodnota  $-100\%$  podle grafu znamená, že se daného typu odpadu v daném uzlu zpracovává určité množství zatímco je produkce nulová. Pro kladné hodnoty, kde  $P > T$  (je zde více odpadu vyprodukováno než zpracováváno), platí vzorec  $\frac{P-T}{P}$ . Hodnota  $100\%$  znamená, že se v daném uzlu nezpracovalo žádné množství daného odpadu, ale při tom se zde tento typ odpadu produkuje. Ve velmi omezeném množství se vyskytují i případy ekvivalentní produkce a zpracování  $T = P$ .

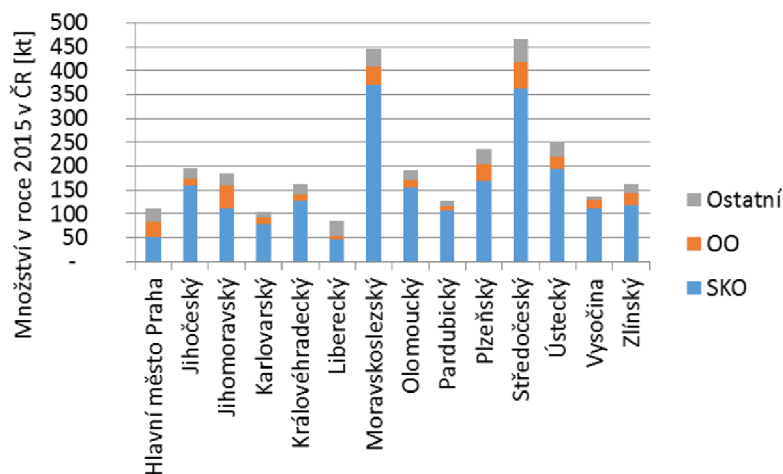
### 3.1.3. Potenciál SO pro EVO

Pro některé územní celky vzniká při nakládání se SO nesoulad mezi produkcí a zpracováním. Z tohoto důvodu je zřejmé, že je SO přepravován mezi jednotlivými ORP. Pracovní databáze ISOH sice pracuje při zpracovávání dat s kódy popisující předání a převzetí odpadu, avšak není jasný producent tohoto odpadu, proto je analýza přepravy SO výrazně komplikovanější, než analýza produkce a nakládání se SO.

Jak bylo již dříve zmíněno, byl pro účel simulace přepravy SO autory článku [1] na základě optimální svozové úlohy vyvinut přístup pro rekonstrukci toku odpadů na území ČR.

Množství SO využitelné pro EVO vnímá tato práce jako potenciál, který podle dat t roku 2015 zatím není v ČR využíván. Za předpokladu, že ČR zůstane do roku 2024 v EU, bude třeba pro splnění podmínky zákazu skladování veškerého odpadu vhodného pro EVO rozšířit infrastrukturu OH v ČR tak, aby všechen tento potenciál mohl být využit. Preferované využití by mělo být energetické.

V grafu na obrázku 3.3 jsou vyobrazena přesná množství tohoto potenciálu SO pro jednotlivé kraje. Mimo jiné je z grafu možno zjistit, že přibližně 90% tohoto potenciálu SO tvoří odpad s kat. č. 200301, tedy SKO a značná část zbytku tvoří odpad s kat. č. 200307, tedy objemný odpad.



Obrázek 3.3: Potenciál SO pro navýšení energetické využití v roce 2015 v ČR, zdroj: VUT na základě komplexního výpočtu, viz [1]

## 3.2. Strukturalizace dat

Jelikož data z ISOH jsou velmi obsáhlá, tak se pro práci s nimi vyplatí využívat Visual Basic for Applications (VBA). Tento programovací jazyk byl vyvinut pro programy Microsoft Office, tedy i pro Microsoft Excel (ME), kde je ho využíváno pro tuto práci. Pro zjednodušení práce s daty jsou využívána takzvaná *makra*, což jsou menší aplikace napsané ve VBA. Aby mohl ME využívat maker, nestačí obyčejný sešit .xls, ale uživatel musí používat speciální sešit s příponou .xlsm, neboli sešit ME s podporou maker. Většinou jsou tyto aplikace využívány pro často opakovanou činnost a nebo pro činnost, kterou je třeba naprosto stejně udělat například pro několik tisíc řádků, jak je tomu i v našem případě.

### Tlačítko Generate

V sešitu Implementace.xlsm, který je použit jako zdroj vstupů, je vytvořeno tlačítko, ke kterému se váže makro jménem "Generate", jehož kód lze vidět v následujícím textu. "Sub" označuje začátek makra, následuje jméno makra a do závorek se vkládají proměnné

```
Sub Generate()  
Call Hrany  
Call Vzdalenosti  
Call Incidence  
End Sub
```

pro vstupy, ty ale v našem případě definujeme až v kódu, proto je prostor mezi závorkami prázdný. "End Sub" potom značí konec makra. "Call" je příkaz pro zavolání jiného makra. Zavolaná makra budou popsána v následujícím textu.

### Makro Hrany

Makro Hrany zastává práci vypsání veškerých užitečných orientovaných hran. Jako vstup vnímá číselné označení uzlu (ORP) pod nadpisem "ID" a také jeho typ pod nadpisem "Typ odpadu/zařízení", neboli zda je producent a jaký typ odpadu produkuje, nebo zda je zpracovatel a jaký typ zařízení používá. "SKO" je řetězec, který makro vnímá jako kód pro producenta odpadu typu SKO. Tabulka vstupů je vidět na obrázku 3.4.

Nejprve dochází pomocí cyklu While Do – > Loop ke spočítání uzlů a rozdělení na počet producentů a zpracovatelů. Následně makro vytvoří dvousloupcové pole udávající přijímaný odpad od jednotlivých zařízení. Dále postupně prochází toto pole, vždy zapíše jen jeden typ producenta a jeden typ zpracovatele, který od něj může přijímat odpad, a postupně vypíše všechny možné cesty, kterými se odpad může ubírat. Pro danou cestu vypíše do jednoho řetězce nejprve uzel producenta, za něj znak "\_" a za něj uzel zpracovatele.

### Makro Vzdalenosti

Makro Vzdalenosti přiřazuje k hranám jejich délky, neboli minimální vzdálenosti od počátečního do koncového uzlu v případě, že bereme v potaz pouze dopravní komunikace. Tyto



	A	B	C
1	ID	ORP	Typ odpadu/zařízení
2	001	Nová Paka	SKO
3	002	Neratovice	SKO
4	003	Nepomuk	SKO
5	004	Náměšť nad Oslavou	SKO
6	005	Náchod	SKO
7	006	Most	SKO
8	007	Moravský Krumlov	SKO
9	008	Moravské Budějovice	SKO
10	009	Moravská Třebová	SKO

Obrázek 3.4: Tabulka vstupních dat pro Makro Hraný

vzdálenosti čerpá z matice vzdáleností z každého uzlu do každého uzlu, již lze vidět na obrázku 3.5.

	1	2	3	4	5	6	7	8
1		96505,92	224585,56	209315,34	63364,84	168804,60	241991,12	202159,68
2	96505,92		142168,49	204597,05	148579,51	95040,36	237272,83	197718,04
3	224585,56	142168,49		234386,47	266680,76	184689,89	267062,24	194874,88
4	209315,34	204597,05	234386,47		187692,40	274222,13	31258,71	40616,99
5	63364,84	148579,51	266680,76	187692,40		233840,42	220225,70	203342,04
6	168804,60	95040,36	184689,89	274222,13	233840,42		306897,91	267343,11
7	241991,12	237272,83	267062,24	31258,71	220225,70	306897,91		47378,53
8	202159,68	197718,04	194874,88	40616,99	203342,04	267343,11	47378,53	

Obrázek 3.5: Úplná matice vzdáleností jednotlivých ORP v ČR

Makro nejprve pomocí stejného algoritmu jako u makra Hraný spočítá počet vygenerovaných hran a uloží je do proměnné  $n$ . Následně u každé hrany rozdělí její název na dva řetězce, kde první řetězec je jméno počátečního uzlu a druhý jméno druhého uzlu. Tyto dva řetězce pak převede na čísla, neboli uloží je do proměnné typu Integer.

Hledání správné vzdálenosti probíhá tak, že pomocí cyklu While program postupně kontroluje první sloupec, dokud nenastane shoda s názvem počátečního uzlu, a pořadí toho řádku zapíše. Naprosto stejně to udělá i s názvem koncového uzlu, u nějž kontroluje shodu s čísly v prvním řádku a nakonec zapíše pořadí sloupce. Když zná umístění vzdálenosti, zkopíruje ho vedle dané hrany a pokračuje další hranou, až dosáhne předpočítaného počtu hran  $n$ . Ukázkou zápisu délek hran je možné pozorovat na obrázku 3.6.

## Makro Incidence

Toto makro slouží pro výpis matice incidence. Ta se většinou vypisuje jako dvourozměrná tabulka. V našem případě by vypisování tak obrovské tabulky postrádalo smysl, protože by byla velmi řídká (každý sloupec představující jednu hrany by měl pouze dvě hodnoty, 1 pro koncový uzel a -1 pro počáteční uzel a zbytek by vyplňovaly prázdné hodnoty). Tím pádem stačí, když tuto tabulku vypíšeme do tří sloupců. První sloupec bude obsahovat

Hrany	Vzdálenost [m]
001_4028	75 321,54
001_4031	198 146,22
001_4087	137 710,32
001_4105	257 585,20
001_4115	240 769,19
001_4157	155 406,51
001_4162	232 600,04
001_4198	280 080,64
001_4214	229 599,44

Obrázek 3.6: Úsek z výpisu hran a jejich délek

každou hranu dvakrát a druhý sloupec vyplní příslušné uzly a třetí sloupec číslice -1 a 1. Náhled lze vidět na obrázku 3.7.

Hrany	Uzly	A
103_4214	103	-1
103_4215	103	-1
103_4219	103	-1
103_4221	103	-1
103_4227	103	-1
104_4028	104	-1
104_4031	104	-1
104_4087	104	-1
104_4105	104	-1
104_4115	104	-1

Obrázek 3.7: Úsek z výpisu incidenční matice

Kód nejprve zkopíruje hrany dvakrát za sebou do prvního sloupce v listu pro incidenční matici. Následně pro první část hran zkopíruje z řetězce první část, tedy počáteční uzel a vloží ho do druhého sloupce. Pro druhou část hran vezme druhou část řetězce, tedy koncový uzel a vloží ho opět do druhého sloupce. Nakonec už stačí jen vypsát číslice -1 pro počáteční uzly a 1 pro ty koncové.

## Makro Vyhřevnosti

Toto makro slouží pro výpis výhřevností přiřazené jednotlivým hranám. Postupně se v rámci ujišťování neprázdnosti kontrolované buňky pomocí Do While algoritmu pro všechny řádky vypsáních hran provádí následující příkazy: Nejprve se pomocí příkazu If zjistí počet znaků vepsaného řetězce. V případě osmi vypsáních znaků je jasné, že počáteční uzel je producent SKO. V případě devíti znaků je třeba pomocí dalšího příkazu If zjistit počáteční číslici kódu producenta. Pro číslici 1 jde o OO, pro číslici 2 o TAP, pro číslici 3 o ZBV a pro číslici 4 o ZBN. V posledním kroku je třeba do buňky vpravo od kontrolované vepsat danou výhřevnost. Jako příklad vepsání výhřevnosti pro hrany s počátečním uzlem produkujícím SKO je vyobrazen příkaz z makra.

```
Sheets("Hrany").Cells(i, 4) = 7 + Rnd * 4
```

Tento příkaz do listu "Hrany" do buňky v řádku  $i$  a sloupci 4, tedy sloupci vedle vypsaných hran, zapíše hodnotu  $7 + 4x$ , kde  $x \in (0, 1)$ .

Vybranou část z e zapsaných dat lze vidět na obrázku 3.8.

Hrany	Výhřevnost [GJ/t]
4031_5077	5,07
4031_5087	14,09
4031_5089	8,63
4031_5105	14,59
4031_5111	14,82
4031_5115	9,18
4031_5127	12,87
4031_5130	5,84
4031_5135	11,70
4031_5157	9,14

Obrázek 3.8: Část vypsaných hodnot výhřevností přidělených k daným hranám

## Tlačítko Names

V sešitu Implementace.xlsm na listu Uzly (počáteční list) je kromě tlačítka "Generate" rovněž tlačítko "Names". První tlačítko je pojmenováno podle jeho hlavního úkonu generování různých obsáhlých dat. Tlačítko "Names" je pojmenováno podle toho, že dává určitým oblastem jejich jména. Toto tlačítko vzniklo pro případ rychlé změny počtu uzlů nebo jejich identifikace, aby se rychle změnily plochy, které do sebe načítá GAMS, a mohl tak rychle reagovat na změny. Makro, které je na tlačítko napojeno, je zobrazeno v následujícím textu.

Jednotlivá makra jsou pojmenována podle toho, jaké jméno dané oblasti makro přiděluje a potažmo také, jaký parametr se tímto úkonem načítá. (Makro pizi je makro načítající dva parametry: pi - uzly producentů a zi - uzly zpracovatelů.) Zde je příklad nejjednoduššího kódu z volaných maker, tedy načítání množiny  $i$ .

Nejprve makro deklaruje proměnnou typ Integer. Dále pomocí cyklu Do While kontroluje řádky prvního sloupce, tedy postupně všechny definované uzly, dokud nenarazí na prázdnou buňku. Dochází k tomu v listu "Uzly". U poslední buňky ještě dochází k připočtení 1 k  $i$ , což je překročení množství o 1, proto je třeba následně 1 odečíst. Nakonec přichází nejdůležitější moment vytvoření názvu pro oblast uzlů. Řádek začínající "This" tedy postupně říká: Pro tento sešit v sekci jména vytvoř nové a pojmenuj ho "i". Toto jméno nechť odkazuje na oblast buněk v na listu "Uzly" od buňky A2 až po buňku Ai.  $i$  musí být mimo uvozovky, protože je typu integer a ne string.

```

Sub Names ()
Call i
Call e
Call M
Call K
Call p
Call pizi
Call A
Call D
Call ik
Call C_pot
Call p_pot
Call poti
Call V
Call RV
Call MV
Call vi
End Sub

```

```

Sub i ()
Dim i As Integer
i = 2
Do While Sheets("Uzly").Cells(i, 1) <> ""
    i = i + 1
Loop
i = i - 1
ThisWorkbook.Names.Add Name:="i", RefersTo:="=Uzly!$a$2:$a$" & i
End Sub

```

### 3.3. Implementovaná data

Sekce implementovaná data se bude zabývat výčtem a popisem přesných dat, se kterými bude model počítat.

#### 3.3.1. Producenti a zpracovatelé

Jako jeden uzel úloha považuje buď producenta daného odpadu v daném ORP a nebo zpracovatele (zařízení na zpracování odpadu) v daném ORP. Následující shrnutí zobrazuje druhy odpadů a typy zařízení, které jsou v modelu zahrnuty. Nalevo jsou buď obecně používané a nebo čistě pro tuto práci zavedené zkratky odpadů a zařízení. Napravo jsou pak katalogová čísla odpadů a celé názvy typu zařízení. Jsou uvažována všechna katalogová čísla z předchozí sekce Případová studie, čili všechna katalogová čísla spalitelných odpadů. Zkratka ZBV je uměle vytvořená zkratka pro odpady vhodné pro výrobu paliv. Tyto odpady mají značně vyšší výhřevnost než odpady, pro které je vytvořena zkratka ZBN, neboli odpady, které nejsou vhodné pro výrobu paliv.

#### Typy produkovaných odpadů

SKO 200301

<b>OO</b>	200307
<b>TAP</b>	191210
<b>ZBV</b>	020103, 020104, 030101, 030105, 030301, 030307, 030308, 040101, 040108, 040209, 040221, 040222, 070213, 090108, 120105, 150101, 150102, 150103, 150105, 150109, 150203, 160119, 170203, 170302, 170604, 191201, 191204, 191207, 191208, 200101, 200110, 200111, 200139
<b>ZBN</b>	020107, 020304, 020601, 150106, 160199, 170201, 180104, 191212, 200138, 200302, 200303

### Typy zařízení

<b>EVO</b>	Zařízení pro energetické využití odpadu (spalovna)
<b>MOB</b>	Monoblok - Zařízení konstruované pro zpracování výhřevných frakcí ze zbytkových odpadů (plasty, papír, textil, ...). Tyto frakce zahrnuje ZBV.
<b>CEM</b>	Cementárna

Každé zařízení může zpracovávat jen určitý druh odpadu. Na následující tabulce 3.1 lze vidět rozdělení přijímaného odpadu do jednotlivých zařízení.

<b>SKO</b>	<b>OO</b>	<b>TAP</b>	<b>ZBV</b>	<b>ZBN</b>
EVO	EVO	EVO	EVO	EVO
	MOB	MOB	MOB	
		CEM	CEM	

Tabulka 3.1: Tabulka možného zpracování jednotlivých odpadů v jednotlivých zařízeních

V úloze Implementace je uvažováno 206 producentů SKO, 194 producentů OO, 26 producentů TAP, 206 producentů ZBV a 206 producentů ZBN. Existující ZEVO se vyskytují v Praze, Brně, Plzni a Liberci. Pro potenciálně nová zařízení je v modelu vybráno dalších 34 míst po ČR. Pro stavbu monobloků je vyhrazeno 10 různých ORP. Model současně zahrnuje 4 existující cementárny v Hranici, v Chrudimi (Holcim Práchevice), v Lovosicích (Lafarge) a ve Šlapanicích. Celkově se tedy počítá s 882 uzly, pomocí kterých bylo vygenerováno 36 728 hran, označujících všechny možnosti rozvozu odpadu ke zpracovateli. Jelikož je incidenční matice generována způsobem popsaným v sekci "Makro Incidence", je v této matici generováno 73 456 řádků, každý pro jednu hranu a její počáteční, nebo koncový uzel. Následující tabulka 3.2 zobrazuje množství vyprodukovaného odpadu a teoretickou maximální kapacitu zařízení v kilotunách a následně pak v procentuálním rozložení.

Největší producent SKO (349 426 tun za rok), OO (35 229 t/r) i ZBN (46 282 t/r) je Praha. TAP (64 726 t/r) se vyprodukuje nejvíce v Chrudimi a ZBV (22 223 t/r) v Rokycanech. Kapacita stávající spalovny (EVO) v Praze je 330 kt/r, v Brně 240 kt/r, v Liberci 96 kt/r a v Plzni 95 kt/r.

	SKO	OO	TAP	ZBV	ZBN	$\Sigma$
Produkce [kt]	2 735	379	185	249	278	3 826
Produkce [%]	71,49	9,91	4,83	6,49	7,27	100
	EVO	MOB	CEM			$\Sigma$
Kapacita [kt]	5 076	1 400	136			6 612
Kapacita [%]	76,77	21,17	2,05			100

Tabulka 3.2: Tabulka množství vyprodukovaného odpadu a teoretické maximální kapacity zařízení v kilo tunách a procentuálním rozložení

### 3.3.2. Náklady za zpracování a dopravu

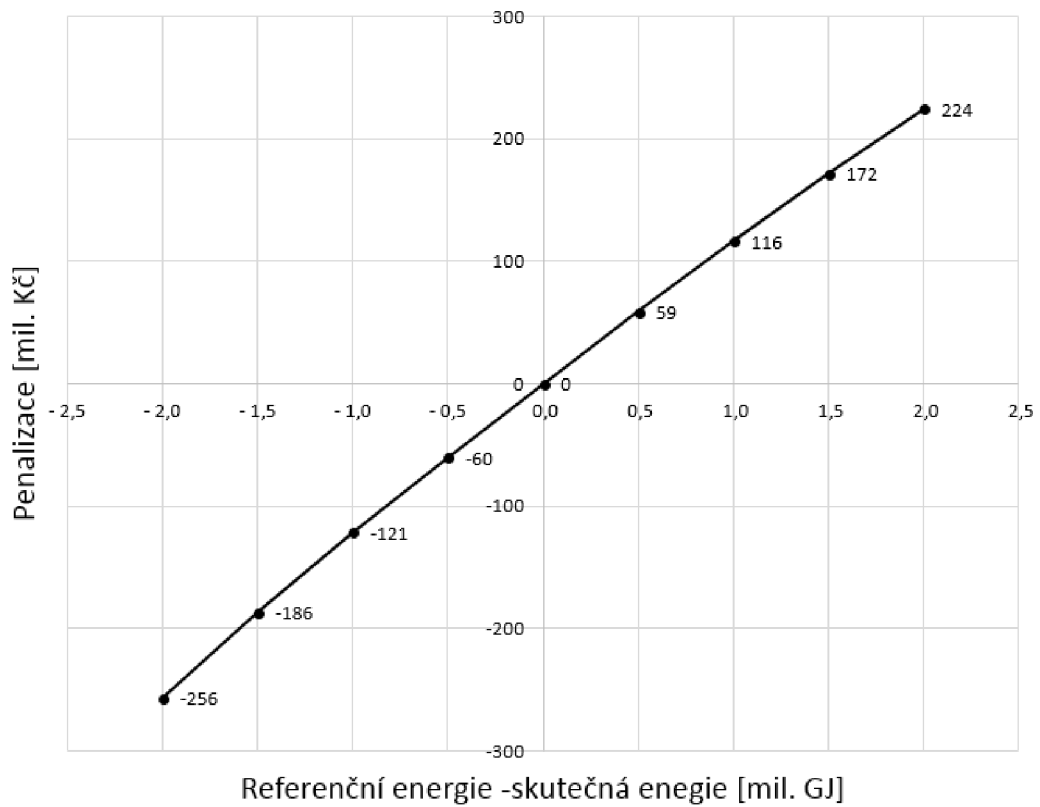
Konstantní cena za dopravu jedné tuny odpadu činí 4 Kč za jeden kilometr. Cena za zpracování jedné tuny v existujících zařízení EVO je v úloze konstantní 1500 Kč/t. Díky vzájemné symbióze odpadového hospodářství a cementáren je daná cena za zpracování v cementárnách stanovena na 0 Kč/t. Cena za zpracování tuny odpadu v potencionálních zařízeních EVO se může vyšplhat až na 5 388 Kč, zatímco pro monoblok pouze na 3 801 Kč. Náklady za zpracování veškerého odpadu v množství kapacity ZEVO dosahují až 793 138 100 Kč. Pro monobloky je tato cena rovna 455 220 300 Kč.

Výhřevnosti pro všechny druhy vyprodukovaného SO jsou stanoveny náhodně dle hodnot v následující tabulce 3.3.

Typ odpadu	výhřevnost [GJ/t]
SKO	(7 - 11)
OO	(10 - 14)
TAP	(8 - 25)
ZBV	(13 - 18)
ZBN	(4 - 8)

Tabulka 3.3: Předpis výhřevností daný druhů SO.

Pro cenu 150 Kč za jeden GJ energie v podobě tepla byla stanovena odpovídající penalizace za nedostatečnou výhřevnost zpracovávané směsi vzhledem k referenční výhřevnosti stanovené pro EVO 10 GJ/t a pro monobloky 16 GJ/t. Minimální výhřevnost pro EVO jsou stanoveny konstantně na 7 GJ/t, pro monobloky na 15 GJ/t a pro cementárny na 20 GJ/t. Tato závislost byla převedena na závislost penalizace na nedostatečném množství vyprodukované energie vynásobením daným množstvím odpadů. Toto množství bylo stanoveno na 500 000 tun a výhřevnosti od -4 až po 4, jelikož v úloze může penalizace nabývat i záporných hodnot, tedy dochází k odměňování za vyšší výhřevnost. Závislost penalizace zařízení na rozdílu referenční energie a skutečné vyprodukované energie v zařízení je zobrazena na následujícím grafu 3.9.

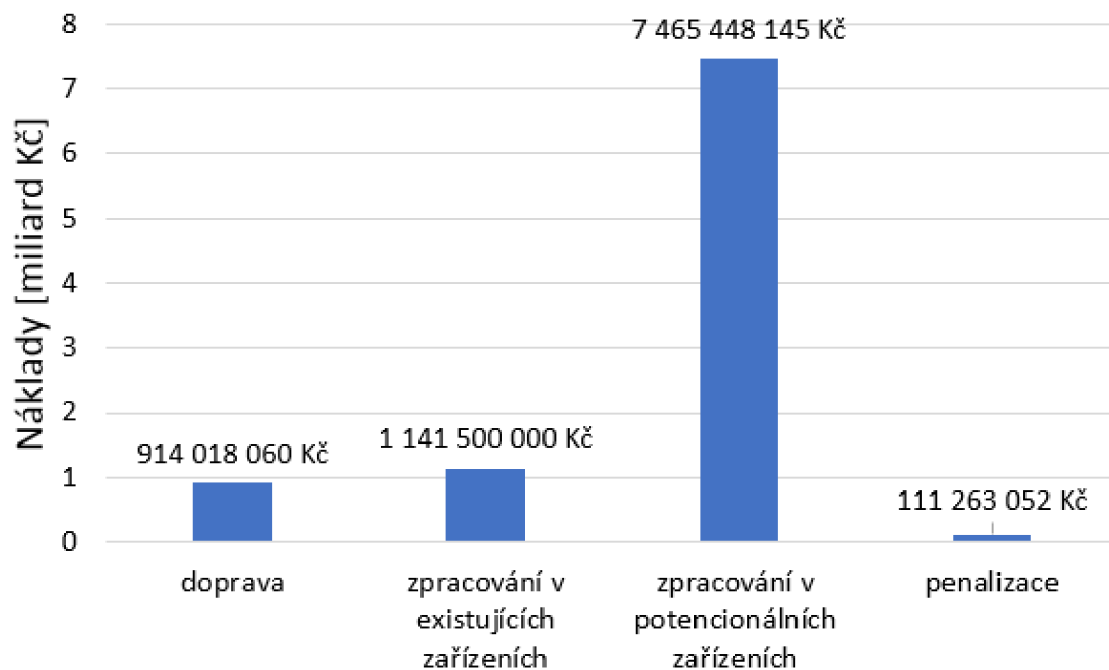


Obrázek 3.9: Závislost penalizace zařízení na rozdílu referenční energie a skutečné vyprodukované energie v zařízení

## 3.4. Řešení úlohy s implementovanými daty

### 3.4.1. Celkové náklady

Celkové náklady na zpracování veškerého SO, který není materiálově zpracováván, jsou 9 632 229 258 Kč. Graf v obrázku 3.10 ukazuje rozdělení těchto nákladů mezi dopravu, zpracování v existujících zařízeních, zpracování v potencionálních zařízeních a penalizaci zařízení.



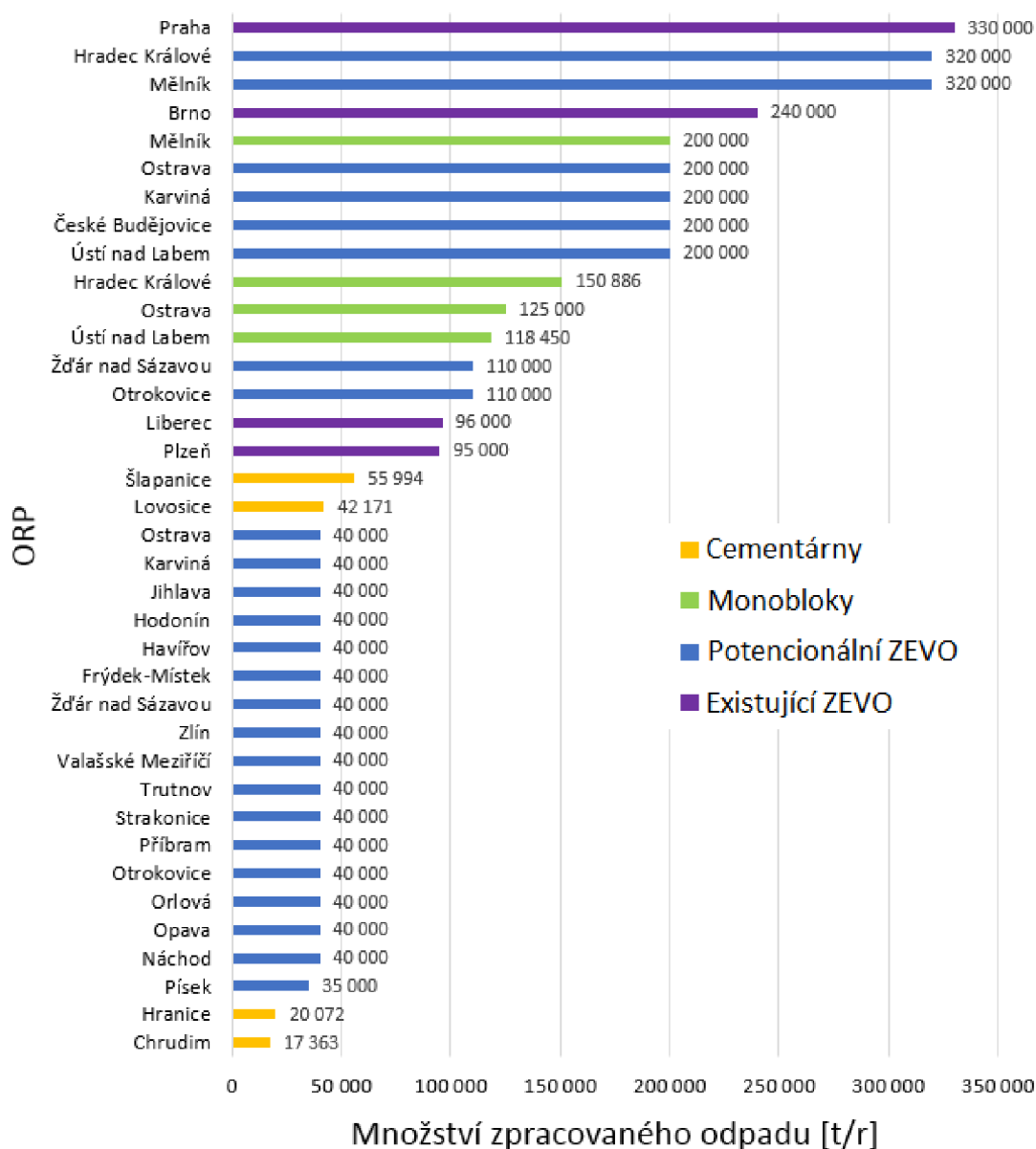
Obrázek 3.10: Rozdělení celkových nákladů mezi dopravu, zpracování v existujících zařízeních, zpracování v potencionálních zařízeních a penalizaci zařízení

Graf ukazuje výsledek modelu MILP s penalizací, který vychází z dat z ISOH v roce 2015. Na základě těchto dat vyhodnotil, že je třeba využít veškerých dostupných existujících zařízení a i přesto je vidět, že většinu odpadů je třeba zpracovat v nových zařízeních. Z toho, že penalizace je kladná, je možné odvodit, že se vážený průměr výhřevnosti zpracováváných směsí pohybuje pod referenčními výhřevnostmi.



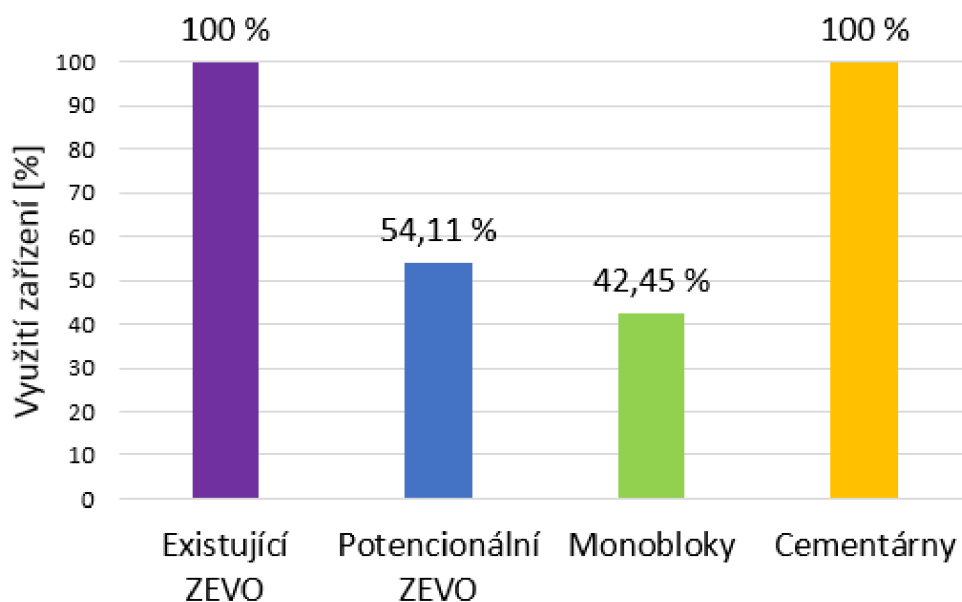
### 3.4.2. Zpracování odpadu

Veškeré zpracování v jednotlivých ORP je možné vidět na následujícím grafu na obrázku 3.11. Až na dvě výjimky ve Žďáru nad Sázavou a Otrokovicích je všech 23 zbylých ZEVO postaveno v nejvyšší možné kapacitě. Kromě již existujícího zařízení v Praze by v tomto případě byla dvě největší zařízení EVO v Hradci Králové a Mělníku. Ještě je nutno dodat, že jedno ORP může obsahovat dvě různá zařízení, jak je to například v Hradci Králové nebo v Mělníku a dokonce i dvě stejná zařízení, jak je tomu například pro dvě ZEVO v Ostravě nebo ve Žďáru nad Sázavou. V našem případě by byly postaveny 4 nové monobloky, maximální kapacity by ovšem nabýval jen monoblok v Mělníku. A konečně je v grafu na obrázku 3.11 vidět, že cementárny jsou samozřejmě využívány v plném rozsahu.



Obrázek 3.11: Celkový zpracovaný odpad za rok v jednotlivých zařízeních

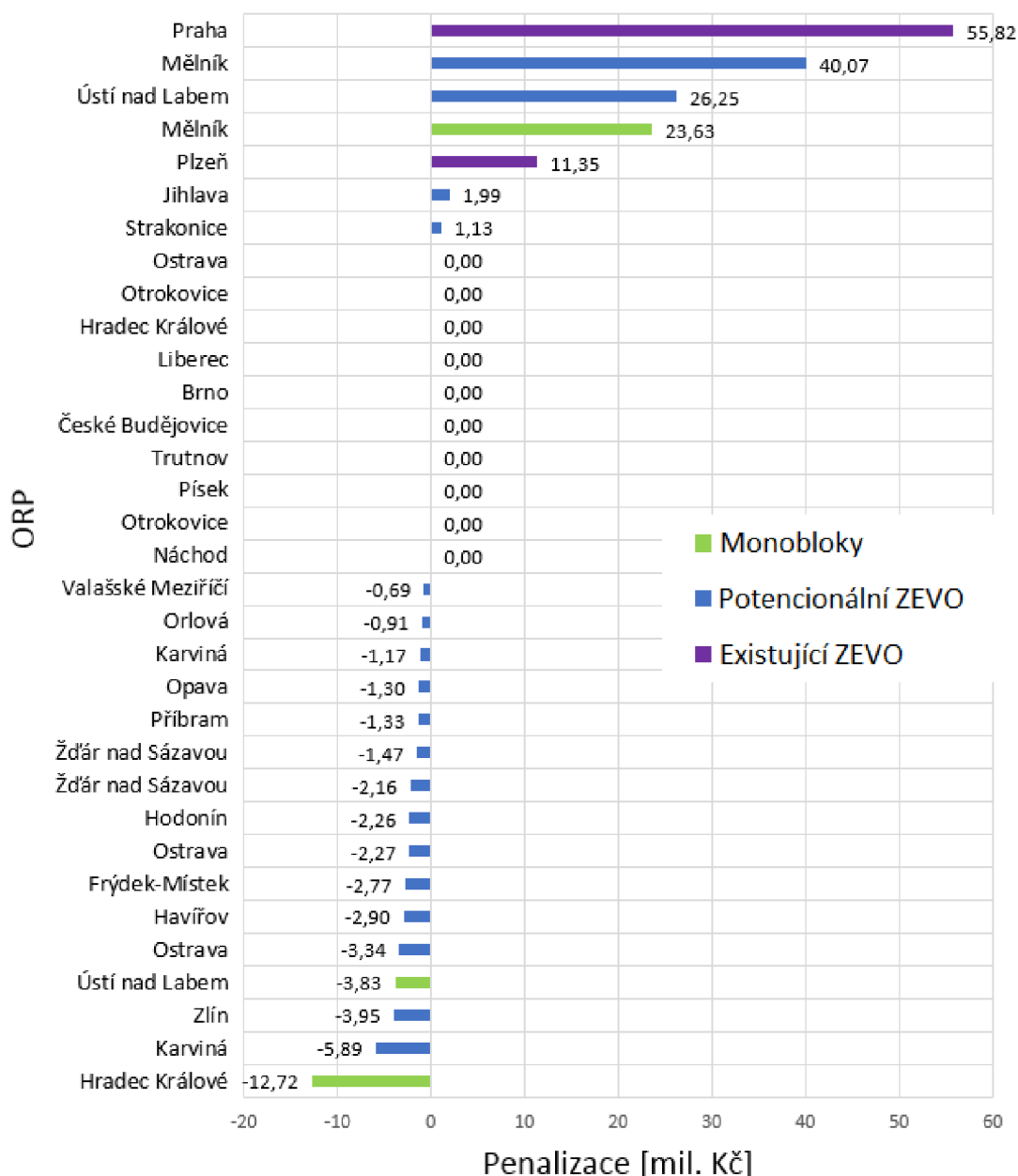
Na grafu v obrázku 3.12 lze vidět procentuální využití všech druhů zařízení. Jelikož je v modelu podmínka na využití celé kapacity nového zařízení, lze tento graf pro potenciační zařízení považovat za závislost celkové postavené kapacity ku předurčené maximální možné kapacitě. Všechna existující zařízení jsou využívána v plném rozsahu.



Obrázek 3.12: Procentuální zastoupení využitých kapacit potenciačních zařízení ku modelem poskytnutým maximálním kapacitám všech potenciačních zařízení

### 3.4.3. Penalizace nedostatečné výhřevnosti zařízení

Graf na obrázku 3.13 ukazuje výslednou penalizaci jednotlivých zařízení vzhledem k jejich plnění produkce energie v podobě tepla.



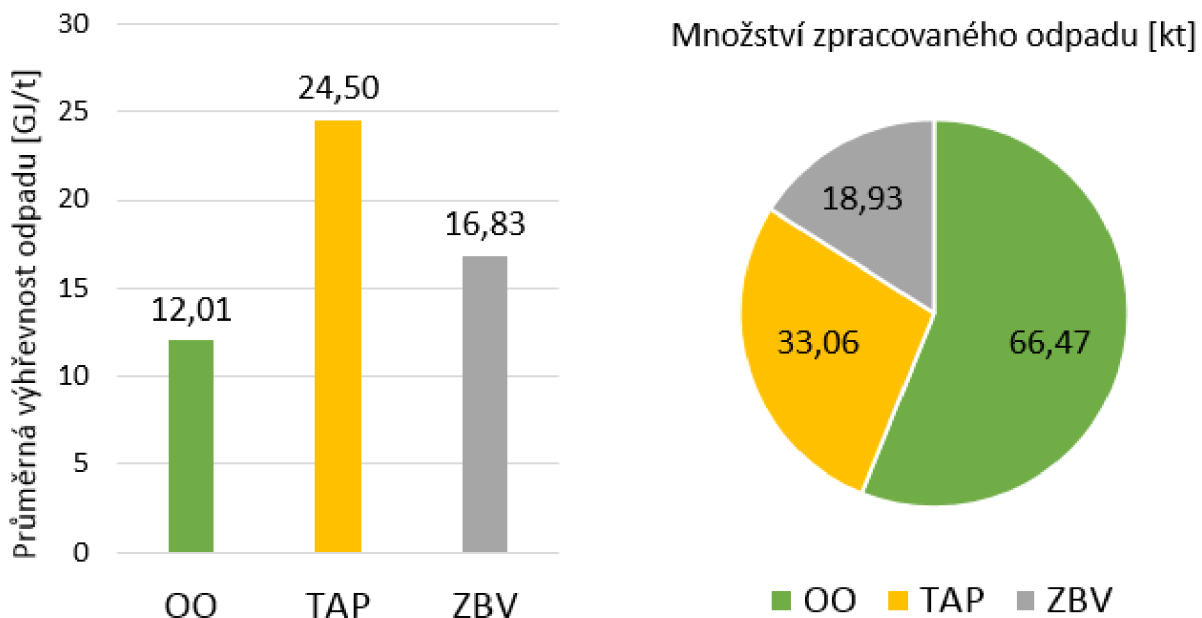
Obrázek 3.13: Penalizace jednotlivých zařízení v ORP

Model vyhodnotil ideální situaci tak, že se u většiny zařízení výhřevnost směsi příliš nelišila od referenční a tím pádem pro ně nevznikla ani příliš velká penalizace, ať už kladná nebo záporná. Tento efekt je přínosný v tom smyslu, že většina zařízení nebude mít problém s přílišnými provozními teplotami nebo s nutností dodávat do spalovacího roštu zemní plyn. Maximální odchylka výhřevnosti od referenční vznikla v existujícím zařízení v ORP Praha, kde převýšila tuto hodnotu o 0,945 GJ/t. Přesto, že v Praze díky tomu vznikla penalizace ve výši necelých 56 milionů Kč, je hodnota 10,945 GJ/t

z hlediska provozního stavu naprosto v pořádku. Stejně tak je tomu u druhého extrému, tedy v monobloku vzniklém Hradci Králové, kde je výhřevnost směsi pouze o 0,211 GJ/t nižší než referenční. Výhřevnost směsi v tomto zařízení je 15,799 GJ/t.

### 3.4.4. Zpracování odpadu v daném ORP

Pro ilustraci zpracování odpadu bylo vybráno zařízení monoblok v ORP Ústí nad Labem. Tomuto zařízení model přidělil kapacitu 118450 tun za rok, což je, v poměru k maximální možné navržené kapacitě 200000 tun za rok, 59 %. Při využití celé jeho kapacity a při průměrné výhřevnosti zpracovávané směsi 16,27 GJ/t byla pro toto zařízení vypočítána záporná penalizace 3,83 milionů Kč. Za těchto podmínek vyprodukuje zařízení ročně 1,93 milionů GJ. Průměrné výhřevnosti jednotlivých druhů přijatých odpadů a rozdělení množství zpracovaného odpadu lze vidět na obrázku 3.14.



Obrázek 3.14: Průměrná výhřevnost přijímaného odpadu a prožství přijímaných odpadů do ORP Ústí nad Labem

# Závěr

V práci byly nejprve zavedeny pojmy z teorie grafů jako orientovaný, neorientovaný graf, incidenční matice a bipartitní graf, na kterém byly definovány tokové a dopravní úlohy. Dále byly zavedeny některé základní pojmy z optimalizace, například smíšená celočíselná lineární optimalizace, která byla později použita pro výpočet modelů uvedených v kapitolách 2.3 a 2.4 a nakonec i pro modifikovaný model, do kterého byla implementována data z ISOH. Pro stochastické úlohy byl představen přístup Here-and-Now, který je užíván ve všech vyvinutých modelech.

Pro správnou funkci vyvinutých modelů v programu GAMS musel být představen použitý řešitel CPLEX, užívaný ve smíšeném celočíselném lineárním programování. Dále pak speciální proměnná SOS2, užívaná pro linearizaci cenových funkcí v potencionálních zařízeních a pro výpočet penalizace zařízení. A nakonec import ze souboru .xlsm a export do souboru .xls.

Dále byly v diplomové práci vyvinuty tři postupně se rozšiřující modely. První model byl zaveden na obecné síti a zabýval se pouze dopravní úlohou s podmínkou minimální výhřevnosti v některých zařízeních. Tento model byl řešen pomocí lineárního programování. Druhý model byl zaveden již na bipartitní síti a byl do něho implementován tzv. location problem, tedy problém určení ideálního umístění a ideální kapacity nových zařízení. Tento problém přináší i nutnost zahrnout do úlohy SOS2 proměnné na linearizaci nelineární závislosti ceny za zpracování na kapacitě zařízení, která je v tomto úseku i odůvodněna pomocí příjmů a výdajů zařízení. Tato úloha je řešena pomocí smíšeného celočíselného lineárního programování. Třetí úloha je stejná jako druhá s rozdílem rozšíření o penalizaci zařízení za nedostatečné plnění přidělené referenční vyprodukované energie za rok. Tato závislost je taktéž nelineární a proto se do úlohy přidává další SOS2 proměnná.

Poslední kapitola se zabývá užitím modelů pro situaci odpadového hospodářství v ČR. Nejprve uvádí případovou studii, kde objasňuje momentální nakládání se SO. Dále ukazuje, jak se v programu Excel pomocí maker zajišťuje rychlá modifikace dat. Nakonec jsou představena přesná vstupní data pro implementaci a jsou shrnuty výsledky implementace.

Tyto výsledky ukazují, že pro většinu SO, který není materiálově zpracováván, je třeba postavit nová zařízení. Celková cena za stavbu a zpracování v těchto nových ZEVO a monobloků bude celkově 7,47 miliardy Kč. Do grafu bylo vykresleno přesné množství zpracovaného materiálu v jednotlivých zařízeních dle ORP a v dalším grafu bylo znázorněno, kolik procent z kapacity jednotlivých druhů zařízení bylo využito. Celkem pro zpracování odpadu bylo navrženo 37 zařízení včetně již existujících. Z maximální navržené kapacity pro potencionální ZEVO bylo využito 54,11% a pro monobloky 42,45%. Nakonec byla vykreslena penalizace zařízení dle ORP a druhu zařízení. Maximální penalizace 55,82 milionů Kč dosáhla existující spalovna v Praze, ve které byla výhřevnost směsi o 0,945 GJ/t menší než referenční. Naopak v monobloku v Hradci Králové výhřevnost přesáhla referenční hodnotu o 0,211 GJ/t. Pro zařízení typu monoblok v ORP Ústí nad Labem je na konci práce poskytnuta drobnější analýza.

# Literatura

- [1] ŠOMPLÁK, R.; SZÁSZIOVÁ, L.; SMEJKALOVÁ, V.; NEVRLÝ, V.; MÁLEK, M.; PAVLAS, M. *Současné toky a zpracování spalitelných odpadů v ČR*. In Sborník konference. Praha: CEMC, 2017. s. 1-10. ISBN: 978-80-85990-30- 0.
- [2] DEMEL, Jiří. *Grafy a jejich aplikace*. Praha: Academia, 2002. ISBN 80-200-0990-6.
- [3] NASH, S. and A. SOFER. *Linear and nonlinear programming*. McGraw-Hill, 1995. ISBN 978-0070460652.
- [4] POPELA, Pavel. *Lineární programování v kostce*. Brno, 2000.
- [5] DANTZIG, George. *Linear Programming and Extensions*. Princeton University Press: Princeton, New Jersey, 1998. ISBN 9780691059136.
- [6] FDBR, 2000. *Acceptance testing of waste incineration plants with grate firing systems*. FDBR Fachverband Dampfkessel-, Behälter- und Rohrleitungsbau, 2000. Guideline edition 04/2000. 50 s.
- [7] JANOŠTÁK, František. *Aplikace optimalizace pro návrh subsystémů energetických celků*. Brno, 2014. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství. Vedoucí práce Ing. MARTIN PAVLAS, Ph.D.
- [8] An Introduction to GAMS. *General Algebraic Modeling System* [online]. Germany: GAMS Software [cit. 2018-04-15]. Dostupné z: <https://www.gams.com/products/introduction/>
- [9] GAMS Modelling Language Manuals. GAMS Company, 2015.
- [10] Specifying types of variables. *General Algebraic Modeling System* [online]. Germany: GAMS Software [cit. 2018-04-15]. Dostupné z: <https://www.gams.com/24.8/docs/userguides/mccarl/specifyingtypesofvariables.htm>
- [11] FERDAN, Tomáš a Radovan ŠOMPLÁK, Lenka ZAVÍRALOVÁ, Martin PAVLAS a Lukáš FRÝBA. A waste-to-energy project: A complex approach towards the assessment of investment risks. *Applied Thermal Engineering*. 2015, (89), 9.
- [12] R. Šomplák, T. Ferdan, M. Pavlas, P. Popela, *Waste-to-energy facility planning under uncertain circumstances*, *Appl. Therm. Eng.* 61 (2013) 106-114, <http://dx.doi.org/10.1016/j.applthermaleng.2013.04.003>.
- [13] Zákon č. 185/2001 Sb. *Zákon o odpadech a o změně některých dalších zákonů*. Sbírka zákonů České republiky, část šestá, § 39 a 40. Dostupný také z <http://www.mzp.cz/www/platnalegislativa.nsf>, [12. 2. 2017].
- [14] ZEVO Malešice. *Ceník energetického využití odpadu od 1.1.2016*, Dostupné z <http://www.psas.cz/psas/assets/File/cenikzevoofi20161.pdf>, [12. 2. 2017].
- [15] TERMIZO Liberec. *Seznam přijímaných odpadů*. Dostupné z <http://tmz.mvv.cz/wp-content/uploads/2014/09/Seznamodpadu.pdf>, [12. 2. 2017].
- [16] SAKO Brno. *Seznam přijímaných odpadů*. Dostupné z <http://www.sako.cz/stranka/cz/259/seznam-prijimanych-odpadu>, [12. 2. 2017].

# Seznam zkratek

ISOH	Informační systém odpadového hospodářství
VISOH	Veřejný informační systému odpadového hospodářství
SO	Spalitelný odpad
SKO	Směsný komunální odpad
OO	Objemný odpad
TAP	Tuhé alternativní palivo
ZBV	Ostatní typy odpadů vhodné pro výrobu paliv
ZBN	Ostatní typy odpadů nevhodné pro výrobu paliv
EVO	Energetické využití odpadů
ZEVO	Zařízení pro energetické využití odpadů
MOB	Monoblok - zařízení pro zpracování odpadů
CEM	Cementárna
LP	Lineární programování
MILP	Smíšené celočíselné lineární programování
MINLP	Smíšené celočíselné nelineární programování
SOS2	Speciálně uspořádaná množina typu 2
ME	Microsoft Excel

# Seznam příloh

CD s vloženými projekty v programu GAMS a daty v programu Microsoft Excel

model_K.gpr	Projekt vyvíjených modelů
maly_model.gms	Vyvinutý model pro základní úlohu
maly_model_bipart_sos.gms	Vyvinutý model pro úlohu MILP
mm_deset.gms	Vyvinutý model pro úlohu MILP rozšířenou o penalizační funkci
maly_model.xls	Sešit programu Excel se vstupními daty modelu pro základní úlohu
vypis_maly_model.xls	Sešit programu Excel s výstupními daty modelu pro základní úlohu
maly_model_bipart_SOS.xls	Sešit programu Excel se vstupními i výstupními daty modelu pro úlohu MILP
mm_deset.xls	Sešit programu Excel se vstupními i výstupními daty modelu pro úlohu MILP rozšířenou o penalizační funkci
Implementace.gpr	Projekt modelu modifikovaného pro načítání dat z ISOH
Implementace.gms	Model modifikovaný pro načítání dat z ISOH
Implementace.xlsm	Sešit programu Excel se vstupními daty modelu modifikovaného pro načítání dat z ISOH
Implementace.xls	Sešit programu Excel s výstupními daty modelu modifikovaného pro načítání dat z ISOH