



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MATEMATIKY

INSTITUTE OF MATHEMATICS

**POKROČILÉ MODELY STOCHASTICKÉHO
PROGRAMOVÁNÍ V ENERGETICE**

ADVANCED STOCHASTIC PROGRAMMING MODELS IN POWER ENGINEERING

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. ONDŘEJ PAVELKA

VEDOUCÍ PRÁCE

SUPERVISOR

RNDr. PAVEL POPELA, Ph.D.

BRNO 2022

Zadání diplomové práce

Ústav: Ústav matematiky
Student: **Bc. Ondřej Pavelka**
Studijní program: Aplikované vědy v inženýrství
Studijní obor: Matematické inženýrství
Vedoucí práce: **RNDr. Pavel Popela, Ph.D.**
Akademický rok: 2021/22

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:

Pokročilé modely stochastického programování v energetice

Stručná charakteristika problematiky úkolu:

Student se seznámí s problematikou řešení složitých optimalizačních úloh v oblasti energetiky. Zaměří se na pokročilé modely stochastického programování a metody jejich řešení. Důraz bude kladen na návrh pokročilých modelů, modifikace řešících algoritmů a jejich efektivní softwarové implementace. Student se z pohledu řešitelnosti uvažovaných úloh bude také zabývat vlastnostmi optimalizačních modelů a jejich vhodnými transformacemi. Vybrané modely bude implementovat pomocí vhodně zvoleného softwarového nástroje a provede testovací výpočty. Předpokládá se zapojení studenta do řešení optimalizačních úloh vybraných výzkumných projektů (COMSI) s využitím reálných dat a za účasti konzultantů specialistů (prof. Štětina, doc. Charvát).

Cíle diplomové práce:

1. Studium problematiky vybrané třídy složitých optimalizačních úloh v energetice.
2. Studium pokročilých modelů stochastického programování.
3. Studium vlastností a transformací vybraných modelů matematického programování pro zvolenou třídu úloh.
4. Výběr algoritmů a jejich případné modifikace pro studované modely.
5. Efektivní softwarová implementace modelů a algoritmů.
6. Aplikace na vybraný reálný problém a testovací výpočty.

Seznam doporučené literatury:

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

KALL, P., WALLACE, S. W. Stochastic Programming. New York: John Wiley & Sons, 1993. ISBN 978-0471951582.

WALLACE, S. W., KING, A. Modeling with Stochastic Programming. Springer Verlag, 2012. ISBN 978-0-387-87816-4.

RUSZCZYNSKI, A. et al. Handbooks in Operations Research and Management Science. vol. 10: Stochastic Programming. Amsterdam: Elsevier, 2003. ISBN 978-0-444-50854-6.

BAZARAA, M. S., SHERALI, H. D., SHETTY, C. M. Nonlinear Programming: Theory and Algorithms. 2nd ed. New York: John Wiley & Sons, 1993. ISBN 0471599735.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2021/22

V Brně, dne

L. S.

doc. Mgr. Petr Vašík, Ph.D.
ředitel ústavu

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

Abstrakt

Tato diplomová práce se zabývá stochastickou optimalizací v oblasti energetiky. V první části práce je popsán dále využitý matematický aparát, konkrétně matematické, lineární, nelineární, celočíselné a stochastické programování. Druhá část práce pojednává o teplárně, ve které je teplo vytvářeno pomocí kotlů na plyn a biomasu. Cílem práce je vytvořit model pro plánování provozu těchto kotlů a jejich rozšíření. Model vychází z dvoustupňové úlohy stochastického programování se scénářovým přístupem. Model je následně řešený pomocí softwaru GAMS. V práci je rovněž provedena analýza citlivosti modelu a popsány návrhy k zlepšení.

Summary

This diploma thesis applies stochastic optimization in the field of the power engineering. In the thesis first part, the needed mathematical theory is described, specifically mathematical, linear, nonlinear, integer and stochastic programming. The second part deals with the heating plant, in which heat is generated by gas and biomass boilers. The aim of this thesis is to design a model for the schedule planning of these boilers. The model is based on two stage stochastic programming with scenario approach. Then the model is solved by GAMS software. In the final part of the text, the focus is on the model sensitivity analysis and suggestions for future improvement.

Klíčová slova

optimalizace, matematické programování, lineární programování, nelineární programování, celočíselné programování, stochastické programování, dvoustupňová úloha stochastického programování, model, modelování, plánování, teplo, výroba tepla, energie, GAMS, energetika

Keywords

optimization, mathematical programming, linear programming, nonlinear programming, integer programming, stochastic programming, two stage stochastic programming, model, modeling, planning, heat, heat production, energy, GAMS, power engineering

PAVELKA, O. *Pokročilé modely stochastického programování v energetice*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2022. 57 s. Vedoucí RNDr. Pavel Popela, Ph.D.

Prohlašuji, že jsem diplomovou práci Pokročilé modely stochastického programování v energetice vypracoval samostatně pod vedením RNDr. Pavel Popela, Ph.D. s použitím materiálů uvedených v seznamu literatury.

Bc. Ondřej Pavelka

Tímto bych rád poděkoval svému vedoucímu RNDr. Pavlu Popelovi, Ph.D. za odborné vedení, množství cenných rad a připomínek při zpracování této diplomové práce. Naše diskuze nad problematikou byly vždy velmi přínosné a obohacující. Dále bych rád poděkoval prof. Ing. Josefu Štětinovi za uvedení do technické části problematiky.

Bc. Ondřej Pavelka

Obsah

1	Úvod	2
2	Optimalizace	3
2.1	Matematické programování	3
2.2	Lineární optimalizace	4
2.3	Nelineární optimalizace	7
2.4	Celočíselná optimalizace	8
2.4.1	Metoda větví a mezí	9
2.5	Stochastická optimalizace	11
2.5.1	Dvoustupňová úloha stochastické optimalizace	12
2.5.2	Scénářový přístup	14
3	Model	16
3.1	Schéma modelu	16
3.2	Poptávané teplo	16
3.2.1	Teplo na vytápění	16
3.2.2	Teplo na ohřev vody	21
3.3	Dodávané teplo	22
3.4	Softwarová implementace	23
3.5	Řešení	31
3.6	Analýza citlivosti	34
3.7	Návrhy na vylepšení	39
4	Závěr	43
5	Literatura	44
6	Seznam použitých zkratk a symbolů	47
7	Seznam příloh	51
8	Přílohy	52

1. Úvod

Optimalizace (neboli matematická optimalizace, matematické programování) je oblast matematiky, která si klade za cíl nalézt nejvhodnější řešení zadaného problému. Jedná se o poměrně starou disciplínu [1]. Již řecký matematik Eukleidés žijící kolem roku 300 před naším letopočtem zkoumal, jaká je nejkratší vzdálenost mezi bodem a přímkou nebo jaký největší obsah může mít obdélník při pevně zvoleném obvodu. V 17. století se německý matematik a astronom Johannes Kepler zabýval otázkou, jaké rozměry má mít sud na víno, aby měl co největší objem při zadaném množství dřeva k jeho výrobě. Významný rozvoj optimalizace přišel s příchodem výpočetní techniky, která umožnila za krátký čas vyřešit rozsáhlé úlohy, jež předtím zabraly několik dní.

Stochastické programování se zabývá optimalizací úloh, do kterých vstupuje náhodná veličina, jejíž hodnota není známá v době řešení úlohy. S tímto typem úloh se setkáváme v každodenním životě. Pokud vytváříme plán výroby elektřiny, tak v době plánování neznáme přesnou hodnotu budoucí poptávky, která v tomto případě představuje náhodnou veličinu. Podobná situace je při plánování výroby tepla. Budoucí poptávka je závislá na venkovní teplotě, která není předem známá, lze ji pouze odhadnout na základě historických dat. V případě, že výroba je nižší než aktuální poptávka, hrozí výpadek dodávky elektřiny (resp. tepla) a v opačné situaci, kdy je výroba vyšší než aktuální poptávka, vznikají ztráty, neboť elektřina i teplo jsou obtížně skladovatelné.

Tato práce si klade za cíl studium problematiky stochastického programování, tvorbu modelů stochastického programování v oblasti energetiky a její následnou softwarovou implementaci. Práce je rozdělena do dvou částí. První část zahrnuje teoretický úvod, na kterém je založeno modelování reálných úloh. Teoretická část obsahuje popis matematického, lineárního, nelineárního a stochastického programování. Ke každému typu jsou uvedeny způsoby řešení včetně vhodných algoritmů. Pro plné pochopení se od čtenáře očekává základní znalost matematické analýzy, algebry a statistiky. Význam jednotlivých symbolů a zkratk je uveden v kapitole 6.

Druhá část se zabývá tvorbou modelu popisující výrobu tepla v teplárně. Teplo je zde vyráběno pomocí kotlů na biomasu a plyn. Cílem úlohy je vytvořit plán provozu těchto kotlů tak, aby byla splněna poptávka a dosáhlo se maximálního zisku. V další části je model transformován a je provedena analýza, jak se změní plán při změně vstupních nákladů. Poslední část se zabývá návrhem, jakou investici v teplárně provést, aby se v budoucnu dosáhlo vyššího zisku.

Nutno poznamenat, že modely čerpají data odpovídající podzimu roku 2021. V době publikování této práce (květen 2022) je situace odlišná. Cena biomasy a zemního plynu se podstatně zvýšila a situace na trhu se mění každým dnem. Viz historické události začátku roku 2022. Model lze ovšem aplikovat i na současná data.

Modely z druhé části jsou zpracovány pomocí softwaru GAMS a MATLAB. Použité kódy jsou k nalezení v digitální příloze. Kódy v programu GAMS lze spustit samostatně, pro komunikaci mezi softwarem GAMS a MATLAB je nutné provést nastavení přístupových cest. Detailní popis nastavení je uveden v seznamu příloh.

2. Optimalizace

V této kapitole uvedeme základní definice a tvrzení, které se týkají optimalizace. *Optimalizace* je matematická disciplína studující a popisující *optimalizační úlohy*. Optimalizační úloha je nějaký reálný problém, který má z podstaty více možných přípustných řešení. Naším cílem je ze všech možných přípustných řešení najít to nejvhodnější, které nazýváme optimum. Kritérium, podle kterého určíme nejvhodnější řešení, je závislé na povaze úlohy. Řešíme-li například úlohu plánování distribuce tepla, snažíme se minimalizovat ztráty. Pokud se ale zabýváme finanční situací, je našim cílem maximalizovat zisk. Hlavními prameny pro tuto kapitolu jsou [2], [3], [4], [5], [6] a [7]. Rozšiřující informace čtenář nalezne v pramenech [8], [9] a [10].

2.1. Matematické programování

Definice 2.1.1 (Úloha matematického programování (MP)). Necht $S \subset \mathbb{R}^n$ a $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Úlohou matematického programování rozumíme nalezení

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

za podmínky $\mathbf{x} \in S$.

Funkce $f(\mathbf{x})$ se nazývá *účelová funkce* a množina S označuje *množinu přípustných řešení*. Bod, ve kterém nabývá svého minima, značíme \mathbf{x}_{\min} . Běžně se užívá zkrácený zápis

$$\min_{\mathbf{x}} \{f(\mathbf{x}) \mid \mathbf{x} \in S\}.$$

Jako první je potřeba odpovědět na otázku, za jakých podmínek má úloha matematického programování zaručenou existenci řešení. Na to nám odpoví Weierstrassova věta.

Věta 2.1.2 (Weierstrassova věta). *Necht $S \subset \mathbb{R}^n$ je neprázdná kompaktní¹ množina a funkce $f : S \rightarrow \mathbb{R}$ je spojitá na S , pak funkce $f(\mathbf{x})$ na množině S nabývá svého minima.*

Důkaz výše uvedené věty lze nalézt např. v [3].

V některých aplikačních úlohách se nepožaduje minimalizace účelové funkce, nýbrž její maximalizace. Maximalizační úlohu lze snadno převést na minimalizační.

Poznámka 2.1.3. Mezi minimalizací a maximalizací platí následující vztahy:

$$\begin{aligned} \max_{\mathbf{x}} \{f(\mathbf{x}) \mid \mathbf{x} \in S\} &= -\min_{\mathbf{x}} \{-f(\mathbf{x}) \mid \mathbf{x} \in S\}, \\ \operatorname{argmax}_{\mathbf{x}} \{f(\mathbf{x}) \mid \mathbf{x} \in S\} &= \operatorname{argmin}_{\mathbf{x}} \{-f(\mathbf{x}) \mid \mathbf{x} \in S\}, \end{aligned}$$

přičemž argmin (resp. argmax) značí množinu všech bodů, ve kterých funkce nabývá svého minima (resp. maxima).

Definice 2.1.4 (Konvexní množina). Množinu $S \subset \mathbb{R}^n$ nazveme konvexní právě tehdy, když $\forall \mathbf{x}_1, \mathbf{x}_2 \in S, \forall \lambda \in [0; 1]$ platí $\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2 \in S$.

¹Uzavřená a omezená.

2.2. LINEÁRNÍ OPTIMALIZACE

Definice 2.1.5 (Konvexní a konkávní funkce). Necht $S \subset \mathbb{R}^n$ je neprázdná konvexní množina a funkce $f : S \rightarrow \mathbb{R}$. Funkci nazveme *konvexní* na množině S právě tehdy, když

$$\forall \mathbf{x}_1, \mathbf{x}_2 \in S, \forall \lambda \in (0, 1) : f(\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1 - \lambda) f(\mathbf{x}_2).$$

Funkci nazveme *konkávní* právě tehdy, když $-f(\mathbf{x})$ je konvexní.

Věta 2.1.6. Mějme neprázdnou konvexní množinu $S \subset \mathbb{R}^n$ a konvexní funkci $f : S \rightarrow \mathbb{R}$, pak každé lokální minimum je i globálním minimem neboli

$$\bar{\mathbf{x}} \in \underset{\mathbf{x}}{\operatorname{arglocmin}}\{f(\mathbf{x}) \mid \mathbf{x} \in S\} \Rightarrow \bar{\mathbf{x}} \in \underset{\mathbf{x}}{\operatorname{argglobmin}}\{f(\mathbf{x}) \mid \mathbf{x} \in S\}.$$

Důkaz. Necht $\bar{\mathbf{x}}$ je lokální minimum, tzn.

$$\bar{\mathbf{x}} \in S \text{ a } \exists O(\bar{\mathbf{x}}) \text{ takové, že } \forall \mathbf{x} \in O(\bar{\mathbf{x}}) \text{ platí } f(\mathbf{x}) \geq f(\bar{\mathbf{x}}).$$

Předpokládejme, že $\exists \mathbf{x}^* \in S$ takové, že

$$f(\mathbf{x}^*) < f(\bar{\mathbf{x}}).$$

Z konvexnosti množiny S plyne

$$\forall \lambda \in [0, 1] : \lambda \bar{\mathbf{x}} + (1 - \lambda) \mathbf{x}^* \in S.$$

Z konvexnosti funkce f plyne

$$\begin{aligned} \forall \lambda \in (0, 1) : f(\lambda \bar{\mathbf{x}} + (1 - \lambda) \mathbf{x}^*) &\leq \lambda f(\bar{\mathbf{x}}) + (1 - \lambda) f(\mathbf{x}^*) \\ &< \lambda f(\bar{\mathbf{x}}) + (1 - \lambda) f(\bar{\mathbf{x}}) \\ &= f(\bar{\mathbf{x}}). \end{aligned}$$

Pokud ale $\lambda \rightarrow 1$, tak $\lambda \bar{\mathbf{x}} + (1 - \lambda) \mathbf{x}^* \rightarrow \bar{\mathbf{x}}$ a ostrá nerovnost je ve sporu s předpokladem o lokálním minimu. \square

2.2. Lineární optimalizace

V této podkapitole představíme lineární programování, které je nejjednodušším typem matematického programování. Definice, věty a důkazy jsou převzaty z [2] a [3].

Definice 2.2.1 (Úloha lineárního programování (LP)). Úlohou lineárního programování ve standardním tvaru je úloha

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{x} \\ \text{za podmínek} \quad & \mathbf{A} \mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{2.1}$$

kde $\mathbf{c} \in \mathbb{R}^n$ a $\mathbf{b} \in \mathbb{R}^m$ jsou sloupcové vektory a matice $\mathbf{A} \in \mathbb{R}^{m \times n}$.

Prvky vektoru \mathbf{c} nazveme *cenové koeficienty*, rovnice $\sum_{j=1}^n a_{ij}x_j = b_i$ představuje i -té omezení (i -tou podmínku). Bod $\mathbf{x} \in \mathbb{R}^n$ splňující všechna omezení nazýváme *přípustné řešení*.

Ve formulaci úlohy požadujeme, aby všechna omezení byla ve tvaru rovnic. V reálných problémech se často setkáváme s omezením ve tvaru nerovnic – např. nejvyšší povolený výkon stroje. Nerovnici lze snadno upravit na rovnici přidáním dodatečné nezáporné proměnné.

$$\begin{aligned} \sum_{j=1}^n a_{ij}x_j \geq b_i &\Leftrightarrow \sum_{j=1}^n a_{ij}x_j - x_{n+1} = b_i \quad , x_{n+1} \geq 0, \\ \sum_{j=1}^n a_{ij}x_j \leq b_i &\Leftrightarrow \sum_{j=1}^n a_{ij}x_j + x_{n+1} = b_i \quad , x_{n+1} \geq 0. \end{aligned}$$

V případě, kdy některá proměnná nabývá libovolných reálných hodnot, provedeme následující úpravu. Proměnnou x_j nahradíme v celé úloze výrazem $x_j^+ - x_j^-$, přičemž požadujeme, aby $x_j^+ \geq 0$ a $x_j^- \geq 0$.

Lemma 2.2.2. a) Účelová funkce úlohy (2.1) je konvexní.
b) Množina přípustných řešení úlohy (2.1) je konvexní množina.

Z předchozího lemmatu plyne, že každé lokální minimum úlohy (2.1) je i globálním minimem.

Definice 2.2.3 (Krajní bod). Necht $S \subset \mathbb{R}^n$ je neprázdná konvexní množina. Bod $\mathbf{x} \in S$ nazveme *krajním bodem* množiny S právě tehdy, když

$$\forall \mathbf{x}_1, \mathbf{x}_2 \in S, \forall \lambda \in (0; 1) \text{ platí } \mathbf{x} = \lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2 \Rightarrow \mathbf{x} = \mathbf{x}_1 = \mathbf{x}_2.$$

Věta 2.2.4. [3, Theorem 83 s. 25] Necht $S = \{\mathbf{x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ a matice $\mathbf{A} \in \mathbb{R}^{m \times n}$ má plnou hodnost. Pak \mathbf{x} je krajním bodem množiny $S \Leftrightarrow$ v matici \mathbf{A} můžeme uspořádat sloupce (a obdobně příslušné řádky vektoru \mathbf{x}) tak, aby $\mathbf{A} = (\mathbf{B}, \mathbf{N})$, $\mathbf{B} \in \mathbb{R}^{m \times m}$, $\mathbf{N} \in \mathbb{R}^{m \times (n-m)}$ a dále platilo $\mathbf{Ax} = \mathbf{Bx}_B + \mathbf{Nx}_N = \mathbf{b}$, přičemž

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{pmatrix} = \begin{pmatrix} \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{0} \end{pmatrix},$$

kde $\mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$, $\mathbf{x}_B \in \mathbb{R}^m$, $\mathbf{x}_N \in \mathbb{R}^{n-m}$.

Matice \mathbf{B} nazveme *bází* popř. *bazickou maticí*, \mathbf{N} se nazývá *nebazická matice*. Prvky vektoru \mathbf{x}_B (resp. \mathbf{x}_N) nazveme *bazické* (resp. *nebazické*) neznámé.

Definice 2.2.5 (Směr). Necht $S \subset \mathbb{R}^n$ je neprázdná uzavřená konvexní množina. Vektor $\mathbf{d} \in \mathbb{R}^n$, $\mathbf{d} \neq \mathbf{0}$ nazveme *směrem množiny S* , jestliže

$$\forall \mathbf{x} \in S, \forall \lambda \geq 0 \text{ platí } \mathbf{x} + \lambda \mathbf{d} \in S.$$

Definice 2.2.6 (Krajní směr). Směr \mathbf{d} množiny $S \subset \mathbb{R}^n$ nazveme *krajním směrem* právě tehdy, když

$$\forall \mathbf{d}_1, \mathbf{d}_2 \text{ směry množiny } S, \forall \lambda_1, \lambda_2 \geq 0 \text{ platí } \mathbf{d} = \lambda_1 \mathbf{d}_1 + \lambda_2 \mathbf{d}_2 \Rightarrow \exists \alpha > 0 : \mathbf{d}_2 = \alpha \mathbf{d}_1.$$

2.2. LINEÁRNÍ OPTIMALIZACE

Věta 2.2.7. *Nechť $S = \{\mathbf{x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\} \neq \emptyset$, matice \mathbf{A} má plnou hodnost, pak množina S má alespoň jeden krajní bod.*

Věta 2.2.8 (Věta o reprezentaci). *Nechť $S = \{\mathbf{x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\} \neq \emptyset$, matice \mathbf{A} má plnou hodnost. Jsou-li $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$ všechny krajní body množiny S a $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_l$ všechny krajní směry množiny S , pak*

$$\mathbf{x} \in S \Leftrightarrow \exists \mu_j \geq 0, j = 1, \dots, l, \exists \lambda_j \geq 0, j = 1, \dots, k : \sum_{j=1}^k \lambda_j = 1 \wedge \mathbf{x} = \sum_{j=1}^k \lambda_j \mathbf{x}_j + \sum_{j=1}^l \mu_j \mathbf{d}_j.$$

Důkazy dvou výše uvedených vět jsou uvedeny v [2, Theorem 2.1, s. 77].

Věta 2.2.9 (Podmínka optimality LP). *Mějme úlohu lineárního programování ve standardním tvaru. Označme $S = \{\mathbf{x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ množinu přípustných řešení a $D = \operatorname{argmin}_{\mathbf{x}} \{\mathbf{c}^\top \mathbf{x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$. Dále předpokládejme, že matice \mathbf{A} má plnou hodnost, $S \neq \emptyset$, $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$ jsou všechny krajní body množiny S a $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_l$ všechny krajní směry množiny S , pak*

$$D \neq \emptyset \Leftrightarrow \forall j \in \{1, \dots, l\} : \mathbf{c}^\top \mathbf{d}_j \geq 0 \Leftrightarrow \exists i \in \{1, \dots, k\} : \mathbf{x}_i \in D.$$

Důkaz. Na základě Věty o reprezentaci 2.2.8 můžeme množinu D přepsat do tvaru

$$D = \operatorname{argmin}_{\mu, \lambda} \left\{ \sum_{j=1}^k \lambda_j (\mathbf{c}^\top \mathbf{x}_j) + \sum_{j=1}^l \mu_j (\mathbf{c}^\top \mathbf{d}_j) \mid \right. \\ \left. \mu_j \geq 0, j = 1, \dots, l, \lambda_j \geq 0, j = 1, \dots, k : \sum_{j=1}^k \lambda_j = 1 \right\}.$$

Pokud existuje $j \in \{1, \dots, l\}$ takové, že $\mathbf{c}^\top \mathbf{d}_j < 0$, tak volbou $\mu_j \rightarrow \infty$ (zbývající μ_j jsou nulové) se účelová funkce stává zdola neohrazenou. Když $\mathbf{c}^\top \mathbf{d}_j \geq 0$ pro všechna $j \in \{1, \dots, l\}$, tak abychom dosáhli minima, musíme položit $\mu_j = 0, \forall j \in \{1, \dots, l\}$. Množinu D pak můžeme upravit do tvaru

$$D = \operatorname{argmin}_{\lambda} \left\{ \sum_{j=1}^k \lambda_j (\mathbf{c}^\top \mathbf{x}_j) \mid \lambda_j \geq 0, j = 1, \dots, k : \sum_{j=1}^k \lambda_j = 1 \right\}.$$

Minima pak docílíme volbou $\lambda_i = 1$ (ostatní $\lambda_j = 0$) pro i splňující $\min\{\mathbf{c}^\top \mathbf{x}_j \mid j = 1, \dots, k\} = \mathbf{c}^\top \mathbf{x}_i$. \square

Nyní víme, že existuje-li řešení úlohy (2.1), musí se nacházet v některém krajním bodě. Postupně procházet všechny krajní body by bylo značně neefektivní, proto potřebujeme nějakou rozumnou prohledávací strategii. Tu nám poskytne *simplexová metoda*.

Simplexová metoda je založena na následující myšlence. Jsme v nějakém krajním bodě. Projdeme všechny hrany, které vycházejí z krajního bodu, a z nich vybereme hranu, po které se sníží hodnota účelové funkce. Pokud je vybraná hrana krajním směrem, účelová funkce je zdola neohrazená, v opačném případě se po hraně dostaneme do nového krajního bodu. Tento proces opakujeme, dokud z krajního bodu nevychází žádná hrana, po které účelová funkce klesá.

Simplexová metoda - algoritmus

1. Řešíme úlohu (2.1), matice \mathbf{A} má plnou hodnost. Mějme počáteční krajní bod. Označme J_B indexy sloupců matice \mathbf{A} , které tvoří matici \mathbf{B} , J_N indexy sloupců matice \mathbf{A} , které tvoří matici \mathbf{N} . Vytvoříme z \mathbf{c} odpovídající vektory \mathbf{c}_B a \mathbf{c}_N .
2. Vypočítáme $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} = \bar{\mathbf{b}}$, položíme $\mathbf{x}_N = \mathbf{0}$ a určíme $z = \mathbf{c}_B^T \mathbf{x}_B$.
3. Vypočítáme $\mathbf{w}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$, spočteme $\forall j \in J_N : z_j - c_j = \mathbf{w}^T \mathbf{a}_j - c_j$. Nalezneme $k = \operatorname{argmax}_{j \in J_N} \{z_j - c_j\}$. Je-li $z_k - c_k \leq 0$, algoritmus končí, optimální řešení je v aktuálním krajním bodě.
4. Vypočítáme $\mathbf{y}_k = \mathbf{B}^{-1} \mathbf{a}_k$. Je-li $\mathbf{y}_k \leq \mathbf{0}$ algoritmus končí, účelová funkce není zdola ohraničená.
5. Určíme $r \in \operatorname{argmin}_{i \in J_B} \left\{ \frac{\bar{b}_i}{y_{ik}} \mid y_{ik} > 0 \right\}$. Pak $J_B := (J_B \setminus \{r\}) \cup \{k\}$, $J_N := (J_N \setminus \{k\}) \cup \{r\}$. Aktualizujeme matici $\mathbf{B} := (\mathbf{a}_j)_{j \in J_B}$, $\mathbf{N} := (\mathbf{a}_j)_{j \in J_N}$, pokračujeme krokem 2.

Z předchozích úvah plyne, že při řešení úlohy (2.1) mohou nastat čtyři případy.

- Úloha nemá přípustné řešení.
- Účelová funkce není zdola ohraničena na množině přípustných řešení S .
- Účelová funkce je zdola ohraničena, minima nabývá pouze v jednom krajním bodě.
- Účelová funkce je zdola ohraničena, minima nabývá v nespočetně mnoha bodech, přičemž aspoň jeden z nich je krajní bod.

2.3. Nelineární optimalizace

Definice 2.3.1 (Nelineární programování (NLP)). Úlohou nelineárního programování nazveme úlohu

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{za podmínek} \quad & g_i(\mathbf{x}) \leq 0, i = 1, \dots, m, \\ & h_i(\mathbf{x}) = 0, i = 1, \dots, l, \\ & \mathbf{x} \in X \subset \mathbb{R}^n, \end{aligned}$$

kde $f, g_1, \dots, g_m, h_1, \dots, h_l$ jsou reálné funkce definované na \mathbb{R}^n . Množinu přípustných řešení označíme symbolem S . Dále budeme pro naše účely požadovat, aby množina X byla omezená a uzavřená.

Řešení úloh nelineárního programování je poměrně komplikované. V praxi se používá mnoho různých pokročilých algoritmů. Mnohé z těchto algoritmů jsou založeny na následující myšlence. Nacházíme se v bodě $\mathbf{x}_k \in X$. Prohlédáme jeho okolí patřící do množiny X a zvolíme směr \mathbf{d}_k , ve kterém účelová funkce klesá. V tomto směru uděláme krok délky λ_k tak, aby platilo

$$f(\mathbf{x}_k) > f(\mathbf{x}_{k+1}),$$

kde

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k \in S.$$

2.4. CELOČÍSELNÁ OPTIMALIZACE

Velikost λ_k volíme tak, aby pokles byl co největší. Tento postup opakujeme, dokud není žádný přípustný směr, ve kterém účelová funkce klesá. Tímto získáme posloupnost bodů $\mathbf{x}_k \in X, k = 0, \dots, K$, přičemž

$$f(\mathbf{x}_0) > f(\mathbf{x}_1) > \dots > f(\mathbf{x}_K).$$

Bod \mathbf{x}_K je pak lokální minimum. Zvolíme-li jiný počáteční bod \mathbf{x}_0 , může algoritmus skončit v jiném lokálním minimu. Jsou-li množina X a funkce f konvexní, pak každé lokální minimum je i globálním minimem. Další vlastnosti nelineárního programování lze odvodit z vlastností funkcí f, g_1, \dots, g_m a h_1, \dots, h_l , viz [3] a [4].

Podmínka pro ukončení se v praxi běžně zeslabuje. Jedním z kritérií pro zastavení je $\|\mathbf{x}_{k+N} - \mathbf{x}_k\| < \varepsilon$ pro pevně zvolené $N \in \mathbb{N}$ a $\varepsilon > 0$. Kritérium říká, že pokud jsme po N krocích příliš blízko od původního bodu \mathbf{x}_k , prohledávání končí a \mathbf{x}_{k+N} je aproximované lokální minimum.

2.4. Celočíselná optimalizace

Definice 2.4.1 (Úloha celočíselného programování). Úlohou celočíselného programování nazveme úlohu

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{za podmíněk} \quad & g_i(\mathbf{x}) \leq 0, i = 1, \dots, m, \\ & h_i(\mathbf{x}) = 0, i = 1, \dots, l, \\ & \mathbf{x} \in X \subset \mathbb{R}^n, \\ & x_i \in \mathbb{Z}, i \in J_0, \end{aligned} \tag{2.2}$$

kde $f, g_1, \dots, g_m, h_1, \dots, h_l$ jsou reálné funkce definovány na \mathbb{R}^n , $J_0 \subset J = \{1, \dots, n\}$, $J_0 \neq \emptyset$ je množina indexů. Jsou-li funkce f, g_i, h_i lineární, hovoříme o *lineární celočíselné optimalizaci*, v opačném případě se jedná o *nelineární celočíselnou optimalizaci*. Pokud $J_0 \neq J$, pak některé složky vektoru \mathbf{x} nemusí být celočíselné a hovoříme o *smíšené (mixed) celočíselné optimalizaci*. Množinu přípustných řešení označíme symbolem S .

Naivní přístup při řešení úloh celočíselného programování by byl „vyřešíme úlohu bez podmínek celočíselnosti a výsledek pak zaokrouhlíme“. Bohužel v mnoha úlohách je celočíselné optimální řešení dosti vzdálené tomu neceločíselnému, viz [5, Example 1.1, s. 4]. K řešení úloh celočíselného programování zavedeme jiný přístup. Podrobné informace jsou uvedeny v [5].

Definice 2.4.2. Úloha (RIP) $z^R = \min_{\mathbf{x}} \{t(\mathbf{x}) \mid \mathbf{x} \in T \subset \mathbb{R}^n\}$ se nazývá *relaxace* úlohy (IP) $z = \min_{\mathbf{x}} \{f(\mathbf{x}) \mid \mathbf{x} \in S \subset \mathbb{R}^n\}$, jestliže $S \subset T$ a $t(\mathbf{x}) \leq f(\mathbf{x})$ pro všechna $\mathbf{x} \in S$.

Věta 2.4.3. Je-li (RIP) relaxací (IP), pak $z^R \leq z$.

Důkaz. Necht \mathbf{x}_{\min} je optimální řešení (IP), pak $\mathbf{x}_{\min} \in S \subset T$ a $z = f(\mathbf{x}_{\min}) \geq t(\mathbf{x}_{\min})$. Jelikož $\mathbf{x}_{\min} \in T$, tak $z^R \leq t(\mathbf{x}_{\min}) \leq z$. \square

Důsledek 2.4.3.1. Mějme úlohu (IP) a její dvě relaxace $z_1^R = \min_{\mathbf{x}} \{t(\mathbf{x}) \mid \mathbf{x} \in T_1 \subset \mathbb{R}^n\}$ a $z_2^R = \min_{\mathbf{x}} \{t(\mathbf{x}) \mid \mathbf{x} \in T_2 \subset \mathbb{R}^n\}$, přičemž $T_1 \subset T_2$, pak $z_1^R \geq z_2^R$.

Věta 2.4.4. a) Nemá-li úloha (RIP) přípustné řešení, pak i původní úloha (IP) nemá přípustné řešení.
b) Necht \mathbf{x}_{\min} je optimální řešení (RIP). Jestliže $\mathbf{x}_{\min} \in S$ a $t(\mathbf{x}_{\min}) = f(\mathbf{x}_{\min})$, tak \mathbf{x}_{\min} je optimální řešení (IP).

Důkaz plyne přímo z definice (RIP). V další části představíme algoritmus k řešení úloh celočíselné optimalizace.

2.4.1. Metoda větví a mezí

Metoda větví a mezí je algoritmus prohledávající množinu přípustných řešení. Množina přípustných řešení se postupně dělí (větví), a tím se vytváří binární strom. Metoda větví a mezí má uplatnění nejen pro řešení lineárních i nelineárních celočíselných úloh, přičemž využívá dvou následujících vět.

Věta 2.4.5. Uvažujme úlohu $z = \min_{\mathbf{x}} \{f(\mathbf{x}) \mid \mathbf{x} \in S \subset \mathbb{R}^n\}$ a necht $\{S_k\}_{k=1}^K$ je rozklad množiny S a necht $z^k = \min_{\mathbf{x}} \{f(\mathbf{x}) \mid \mathbf{x} \in S_k\}$ pro $k = 1, \dots, K$, pak $z = \min\{z^k \mid k = 1, \dots, K\}$.

Věta 2.4.6. Necht $\{S_k\}_{k=1}^K$ je rozklad množiny S , $z^k = \min_{\mathbf{x}} \{f(\mathbf{x}) \mid \mathbf{x} \in S_k\}$ pro $k = 1, \dots, K$, \bar{z}^k je horní mez z^k , \underline{z}^k je dolní mez z^k ($\bar{z}^k \geq z^k \geq \underline{z}^k$), pak $\min\{\bar{z}^k \mid k = 1, \dots, K\} = \bar{z} \geq z \geq \underline{z} = \min\{\underline{z}^k \mid k = 1, \dots, K\}$.

Hlavní myšlenka algoritmu je založena na tom, že množinu přípustných řešení rozkládáme (větvíme) na menší množiny a pomocí relaxací určujeme meze. Toto opakujeme, dokud dalším rozkladem nemůžeme zlepšit optimální řešení.

Stručný popis algoritmu

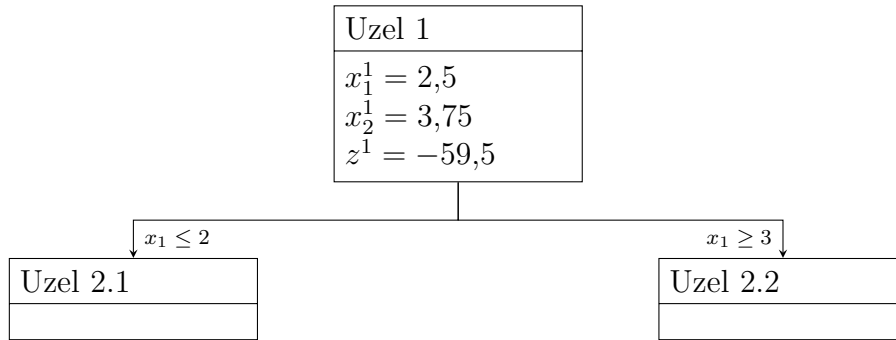
1. Máme zadanou úlohu $z = \min_{\mathbf{x}} \{f(\mathbf{x}) \mid \mathbf{x} \in S \subset \mathbb{R}^n\}$, viz (2.2). Provedeme relaxaci úlohy. Množinu S rozšíříme na množinu T vynecháním podmínek celočíselnosti. Dále definujeme proměnou pro optimální řešení \mathbf{x}^* a horní odhad $\bar{z} = +\infty$. Množina T přísluší prvnímu uzlu binárního stromu.
2. Vybereme neprohledaný uzel binárního stromu. Řešíme úlohu $z^k = \min_{\mathbf{x}} \{f(\mathbf{x}) \mid \mathbf{x} \in T^k\}$, kde T^k je množina příslušného uzlu. Optimální řešení značíme \mathbf{x}_{\min} . Uzel označíme za prohledaný.
3. Nemá-li úloha přípustné řešení, pokračujeme krokem 7.
4. Je-li $z^k \geq \bar{z}$, pokračujeme krokem 7.
5. Je-li $\mathbf{x}_{\min} \in S$, přepočteme $\bar{z} := z^k$ a $\mathbf{x}^* := \mathbf{x}_{\min}$, pokračujeme krokem 7.
6. Z vektoru \mathbf{x}_{\min} vybereme i -tou složku (značíme x_{\min}^i) takovou, že $x_{\min}^i \notin \mathbb{Z}$, $i \in J_0$. Prohledávanému uzlu vytvoříme dva navazující uzly, do kterých vložíme množinu $T^k \cap \{\mathbf{x} \mid x^i \leq \lfloor x_{\min}^i \rfloor\}$ a $T^k \cap \{\mathbf{x} \mid x^i \geq \lceil x_{\min}^i \rceil\}$.
7. Jsou-li všechny uzly stromu prohledány, algoritmus je ukončen a $z = \bar{z}$, v opačném případě pokračujeme krokem 2.

2.4. CELOČÍSELNÁ OPTIMALIZACE

Příklad: Řešme úlohu

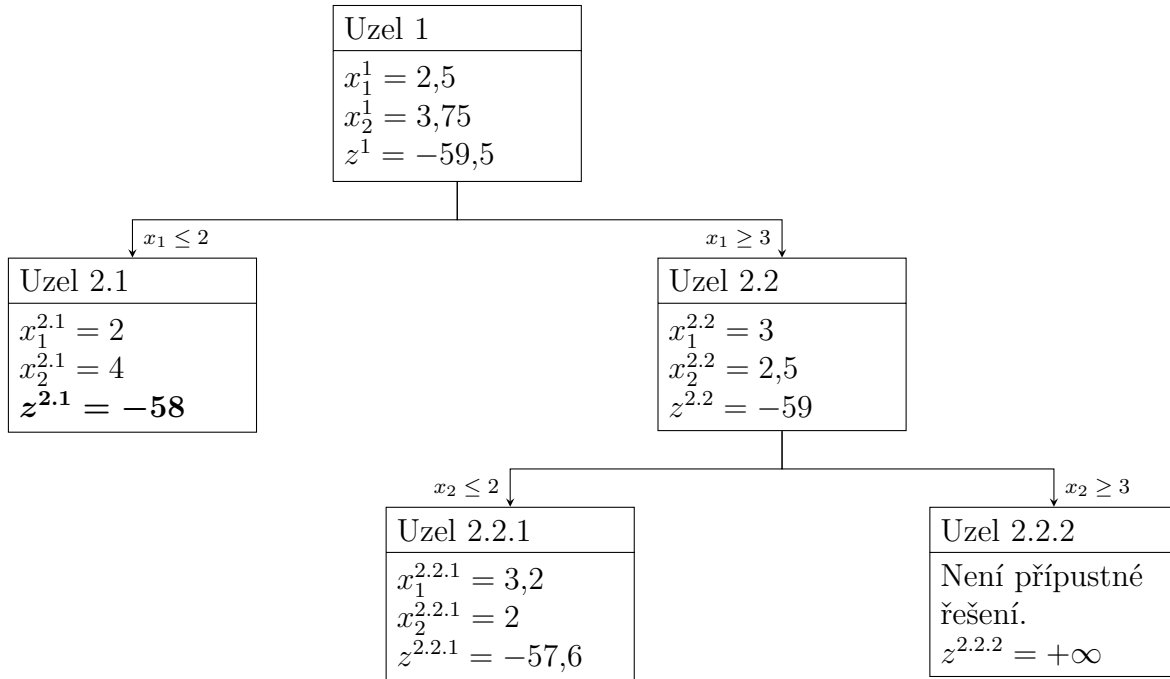
$$\begin{aligned} \min \quad & -13x_1 - 8x_2 \\ \text{za podmínek} \quad & x_1 + 2x_2 \leq 10, \\ & 5x_1 + 2x_2 \leq 20, \\ & x_1 \geq 0, x_2 \geq 0, x_1, x_2 \in \mathbb{Z}. \end{aligned} \tag{2.3}$$

Provedeme relaxaci, čímž získáme úlohu lineárního programování. Vyřešením získáme $\mathbf{x}_{\min}^1 = (2,5; 3,75)^\top$, $z^1 = -59,5$. První složka vektoru \mathbf{x}_{\min}^1 nesplňuje podmínku celočíselnosti, vytvoříme tedy dva nové uzly, do kterých přidáme podmínku $x_1 \leq 2$, resp. $x_1 \geq 3$, viz větvení na obrázku 2.1.



Obrázek 2.1: Binární strom úlohy (2.3) po první iteraci. (Vlastní tvorba.)

V dalším kroku zvolíme uzel 2.1. Oproti uzlu 1 máme navíc podmínku $x_1 \leq 2$. Řešením získáme $\mathbf{x}_{\min}^{2.1} = (2; 4)^\top$, $z^{2.1} = -58$. Bod $\mathbf{x}_{\min}^{2.1} \in S$, přepočteme $\bar{z} = -58$ a $\mathbf{x}^* = (2; 4)^\top$ (uzel dále nevětvíme, viz krok 5 algoritmu). Analogicky pokračujeme uzlem 2.2. $\mathbf{x}_{\min}^{2.2} = (3; 2,5)^\top$, $z^{2.2} = -59$. Přidáme uzly 2.2.1 a 2.2.2 s podmínkou $x_2 \leq 2$, resp. $x_2 \geq 3$.



Obrázek 2.2: Binární strom úlohy (2.3). (Vlastní tvorba.)

Uzel 2.2.1 má $z^{2.2.1} = -57,6 > \bar{z} = -58$ (uzel dále nevětvíme, viz krok 4 algoritmu), Uzel 2.2.2 nemá přípustné řešení (uzel dále nevětvíme, viz krok 3 algoritmu). Všechny uzly jsou prohledány, algoritmus je ukončen, $z = -58, \mathbf{x}^* = (2; 4)^\top$.

Velikost vyhledávacího stromu je závislá na $|J_0|$. Pro velké $|J_0|$ je téměř nemožné prohledat celý stavový prostor, takže algoritmus musíme modifikovat.

S využitím předchozích vět víme, že $z^k \leq \min\{z^{k.1}; z^{k.2}\}$. Díky této vlastnosti můžeme získat dolní odhad \underline{z} . Pak víme, že $\bar{z} \geq z \geq \underline{z}$. Rozdíl $\bar{z} - \underline{z}$ nazveme absolutní přesnost a podíl $|(\bar{z} - \underline{z})/\bar{z}|$ relativní přesnost. Do kódu pak přidáme podmínku, že při dosažení zvolené relativní přesnosti se prohledávání ukončí, viz [11].

2.5. Stochastická optimalizace

V poslední části této kapitoly se budeme zabývat stochastickou optimalizací ([6], [7]). Některé parametry v reálných optimalizačních úlohách nejsou předem známy, jejich hodnoty jsou náhodné. Jako příklad můžeme uvést obchodování s cennými papíry. Cena vždy závisí na aktuální nabídce a poptávce a její budoucí vývoj není předem znám. Dalším příkladem je plánování výroby tepla a elektřiny. Odebírané množství není předem známé a skladování je komplikované.

Tento typ úloh budeme řešit pomocí *stochastické optimalizace*, přičemž neznámé parametry budeme reprezentovat pomocí *náhodných veličin*. Mějme pravděpodobnostní prostor (Ω, \mathcal{A}, P) , kde Ω je základní prostor skládající se z výsledků ω . \mathcal{A} značí jevové pole [12] a P je nezáporná, normovaná, σ -aditivní funkce, která každému jevu přiřadí číslo z intervalu $[0, 1]$, viz [12]. Zobrazení $\xi : \Omega \rightarrow \mathbb{R}$ nazveme náhodnou veličinou vzhledem k \mathcal{A} , pokud pro všechna $x \in \mathbb{R}$ platí $\{\omega \mid \omega \in \Omega, \xi(\omega) \leq x\} \in \mathcal{A}$. Bližší podrobnosti jsou uvedeny v [12].

Úloha stochastického lineárního programování je úloha odvozená od lineárního programování, ve které jsou některé parametry náhodné veličiny se známým rozdělením pravděpodobnosti.

Úlohy s kompenzací jsou úlohy, ve kterých jsou některá rozhodnutí uskutečněná až po realizaci náhodné veličiny. Rozhodnutí v nich můžeme rozdělit do dvou skupin:

- Rozhodnutí uskutečněná před pozorováním realizací náhodných veličin. Tato rozhodnutí nazýváme *rozhodnutí prvního stupně* tedy typu here-and-now (HN, „právě teď“) a doba, ve které se tato rozhodnutí uskuteční, se nazývá *první stupeň*.
- Rozhodnutí, která se uskuteční až po realizaci náhodných veličin, se nazývají *rozhodnutí druhého stupně* tedy typu wait-and-see (WS, „počkej a uvidíš“). Odpovídající období se nazývá *druhý stupeň*.

Rozhodnutí uskutečněná v prvním stupni jsou reprezentovaná vektorem \mathbf{x} , zatímco druhý stupeň reprezentujeme vektorem \mathbf{y} , resp. $\mathbf{y}(\omega, \mathbf{x})$ chceme-li zdůraznit, že rozhodnutí druhého stupně jsou závislá na náhodné veličině a na rozhodnutí prvního stupně. Posloupnost událostí a rozhodnutí můžeme shrnout

$$\mathbf{x} \rightarrow \xi(\omega) \rightarrow \mathbf{y}(\omega, \mathbf{x}).$$

V případě obchodování s cennými papíry první stupeň odpovídá prvotnímu nákupu a druhý stupeň zahrnuje rozhodnutí, jak s nakoupenými papíry naložit po uplynutí určité

2.5. STOCHASTICKÁ OPTIMALIZACE

doby - prodat, přikoupit apod. Poznamenejme, že počet stupňů může být větší než dva, terminologie je pak analogická [6].

2.5.1. Dvoustupňová úloha stochastické optimalizace

Definice 2.5.1 (Dvoustupňová úloha stochastického lineární programování). Dvoustupňová úloha stochastického lineárního programování s konstantní kompenzací je formulována jako

$$\min_{\mathbf{x}} \left\{ \mathbf{c}^\top \mathbf{x} + E_{\boldsymbol{\xi}} \left[\left\{ \min_{\mathbf{y}} \mathbf{q}(\omega)^\top \mathbf{y}(\omega) \mid \mathbf{T}(\omega) \mathbf{x} + \mathbf{W} \mathbf{y}(\omega) = \mathbf{h}(\omega), \mathbf{y}(\omega) \geq \mathbf{0} \right\} \right] \mid \mathbf{A} \mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0} \right\}, \quad (2.4)$$

kde $\mathbf{c} \in \mathbb{R}^{n_1}$ je známý vektor, $\mathbf{b} \in \mathbb{R}^{m_1}$ je známý vektor, $\mathbf{A} \in \mathbb{R}^{n_1 \times m_1}$ a $\mathbf{W} \in \mathbb{R}^{m_2 \times n_2}$ jsou známe matice. Matici \mathbf{W} nazýváme *kompenzační matici* a předpokládáme, že je konstantní.

Pro každé $\omega \in \Omega$, $\mathbf{T}(\omega) \in \mathbb{R}^{m_2 \times n_1}$, $\mathbf{q}(\omega) \in \mathbb{R}^{n_2}$, $\mathbf{h}(\omega) \in \mathbb{R}^{m_2}$. Pokud všechny náhodné složky vložíme do jednoho vektoru, získáme vektor

$$\boldsymbol{\xi}^\top(\omega) = (\mathbf{q}(\omega)^\top, \mathbf{h}(\omega)^\top, \mathbf{T}_{1.}(\omega), \dots, \mathbf{T}_{m_2.}(\omega))$$

s $N = n_2 + m_2 + m_2 \cdot n_1$ složkami, přičemž $\mathbf{T}_{i.}(\omega)$ značí i -tý řádek *technologické matice* $\mathbf{T}(\omega)$. Rozhodnutí prvního stupně je reprezentováno pomocí vektoru $\mathbf{x} \in \mathbb{R}^{n_1}$. Jedná se o *neanticipativní* rozhodnutí, neboť je nezávislé na $\boldsymbol{\xi}$. Rozhodnutí druhého stupně $\mathbf{y}(\omega) \in \mathbb{R}^{n_2}$ je již závislé na $\boldsymbol{\xi}$.

Účelová funkce se skládá ze součtu deterministického členu $\mathbf{c}^\top \mathbf{x}$ a kompenzace, viz $E_{\boldsymbol{\xi}}[\cdot]$ v (2.4), kde $E_{\boldsymbol{\xi}}$ značí střední hodnotu vzhledem k $\boldsymbol{\xi}$. Necht $\Xi \subset \mathbb{R}^N$ je nosič $\boldsymbol{\xi}$, tj. nejmenší uzavřená podmnožina v \mathbb{R}^N splňující $P\{\boldsymbol{\xi} \in \Xi\} = 1$. Poznamenejme, že výpočet druhého členu je mnohem náročnější, protože pro každé ω musíme řešit lineární optimalizační úlohu. Úlohu (2.4) můžeme přepsat do tvaru

$$\begin{aligned} & \min_{\mathbf{x}} \mathbf{c}^\top \mathbf{x} + Q(\mathbf{x}) \\ & \text{za podmíněk } \mathbf{A} \mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned} \quad (2.5)$$

kde

$$Q(\mathbf{x}) = E_{\boldsymbol{\xi}} Q(\mathbf{x}, \boldsymbol{\xi}(\omega))$$

a

$$\begin{aligned} Q(\mathbf{x}, \boldsymbol{\xi}(\omega)) &= \min_{\mathbf{y}} \mathbf{q}(\omega)^\top \mathbf{y} \\ & \text{za podmíněk } \mathbf{W} \mathbf{y} = \mathbf{h}(\omega) - \mathbf{T}(\omega) \mathbf{x}, \\ & \mathbf{y} \geq \mathbf{0}. \end{aligned}$$

Úloha (2.4) ve tvaru (2.5) představuje základní tvar dvoustupňového stochastického programování. V některých úlohách můžeme požadovat, aby některé proměnné nabývaly

pouze celočíselných hodnot. Podobně lze požadovat, aby omezení či účelová funkce byly nelineární. Zobecněním získáme úlohu

$$\begin{aligned} & \min f^1(\mathbf{x}) + Q(\mathbf{x}) \\ & \text{za podmíněk } g_i^1(\mathbf{x}) \leq 0, i = 1, \dots, m_1, \\ & \quad h_i^1(\mathbf{x}) = 0, i = 1, \dots, l_1, \\ & \quad x_i \in \mathbb{Z}, i \in J_1, \\ & \quad \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

kde

$$Q(\mathbf{x}) = E_{\xi}[Q(\mathbf{x}, \xi(\omega))]$$

a

$$\begin{aligned} Q(\mathbf{x}, \xi(\omega)) &= \min f^2(\mathbf{y}, \xi(\omega)) \\ & \text{za podmíněk } g_i^2(\mathbf{x}, \mathbf{y}, \xi(\omega)) \leq 0, i = 1, \dots, m_2, \\ & \quad h_i^2(\mathbf{x}, \mathbf{y}, \xi(\omega)) = 0, i = 1, \dots, l_2, \\ & \quad y_i \in \mathbb{Z}, i \in J_2, \\ & \quad \mathbf{y} \geq \mathbf{0}, \end{aligned}$$

kde $f^1, g_i^1, i = 1, \dots, m_1, h_i^1, i = 1, \dots, l_1, f^2, g_i^2, i = 1, \dots, m_2, h_i^2, i = 1, \dots, l_2$ jsou reálné funkce, $J_1 \subset \{1, \dots, n_1\}$ a $J_2 \subset \{1, \dots, n_2\}$ množiny indexů.

Nyní se důkladněji zaměříme na dvoustupňovou úlohu lineárního stochastického programování. Budeme zkoumat, jak vypadá množina přípustných řešení a podmínky optimality. Ačkoliv jsme předpokládali, že matice \mathbf{W} je konstantní, tak nyní připustíme, že $\mathbf{W}(\omega)$. Pro pevně zvolené \mathbf{x} a ω má druhý stupeň tvar

$$\begin{aligned} Q(\mathbf{x}, \xi(\omega)) &= \min \mathbf{q}(\omega)^T \mathbf{y} \\ & \text{za podmíněk } \mathbf{W}(\omega) \mathbf{y} = \mathbf{h}(\omega) - \mathbf{T}(\omega) \mathbf{x}, \\ & \quad \mathbf{y} \geq \mathbf{0}. \end{aligned}$$

Je-li úloha zdola neomezená, položme $Q(\mathbf{x}, \xi(\omega)) = -\infty$, nemá-li přípustné řešení, tak $Q(\mathbf{x}, \xi(\omega)) = +\infty$. Uvažujme nyní situaci, že ξ je konečný diskretní náhodný vektor (Ξ je konečná, nebo spočetná množina). Druhý stupeň určený vztahem

$$Q(\mathbf{x}) = E_{\xi} Q(\mathbf{x}, \xi(\omega))$$

je pak vážená suma hodnot $Q(\mathbf{x}, \xi(\omega))$ přes všechny přípustné realizace ξ . Aby byla definice korektní, definujeme $+\infty + (-\infty) = +\infty$. Označme

$$\begin{aligned} K_1 &= \{\mathbf{x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}, \\ K_2 &= \{\mathbf{x} \mid Q(\mathbf{x}) < \infty\}, \\ K_2^P &= \{\mathbf{x} \mid \forall \xi \in \Xi, \exists \mathbf{y} \geq \mathbf{0} : \mathbf{Wy} = \mathbf{h} - \mathbf{T}\mathbf{x}\}. \end{aligned}$$

Vektor \mathbf{x} patří do množiny K_2^P pouze tehdy, když pro všechny přípustné hodnoty ξ existuje přípustné řešení \mathbf{y} druhého stupně.

2.5. STOCHASTICKÁ OPTIMALIZACE

Věta 2.5.2. [6, Theorem 1, s. 86]

- a) Pro každé $\xi(\omega)$ je množina přípustných řešení uzavřený konvexní mnohostěn, proto K_2^P je uzavřená konvexní množina.
b) Je-li Ξ konečná, tak K_2^P je mnohostěn shodující se s K_2 .

Věta 2.5.3. [6, Proposition 2, s. 87] Má-li ξ konečné druhé momenty, pak

$$P(\omega \mid Q(\mathbf{x}, \xi(\omega)) < \infty) = 1 \Rightarrow Q(\mathbf{x}) < \infty.$$

Věta 2.5.4. [6, Theorem 3, s. 88] Mějme úlohu lineárního stochastického programování s konstantní maticí \mathbf{W} a ξ má konečné druhé momenty, pak množiny K_2 a K_2^P se shodují.

Z poslední věty plyne, že v situaci, kdy matice \mathbf{W} je konstantní a ξ nabývá konečně mnoha hodnot, pak $K_2 = K_2^P$.

Věta 2.5.5. [6, Theorem 4, s. 88] Mějme úlohu lineárního stochastického programování s konstantní maticí \mathbf{W} a ξ má konečné druhé momenty, pak

- a) K_2 je uzavřená konvexní množina,
b) je-li \mathbf{T} konstantní, pak K_2 je mnohostěn.

Nyní se zaměříme na funkce $Q(\mathbf{x}, \xi(\omega))$ a $Q(\mathbf{x})$.

Věta 2.5.6. [6, Theorem 5, s. 89] Mějme úlohu lineárního stochastického programování s konstantní maticí \mathbf{W} a $Q(\mathbf{x}, \xi(\omega))$ je zdola ohraničena, pak $Q(\mathbf{x}, \xi(\omega))$ je

- a) po částech lineární konvexní funkce v (\mathbf{h}, \mathbf{T}) ,
b) po částech lineární konkávní funkce v \mathbf{q} ,
c) po částech lineární konvexní funkce v \mathbf{x} , $\forall \mathbf{x} \in K = K_1 \cap K_2$.

Věta 2.5.7. [6, Theorem 6 s. 90] Mějme úlohu lineárního stochastického programování s konstantní maticí \mathbf{W} a nechť ξ má konečné druhé momenty, pak

- a) $Q(\mathbf{x})$ je konvexní funkce a je konečná na K_2 ,
b) nabývá-li ξ konečně mnoha hodnot, $Q(\mathbf{x})$ je po částech lineární funkcí.

Věta 2.5.8. [6, Theorem 8 s. 94] Nechť má ξ konečné druhé momenty a množina $K = K_1 \cap K_2$ je omezená, pak úloha (2.4) má optimální řešení.

Výše uvedené věty jsou velmi důležité, neboť dávají do souvislosti stochastické programování s předchozími kapitolami.

2.5.2. Scénářový přístup

Uvažujme, že náhodný vektor ξ má konečné diskrétní rozdělení, tj. $|\Xi| = S < \infty$. Označme $\xi^s, s = 1, \dots, S$ hodnoty, kterých může vektor ξ nabývat. Tyto hodnoty budeme nazývat scénáře. Dále označme $p_s = P(\xi = \xi^s) \geq 0$, přičemž $\sum_{s=1}^S p_s = 1$. Druhý stupeň pak můžeme vyjádřit ve tvaru

$$E_{\xi} Q(\mathbf{x}, \xi(\omega)) = \sum_{s=1}^S p_s Q(\mathbf{x}, \xi^s).$$

Dále označme realizace $\mathbf{q}_s = \mathbf{q}(\xi^s)$, $\mathbf{T}_s = \mathbf{T}(\xi^s)$, $\mathbf{h}_s = \mathbf{h}(\xi^s)$ a rozhodnutí druhého stupně $\mathbf{y}_s = \mathbf{y}(\xi^s)$ pro $s = 1, \dots, S$. Tento způsob pojetí stochastické optimalizace nazýváme scénářový přístup.

Úlohu (2.5) přepíšeme do tvaru

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{x} + \sum_{s=1}^S p_s Q(\mathbf{x}, \boldsymbol{\xi}^s) \\ \text{za podmínek} \quad & \mathbf{Ax} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{2.6}$$

kde

$$\begin{aligned} Q(\mathbf{x}, \boldsymbol{\xi}^s) = \min_{\mathbf{y}_s} \quad & \mathbf{q}_s^\top \mathbf{y}_s \\ \text{za podmínek} \quad & \mathbf{Wy}_s = \mathbf{h}_s - \mathbf{T}_s \mathbf{x}, \\ & \mathbf{y}_s \geq \mathbf{0}. \end{aligned} \tag{2.7}$$

Dosazením (2.7) do účelové funkce (2.6) získáme rozsáhlou úlohu lineárního programování. Scénářový přístup se využívá při počítačovém modelování. Velikost úlohy rychle roste se zvětšujícím se S , proto při modelování požadujeme, aby S bylo „rozumně“ velké (v závislosti na hardwarovém vybavení). V případech, kdy počet scénářů je nad rámec výpočetní kapacity, je nutné tento počet vhodným způsobem zredukovat. Scénářový přístup uplatníme v následující kapitole v úloze řešící provoz teplárny.

3. Model

V této kapitole se budeme věnovat tvorbě modelu pro plánování provozu teplárny a jeho následnému řešení. Některé základní myšlenky a technické parametry strojů byly převzaty z [13].

Teplárna musí být v každém čase schopna generovat dostatečné množství tepla, aby uspokojila aktuální poptávku v oblasti, kterou zásobuje. Výše poptávky není předem známa, jedná se o stochastickou veličinu. Je závislá především na venkovní teplotě a aktuální denní hodině, lze ji pouze odhadnout na základě historických dat.

Teple je v našem případě vytvářeno kotli na plyn a biomasu. Provoz kotlů na biomasu se musí plánovat s dostatečným předstihem. Je nutné zajistit palivo, tj. dřevní štěpku, pro jejich provoz. Kotle jsou také obtížněji říditelné. Mohou být buď vypnuté, nebo pracovat na plný výkon. Doba náběhu kotlů i útlumu je přibližně 30 minut. Výhodou těchto kotlů jsou ale nižší náklady na výrobu tepla [14] a jejich ekologičnost. Plynové kotle je doplňují. Jejich provoz není nutné plánovat dopředu, mají krátkou dobu náběhu, jejich výkon je ovladatelný, jedinými nevýhodami jsou vyšší provozní náklady a fakt, že plyn patří mezi neobnovitelné zdroje, a tudíž má špatný faktor primární energie [15].

Z předchozího odstavce plyne, že provoz kotlů na biomasu musí být určen dříve, než je známa skutečná poptávka, jedná se o *here-and-now* (HN) přístup. Rozdíl mezi plánovanou výrobou a skutečnou poptávkou bude dorovnán pomocí dražších plynových kotlů, jedná se o *wait-and-see* (WS) přístup.

Cílem je vytvořit plán provozu teplárny tak, aby byla uspokojena poptávka a dosáhlo se maximálního zisku.

3.1. Schéma modelu

V následující části se budeme zabývat popisem modelu. Na obrázku 3.1 je zobrazeno schéma modelu. Prvky diagramu jsou pro lepší orientaci očíslovány. Vstupy můžeme rozdělit do tří skupin. První skupinu tvoří finanční vstupy, které jsou znázorněny pomocí šestiúhelníků. Do druhé skupiny řadíme fyzikální, technické a demografické vstupy, u kterých předpokládáme, že jsou po celou dobu konstantní. Tyto vstupy mají tvar oválu. Poslední skupinou jsou řídicí proměnné značené pomocí kosodélníků.

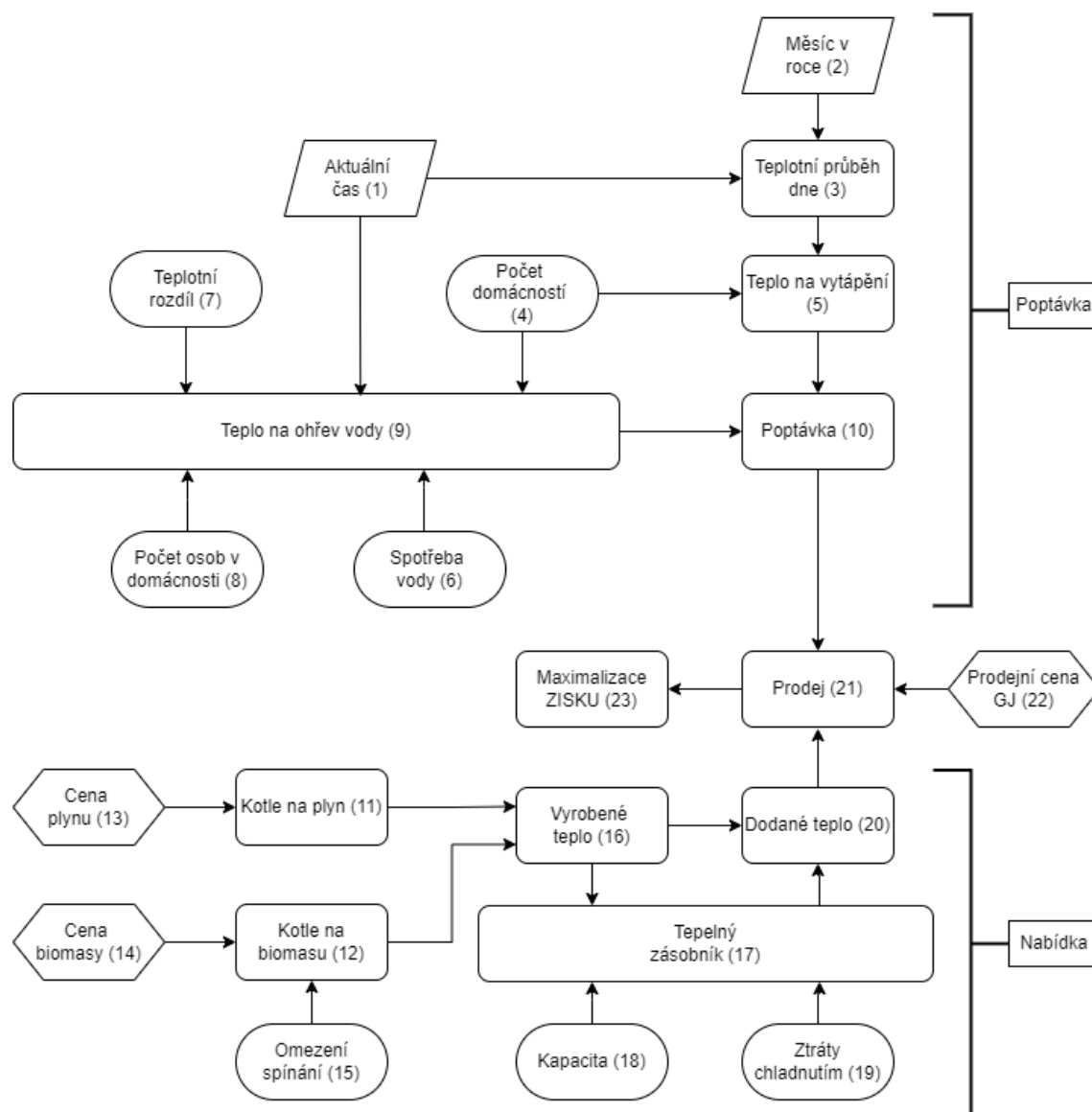
V (zaoblených) obdélnících jsou prvky závislé na vstupech. Vztahy a závislosti jsou značeny pomocí šipek. Nyní si popíšeme jednotlivé prvky diagramu, v závorce je vždy uvedeno označení prvku v diagramu.

3.2. Poptávané teplo

Poptávka (10). Poptávané teplo se skládá z tepla na vytápění a z tepla potřebného k přípravě teplé vody.

3.2.1. Teplo na vytápění

Aktuální čas (1). Jako základní perioda pro dané období byl zvolen jeden den, který byl rozdělen do 48 půlhodinových intervalů. Půlhodinový interval odpovídá době náběhu



Obrázek 3.1: Diagram modelu. (Vlastní tvorba.)

kotlů na biomasu. Předpokládáme, že výše poptávky je v rámci půlhodinových intervalů konstantní.

Měsíc v roce (2). Poptávka po teple na vytápění je závislá na ročním období. V zimních měsících, kdy jsou teploty výrazně nižší, je poptávka výrazně vyšší než v letních měsících. Rok rozdělíme na 12 měsíců a v každém vytvoříme plán základní periody, tj. jednoho dne. U měsíců budeme zohledňovat jejich délku, měsíci únor přiřadíme průměrnou délku zohledňující přestupné roky, tj. 28,25 dne.

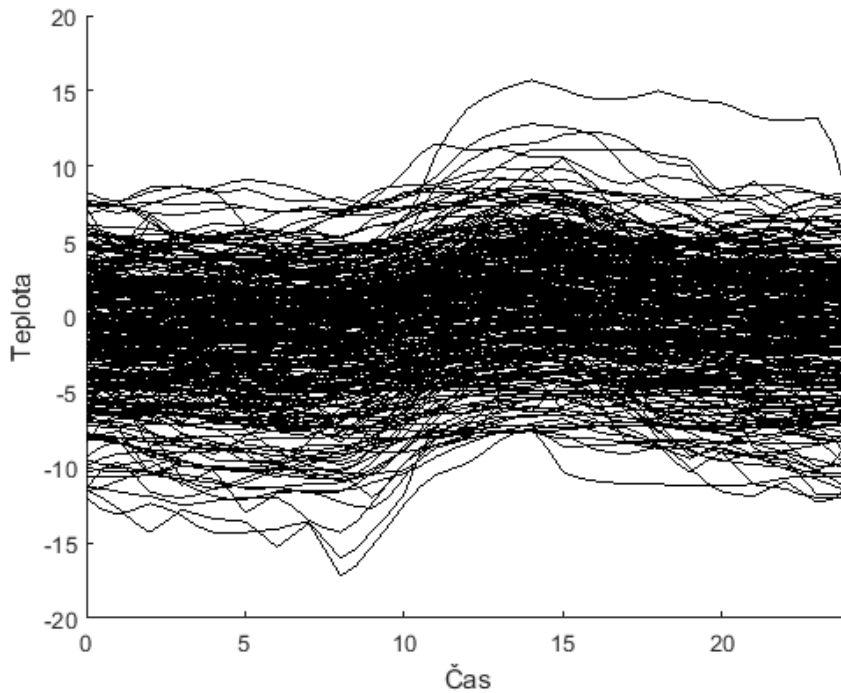
Teplotní průběh dne (3). Množství tepla potřebné k vytápění domácností je závislé na venkovní teplotě, proto poptávku po teple stanovíme na základě historických dat o teplotě. Český hydrometeorologický ústav poskytuje historická data pouze komerčně - dle zákona

3.2. POPTÁVANÉ TEPLLO

jsou povinni zveřejňovat pouze průměrnou, minimální a maximální teplotu, nikoliv průběh teploty během dne.

Jako alternativní zdroj dat použijeme webovou stránku [16] zobrazující historická data z meteorostanice *Brno - Tuřany*. Data budeme považovat za správná. Stránka zobrazuje průběh denní teploty pouze ve formě grafu. V rámci sběru dat byl vytvořen program, který postupně prošel všechny denní průběhy teplot v letech 2011 až 2020 a z grafů určil hodnoty teplot v půlhodinových intervalech. Vzhledem k velikosti jsou původní data a program pouze na vyžádání u autora. Přibližně 5 % původních časových řad bylo vyřazeno, neboť nebyly kompletní (chybějící data, údržba stanice, změna času apod.).

Po očištění dat máme k dispozici ke každému měsíci 250 - 300 časových řad. Na obrázku 3.2 jsou znázorněny časové řady měsíce ledna. Tento počet scénářů není pro výpočty vhodný kvůli kombinatorické explozi (viz kapitola 2.4), proto ho zredukujeme a pro každý měsíc vytvoříme 10 reprezentativních scénářů s příslušnými pravděpodobnostmi.



Obrázek 3.2: Denní průběh teploty - leden. (Vlastní tvorba.)

Generování scénářů na základě momentových charakteristik.

Časové řady v každém měsíci rozdělíme do 10 disjunktních tříd. Z každé třídy poté vytvoříme jeden scénář s příslušnou pravděpodobností. Kvalitu třídy (resp. shluku) budeme hodnotit pomocí rozptylu časových řad v dané třídě [17]. Požadujeme, aby součet rozptylů přes všechny třídy byl minimální. Řešíme úlohu

$$\min_{m \in M} \min_{s \in S} \min_{t \in T} \left(\frac{\sum_{j \in J} u_{mjs} \cdot TE_{mjt}^2}{n_{ms}} - \left(\frac{\sum_{j \in J} u_{mjs} \cdot TE_{mjt}}{n_{ms}} \right)^2 \right)$$

$$\begin{aligned}
&\text{za podmínek } \sum_{j \in J} u_{mjs} = n_{ms}, \quad \forall m \in M, \forall s \in S, \\
&\sum_{s \in S} u_{mjs} = 1, \quad \forall j \in J, \forall m \in M, \\
&n_{ms} \geq 1, \quad \forall m \in M, \forall s \in S, \\
&u_{mjs} \in \{0; 1\}, \quad \forall j \in J, \forall m \in M, \forall s \in S, \\
&M = \{1, 2, \dots, 12\}, \\
&T = \{1, 2, \dots, 48\}, \\
&S = \{1, 2, \dots, 10\}, \\
&J = \{1, 2, \dots, J_m\},
\end{aligned}$$

kde J_m je počet časových řad v měsíci m , TE_{mjt} je teplota v řadě j , čase t a měsíci m , binární proměnná u_{mjs} sděluje, zda j -tá časová řada v měsíci m patří do třídy s , n_{ms} značí počet časových řad v s -té třídě v měsíci m . Jedná se o nelineární přiřazovací úlohu, kterou řešíme pomocí programu GAMS (viz sekce 3.4).

Po vyřešení z každé třídy vytvoříme jeden scénář Te_{mst} pomocí průměru. Pravděpodobnost scénáře ps_{ms} bude úměrná počtu časových řad v dané třídě. Vygenerované scénáře pro měsíc leden jsou zobrazeny na obrázku 3.3.

$$\begin{aligned}
Te_{mst} &= \frac{\sum_{j \in J} u_{mjs} \cdot TE_{mjt}}{n_{ms}}, \quad \forall m \in M, \forall s \in S, \forall t \in T, \\
ps_{ms} &= \frac{n_{ms}}{J_m}, \quad \forall m \in M, \forall s \in S.
\end{aligned}$$

Určení pomocí shlukové analýzy.

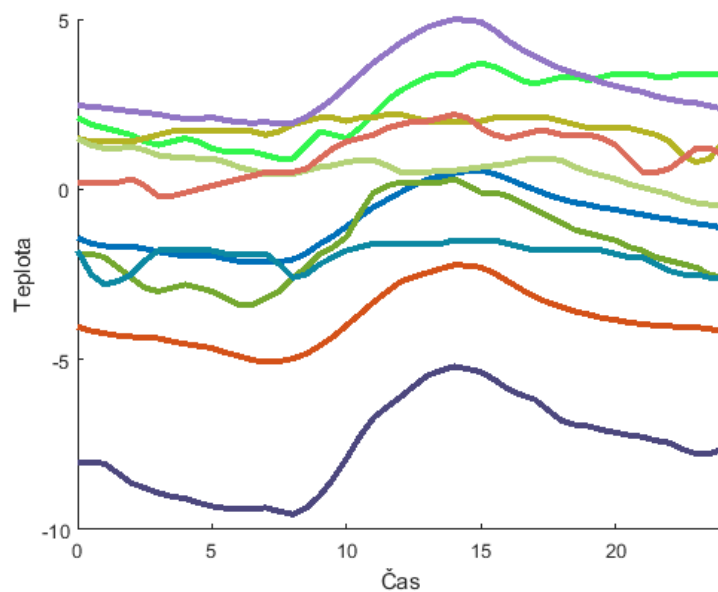
Naši časovou řadu můžeme chápat jako bod v prostoru \mathbb{R}^{48} . Pomocí nehierarchické shlukové analýzy můžeme časové řady rozdělit do 10 shluků. Výsledek shlukové analýzy časových řad měsíce ledna provedené v programu MATLAB [18] je zobrazen na obrázku 3.4.

Z výše uvedených metod se pro daný problém jeví jako nejvhodnější rozdělení na základě momentových charakteristik, zbylá data jsou součástí digitální přílohy. V dalších úvahách bude teplota funkcí měsíce, scénáře a času.

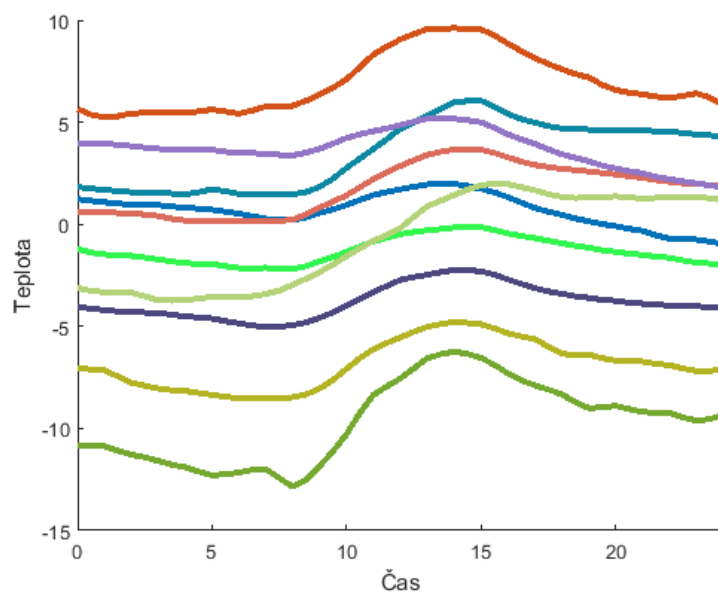
Počet domácností (4). Počet domácností, jež jsou zásobovány teplárnou, byl stanoven na 2 600. Jedná se o konstantní parametr - nepředpokládá se, že budoucí výstavba znaitelně ovlivní tento počet.

Teplo na vytápění (5). Teplo potřebné k vytápění je závislé na venkovní teplotě [19, s. 510]. Je-li teplý letní den, není nutné domácnosti vytápět. Poklesne-li průměrná venkovní teplota vzduchu pod 13 °C ve dvou dnech po sobě následujících [20], je dle legislativy zahájena topná sezóna. V rámci zjednodušení vynecháme požadavek dvou po sobě jdoucích dní a topnou sezónu budeme stanovovat jen na základě aktuální venkovní teploty. Kotle jsou navrženy tak, aby i při částečném výpadku byly schopny zásobovat domácnosti, a to i při silných mrazech.

3.2. POPTÁVANÉ TEPLŮ



Obrázek 3.3: Scénáře měsíce ledna generované pomocí momentových charakteristik. (Vlastní tvorba.)



Obrázek 3.4: Scénáře měsíce ledna generované pomocí shlukové analýzy v MATLABu. (Vlastní tvorba.)

Množství poptávaného tepla Q_t v kWh během půlhodinového časového intervalu určíme vztahem

$$Q_t = \begin{cases} k \cdot d \cdot (13 - Te) & Te \leq 13 \\ 0 & Te > 13 \end{cases},$$

kde Te je venkovní teplota ve stupních Celsia a d počet domácností. Koeficient k zahrnuje zbylé fyzikální atributy (průměrná velikost vnějšího pláště budovy, součinitel tepelné vodivosti, tloušťka stěn apod.). Na základně historických dat a reálného množství vyrobené energie byl vyčíslen koeficient $k = 0,0922$ [21].

3.2.2. Teplo na ohřev vody

Spotřeba vody (6). Průměrná spotřeba teplé vody (TUV) je stanovena na 35 litrů na osobu a den, viz [22].

Teplotní rozdíl (7). Teplá voda dopravovaná do domácností musí dosahovat teploty okolo 55 °C, viz [23].

Počet osob v domácnosti (8). Průměrnou domácnost tvoří 2,1 osob, viz [24].

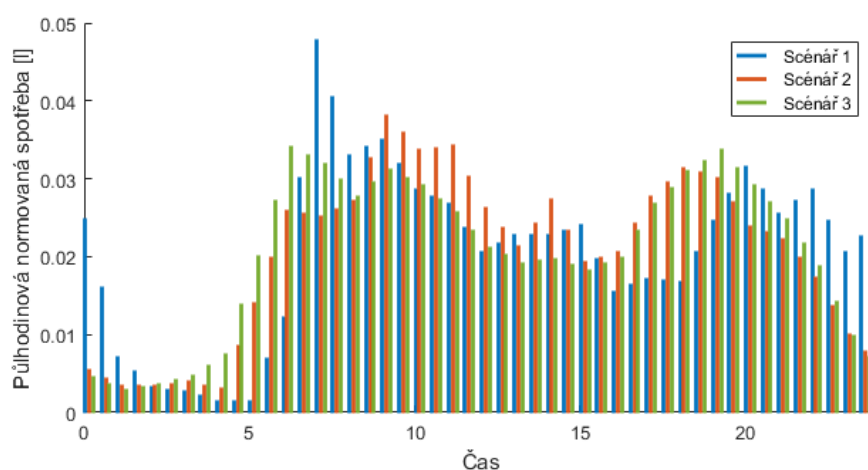
Teplo na ohřev vody (9). Spotřeba vody je závislá na části dne. Největší odběr je zaznamenáván během ranních hodin, nejmenší naopak v nočních hodinách. Přesný odběr v závislosti na čase není znám, pro účely modelu použijeme 3 scénáře převzaté z [25] a [26], každému přidělíme pravděpodobnost 1/3. Dále předpokládáme, že spotřeba vody je během roku konstantní.

Obrázek 3.5 znázorňuje normovanou spotřebu vody pro jednotlivé scénáře, tj. jaká část denního odběru vody je spotřebována v daném půlhodinovém intervalu.

Množství tepla Q_v potřebného k ohřívání je určeno vztahem [19, s. 501]

$$Q_v = m \cdot c \cdot dT,$$

kde m je hmotnost, c značí měrnou tepelnou kapacitu, která pro vodu nabývá hodnoty $4180 \text{ J kg}^{-1} \text{ °C}^{-1}$, dT teplotní rozdíl mezi koncovým a počátečním stavem. Pro vodu platí, že 1 l má hmotnost 1 kg.



Obrázek 3.5: Normovaná spotřeba teplé vody během dne, 3 scénáře. (Převzato z [25] a [26]).

3.3. Dodávané teplo

Kotle na plyn (11). V uvažované lokalitě jsou k dispozici 4 plynové kotle, každý o výkonu 4 650 kW. Kotle jsou snadno ovladatelné, plného výkonu jsou schopny dosáhnout během desítek sekund (*wait-and-see* přístup, výkon kotle může reagovat na aktuální poptávku). Množství tepla dodávané soustavou plynových kotlů je závislé na měsíci, čase, scénáři pro poptávku tepla a scénáři pro poptávku vody.

Kotle na biomasu (12). V teplárně jsou nainstalovány dva kotle na biomasu o výkonu 1 100 kW a 1 500 kW. Mohou být buď odstavené, nebo pracovat na plný výkon - během půlhodiny pak dodávají 550 kWh, resp. 750 kWh. Doba náběhu kotlů je přibližně 30 minut. Plán jejich provozu musí být stanoven s dostatečným předstihem (*here-and-now* přístup, nereaguje na aktuální poptávku), aby bylo zajištěno palivo a provoz.

Cena plynu (13). Přímé náklady plynu na výrobu 1 kWh byly vyčísleny na 1,2 Kč.¹

Cena biomasy (14). Přímé náklady biomasy na výrobu 1 kWh byly vyčísleny na 1 Kč.¹

Omezení sepnutí (15). Každý kotel na biomasu lze spustit (resp. zastavit) nejvýše dvakrát denně.

Vyrobené teplo (16). Vyrobené teplo je součtem tepla vyrobeného kotli na plyn a biomasu.

Tepelný zásobník (17). V soustavě je zabudován tepelný zásobník sloužící k uchování tepla, které není aktuálně spotřebováno. Je limitován svou kapacitou. Při používání zásobníku vznikají tepelné ztráty.

Kapacita (18). Kapacita tepelného zásobníku je stanovena na energetický ekvivalent 2 000 kWh.

Ztráty chlazením (19). Půlhodinová tepelná ztráta zásobníku je stanovena na 5 % jeho aktuálního stavu zásob.

Dodané teplo (20). Dodané teplo je součtem vyrobeného tepla a tepla použitého ze zásobníku.

Prodej (21). Poptávka po teple musí být v každém okamžiku plně uspokojena.

Prodejní cena GJ (22). Dle ceníku [27] se dodané teplo prodává za 621 Kč/GJ, což odpovídá 2,236 Kč/kWh.

Maximalizace zisku (23). Cílem je maximalizovat hodnotu účelové funkce, která představuje zisk.

¹Cena z podzimu 2021.

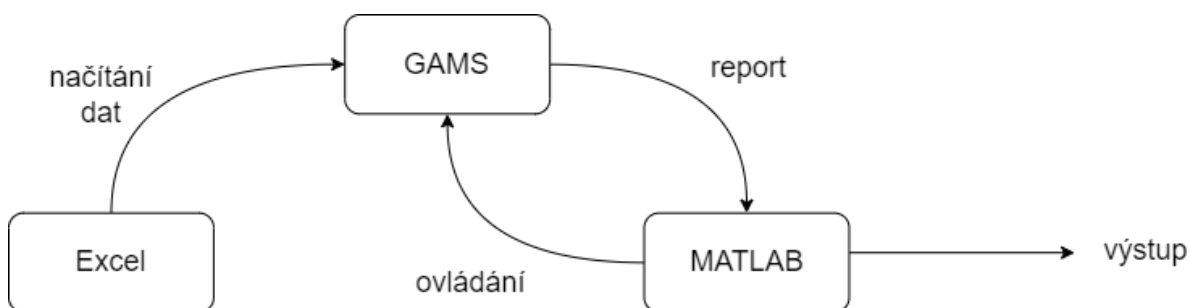
3.4. Softwarová implementace

V této části se budeme zabývat programováním našeho modelu a následným řešením. Hlavním nástrojem je modelovací jazyk GAMS², který bude představen v následujícím odstavci.

GAMS (General Algebraic Modeling System/Obecný algebraický modelovací systém) je vysokoúrovňový modelovací systém určený pro matematickou optimalizaci. Je navržen pro lineární, nelineární i celočíselné optimalizační úlohy a využívá se pro velké komplexní úlohy. Detailnější popis jazyka GAMS včetně jeho syntaxe a sémantiky lze nalézt v [11].

GAMS je sice pokročilý modelovací systém, ale není vhodný k statistickému zpracování a vizualizaci výsledných dat. K tomu využijeme program MATLAB³. Poznamenejme, že z důvodu jednoduchosti jsou některá data načítána z aplikace Excel⁴.

Data mezi Excelem, GAMSem a MATLABem se předávají pomocí GDX (GAMS data exchange) souboru. MATLAB má implementované funkce pro zápis i čtení GDX souborů, jsou ale poměrně složité a uživatelsky nepříjemné, proto byla použita knihovna [31] pro pohodlnější práci s GDX soubory. Vztahy mezi jednotlivými softwary jsou zaznačeny na obrázku 3.6.



Obrázek 3.6: Vztahy mezi použitými softwary. (Vlastní tvorba.)

Nyní se zaměříme na samotný kód v GAMSu. Uvedeme a okomentujeme jeho stěžejní části. Plná verze a zbylé kódy jsou součástí digitální přílohy.

Na začátku definujeme indexové množiny, které budeme používat.

$$\begin{aligned}
 t &\in \{1, 2, \dots, 48\} = T, \\
 s &\in \{1, 2, \dots, 10\} = S, \\
 r &\in \{1, 2, 3\} = R, \\
 m &\in \{1, 2, \dots, 12\} = M.
 \end{aligned} \tag{3.1}$$

Set			
t	čas (půlhodinový interval)	/ 1 * 48 / ,	
s	scénáře - teplota	/ 1 * 10 / ,	
r	scénáře - voda	/ 1 * 3 / ,	
m	měsíce	/ 1 * 12 / ;	

²GAMS 28.2 [28].

³MATLAB 2016a [29].

⁴Microsoft Excel 2007 [30].

3.4. SOFTWAREOVÁ IMPLEMENTACE

Je-li kdekoliv v kódu použitý index značící množinu, musí daný vztah platit pro všechny prvky uvedené množiny, není-li uvedeno jinak. Dále zadefinujeme parametry.

$$\begin{aligned}
 \text{Počet domácností:} & d = 2600, \\
 \text{počet osob na domácnost:} & o = 2,1, \\
 \text{změna teploty vody během ohřevu:} & dT = 60, \\
 \text{spotřeba teplé vody na osobu na den:} & vc = 35, \\
 \text{koeficient vytápění:} & k = 0,0922, \\
 \text{pravděpodobnost scénáře r:} & pr = \left(\frac{1}{3}; \frac{1}{3}; \frac{1}{3} \right).
 \end{aligned} \tag{3.2}$$

Parameter			
domacnosti	počet domácností	/ 2600 /	,
osob_v_domacnosti	počet osob v domácnosti	/ 2.1 /	,
d_T	teplotní rozdíl ohřívání vody	/ 60 /	,
spotreba_vody_na_osobu	denní spotřeba TUV na osobu	/ 35 /	,
koeficient_vytapeni	koeficient vytápění	/ 0.0922 /	,
p_r(r)	pravděpodobnost scénáře r	/ 1 0.333333, 2 0.333333 , 3 0.333333 / ;	

Poslední řádek kódu říká, že pravděpodobnost všech tří scénářů r je rovna $1/|R| = 1/3$. Dále z Excelu načteme hodnoty venkovních teplot v závislosti na měsíci, scénáři a čase T_{emst} , poté jejich pravděpodobnosti ps_{ms} a nakonec půlhodinovou spotřebu teplé vody pro jednotlivé scénáře v_{rt} .

```

$onEcho > data_teplota.txt
I="%system.fp%data.xlsx"      R=List1!B2:AZ123      O=data_teplota.inc
offEcho
$call = xls2gms @data_teplota.txt
Table teplota(m,s,t) teplota v měsíci m scénáři s čase t
                        $include data_teplota.inc ;

$onEcho > p_s.txt
I="%system.fp%data.xlsx"      R=List2!B2:L14        O=p_s.inc
$offEcho
$call = xls2gms @p_s.txt
Table p_s(m,s) pravděpodobnost scénáře s v měsíci m
                        $include p_s.inc ;

$onEcho > spotreba_vody.txt
I="%system.fp%data.xlsx"      R=List3!B2:AX5        O=spotreba_vody.inc
$offEcho
$call = xls2gms @spotreba_vody.txt
Table spotreba_vody(r,t) spotřeba vody scénáře r v čase t
                        $include spotreba_vody.inc ;

```

Poptávka po teple je určena vztahy

$$\begin{aligned}
 \text{teplo na vytápění:} & \quad Q_{t_{mst}} = \max\{13 - T_{e_{mst}}; 0\} \cdot d \cdot k, \\
 \text{teplo na ohřev vody:} & \quad Q_{v_{rt}} = v_{rt} \cdot vc \cdot d \cdot o \cdot dT \cdot 4,18/3600, \\
 \text{celkové poptávané teplo :} & \quad Q_{msrt} = Q_{t_{mst}} + Q_{v_{rt}}.
 \end{aligned} \tag{3.3}$$

```

teplo_na_vytapeni(m,s1,t) = max( 13 - teplota(m,s1,t), 0 ) *
                                domacnosti * koeficient_vytapeni;

teplo_na_ohrev_vody(s2,t) = spotreba_vody(s2,t) * domacnosti *
                                osob_v_domacnosti * spotreba_vody_na_osobu * d_T * 4.18 / 3600;

teplo_poptavane(m,s1,s2,t) = teplo_na_vytapeni(m,s1,t) +
                                teplo_na_ohrev_vody(s2,t);

```

Nyní se podíváme na stranu nabídky. Pro označení kotlů na biomasu

$$i \in \{1, 2\} = I, \tag{3.4}$$

kde 1 je použitá pro menší kotel, 2 pro větší kotel.

Set	
i	označení kotlů na biomasu / 1 * 2 / ;

Dále definujeme potřebné parametry

$$\begin{aligned}
 \text{energie z i-tého kotle na biomasu:} & \quad e_i = (550; 750), \\
 \text{cena 1 kWh plynu:} & \quad cp = 1,2, \\
 \text{cena 1 kWh biomasy:} & \quad cb = 1,0, \\
 \text{cena 1 GJ:} & \quad cg = 621, \\
 \text{omezení počtu sepnutí:} & \quad om = 4, \\
 \text{koeficient chladnutí zásobníku:} & \quad ch = 0,95.
 \end{aligned} \tag{3.5}$$

Parameter	
energie(i)	energie z i-tého kotle / 1 550, 2 750 / ,
cena_plyn	cena výroby 1kWh plynu / 1.2 / ,
cena_biomasa	cena výroby 1kWh biomasy / 1.0 / ,
cena_GJ	prodejní cena 1 GJ / 621 / ,
sepnuti_omezeni	omezení počtu sepnutí / 4 / ,
koef_chladnuti	koeficient chladnutí zásobníku / 0.95 / ;

Kotle na biomasu jsou buď odstavené, nebo pracují na plný výkon. Pro popis jejich činnosti definujeme binární proměnou

$$x_{mit} \in \{0, 1\}, \tag{3.6}$$

3.4. SOFTWAREOVÁ IMPLEMENTACE

která nabývá hodnoty 1, je-li v m-tém měsíci i-tý kotel v čase t v provozu.

Binary variables

$x(m,i,t)$ činnost i-tého kotle v čase t v měsíci m ;

Následně vytvoříme nezáporné proměnné

$$\begin{aligned}
 \text{energie z plynových kotlů:} & \quad y_{mtsr} \geq 0, \\
 \text{stav zásobníku:} & \quad za_{mtsr} \geq 0, \\
 \text{stav zásobníku o půlnoci:} & \quad zp_m \geq 0, \\
 \text{označení zapnutí kotle:} & \quad dxp_{mit} \geq 0, \\
 \text{označení vypnutí kotle:} & \quad dxm_{mit} \geq 0.
 \end{aligned} \tag{3.7}$$

První proměnná popisuje, jaké množství tepla dodávají plynové kotle, druhá určuje aktuální stav tepelného zásobníku. Třetí proměnná slouží k tomu, aby stav zásobníků na začátku a konci každé periody byl pro dané období stejný. Poslední dvě proměnné jsou pomocné k určení počtu sepnutí.

Positive variables

plynovy_kotel(m,t,s,r) výroba plynovými kotli v měsíci m
čas t scénáři s r ,
stav_zasobniku(m,t,s,r) stav zásobníku v měsíci m čas t
scénáři s r ,
zasobnik_pulnoc(m) stav zásobníku o půlnoci,
dxp(m,i,t) čítat zapnutí ,
dxm(m,i,t) čítat vypnutí ;

Maximální výkon soustavy plynových kotlů je 18 600 kW, tudíž během půlhodiny jsou schopny dodat nejvýše 9 300 kWh.

$$y_{mtsr} \leq 9300 \tag{3.8}$$

plynovy_kotel.up(m,t,s,r) = 9300;

Maximální množství tepla, které je schopen zásobník pojmout, je stanoveno na 2 000 kWh.

$$za_{mtsr} \leq 2000 \tag{3.9}$$

stav_zasobniku.up(m,t,s,r) = 2000;

Jak již bylo zmíněno výše, na začátku a na konci periody musí být tepelný zásobník ve stejném stavu. Toho docílíme podmínkami

$$\begin{aligned}
 za_{m1sr} &= zp_m, \\
 za_{m48sr} &= zp_m.
 \end{aligned} \tag{3.10}$$

Omezení počtu sepnutí naplníme pomocí podmínky

$$\begin{aligned} dxp_{mit} - dxm_{mit} &= x_{mit} - x_{mi(t-1)} \quad , \forall t > 1, \\ \sum_{t \in T} (dxp_{mit} + dxm_{mit}) &\leq om. \end{aligned} \quad (3.11)$$

Dále musí platit, že po vypnutí není možné kotel hned zapnout a obráceně.

$$\begin{aligned} dxp_{mi(t-1)} + dxm_{mit} &\leq 1 \quad , \forall t > 1, \\ dxp_{mit} + dxm_{mi(t-1)} &\leq 1 \quad , \forall t > 1. \end{aligned} \quad (3.12)$$

```
vypocet_pulnoc_1(m,s,r) ..
    stav_zasobniku(m,'1',s,r) =E= zasobnik_pulnoc(m);

vypocet_pulnoc_2(m,s,r) ..
    stav_zasobniku(m,'48',s,r) =E= zasobnik_pulnoc(m);

vypocet_sepnuti(m,i,t)$(ord(t) GT 1) ..
    dxp(m,i,t) - dxm(m,i,t) =E= x(m,i,t) - x(m,i,t-1);

vypocet_sepnuti_omezeni(m,i) ..
    sum(t, dxp(m,i,t) + dxm(m,i,t)) =L= sepnuti_omezeni;

vypocet_minimalni_doba_zapnuti(m,i,t)$(ord(t) GT 1) ..
    dxp(m,i,t-1) + dxm(m,i,t) =L= 1;

vypocet_minimalni_doba_vypnuti(m,i,t)$(ord(t) GT 1) ..
    dxp(m,i,t) + dxm(m,i,t-1) =L= 1;
```

Poptávka musí být vždy plně uspokojena.

$$\begin{aligned} Q_{msrt} &\leq y_{mts r} + \sum_{i \in I} (x_{mit} \cdot e_i) + za_{m(t-1)sr} \cdot ch - za_{mts r} \quad , \forall t > 1, \\ Q_{msr1} &\leq y_{m1sr} + \sum_{i \in I} (x_{mi1} \cdot e_i) + za_{m48sr} \cdot ch - za_{m1sr}. \end{aligned} \quad (3.13)$$

První vztah zajišťuje splnění poptávky během dne, druhý vztah řeší situaci mezi začátkem a koncem periody.

```
splneni_poptavky(m,s,r,t)$(ord(t) GT 1) ..
    teplo_poptavane(m,s,r,t) =L= plynovy_kotel(m,t,s,r) +
    sum(i, x(m,i,t) * energie(i)) + stav_zasobniku(m,t-1,s,r) *
    koef_chladnuti - stav_zasobniku(m,t,s,r) ;

splneni_poptavky_pulnoc(m,s,r) ..
    teplo_poptavane(m,s,r,'1') =L= plynovy_kotel(m,'1',s,r) +
    sum(i, x(m,i,'1') * energie(i)) + stav_zasobniku(m,'48',s,r) *
    koef_chladnuti - stav_zasobniku(m,'1',s,r) ;
```

3.4. SOFTWAREOVÁ IMPLEMENTACE

Poslední část je věnována finanční stránce modelu. Pro každý měsíc tvoříme plán provozu. Počet dní v jednotlivých měsících je různý, proto definujeme parametr

$$\text{počet dní v měsíci: } md_m = (31; 28, 25; 31; 30; 31; 30; 31; 31; 30; 31; 30; 31). \quad (3.14)$$

Parameter

pocet_dni_v_mesici(m) počet dní v jednotlivých měsících
 / 1 31, 2 28.25, 3 31, 4 30, 5 31, 6 30,
 7 31, 8 31, 9 31, 10 31, 11 30, 12 31 / ;

Dále definujeme nezáporné proměnné a jednu volnou proměnnou.

$$\begin{aligned} \text{Denní náklady na plyn:} & \quad np_{msr} \geq 0, \\ \text{denní náklady na biomasu:} & \quad nb_m \geq 0, \\ \text{celkové denní náklady:} & \quad nc_{msr} \geq 0, \\ \text{denní tržby:} & \quad tr_{msr} \geq 0, \\ \text{měsíční náklady:} & \quad mn_m \geq 0, \\ \text{měsíční tržby:} & \quad mt_m \geq 0, \\ \text{celkové náklady:} & \quad cn \geq 0, \\ \text{celkové tržby:} & \quad ct \geq 0, \\ \text{zisk:} & \quad z \in \mathbb{R}. \end{aligned} \quad (3.15)$$

Positive variables

naklady_plyn(m,s,r) denní náklady na výrobu plynu,
 naklady_biomasa(m) denní náklady na biomasu,
 naklady_celkem(m,s,r) celkové denní náklady,
 trzby(m,s,r) denní tržby,
 mesicni_naklady(m) měsíční náklady,
 mesicni_trzby(m) měsíční tržby,
 celkove_naklady roční náklady,
 celkove_trzby roční tržby;

Variables

zisk zisk;

Závěrem uvedeme účelovou funkci skládající se z dílčích nákladu a tržeb. Denní náklady a tržby jsou určeny vztahy

$$\begin{aligned} np_{msr} &= \sum_{t \in T} y_{mts} \cdot cp, \\ nb_m &= \sum_{t \in T} \sum_{i \in I} (x_{mit} \cdot e_i) \cdot cp, \\ nc_{msr} &= np_{msr} + nb_m, \\ tr_{msr} &= \sum_{t \in T} Q_{msrt} \cdot cg / 277,77. \end{aligned} \quad (3.16)$$

Náklady a tržby pro jednotlivé měsíce určíme vztahy

$$\begin{aligned} mn_m &= md_m \cdot \sum_{s \in S} \sum_{r \in R} (nc_{msr} \cdot ps_{ms} \cdot pr_r), \\ mt_m &= md_m \cdot \sum_{s \in S} \sum_{r \in R} (tr_{msr} \cdot ps_{ms} \cdot pr_r), \end{aligned} \quad (3.17)$$

a nakonec určíme celkové roční náklady, celkové roční tržby a zisk

$$\begin{aligned} cn &= \sum_{m \in M} mn_m, \\ ct &= \sum_{m \in M} mt_m, \\ z &= ct - cn. \end{aligned} \quad (3.18)$$

```
vypocet_naklady_plyn(m,s,r) ..
    naklady_plyn(m,s,r) =E= sum(t, plynovy_kotel(m,t,s,r)) *
        cena_plyn;

vypocet_naklady_biomasa(m) ..
    naklady_biomasa(m) =E= sum(t, sum(i, x(m,i,t)*energie(i) )) *
        cena_biomasa;

vypocet_naklady_celkem(m,s,r) ..
    naklady_celkem(m,s,r) =E= naklady_plyn(m,s,r) +
        naklady_biomasa(m);

vypocet_trzby(m,s,r) ..
    trzby(m,s,r) =E= sum(t, teplo_poptavane(m,s,r,t)) *
        cena_GJ / 277.78 ;

vypocet_mesicni_naklady(m) ..
    mesicni_naklady(m) =E= pocet_dni_v_mesici(m) * sum(s, sum(r,
        naklady_celkem(m,s,r) * p_s(m,s) * p_r(r) )) ;

vypocet_mesicni_trzby(m) ..
    mesicni_trzby(m) =E= pocet_dni_v_mesici(m) * sum(s, sum(r,
        trzby(m,s,r) * p_s(m,s) * p_r(r) )) ;

vypocet_celkove_naklady ..
    celkove_naklady =E= sum(m,mesicni_naklady(m));
```

3.4. SOFTWAREOVÁ IMPLEMENTACE

```
vypocet_celkove_trzby ..  
    celkove_trzby =E= sum(m,mesicni_trzby(m));  
  
vypocet_zisk ..  
    zisk =E= celkove_trzby - celkove_naklady ;
```

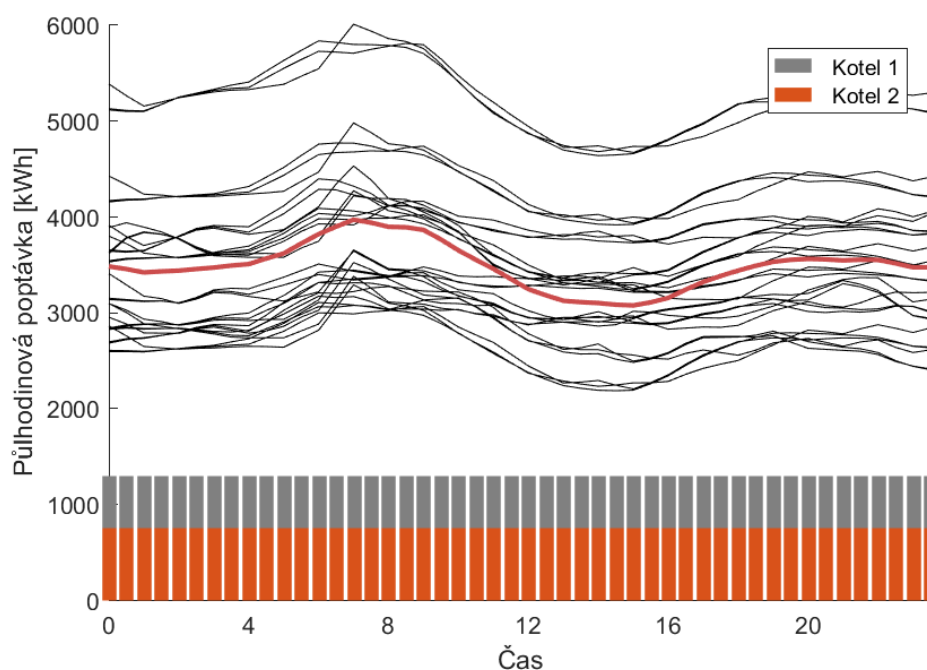
Cílem je maximalizovat účelovou funkci, která představuje zisk označený z , za podmínek (3.1) - (3.18). Model představuje úlohu lineárního dvoustupňového stochastického programování se scénářovým přístupem (viz kapitola 2.5.2), proto jej lze řešit jako lineární celočíselnou úlohu (MIP, *Mixed Integer Programming*), kterou vyřešíme pomocí solveru GUROBI se zadanou relativní přesností 0,0001. Celý model je uveden v příloze.

```
model planovani_provozu /all/;  
Option MIP = gurobi;  
Option optcr = 0.0001;  
solve planovani_provozu maximizing zisk using MIP;
```


3.5. Řešení

Po úvodním preprocessingu se úloha sestává z 27 587 neznámých, z nichž je 1 152 binárních, a 15 610 lineárních omezení. Po 34 341 691 iteracích bylo nalezeno optimální řešení. Plán provozu kotlů na biomasu pro jednotlivé měsíce je znázorněn na obrázku 3.8. Vybarvené políčko značí, že daný kotel má být v daném měsíci a čase v provozu. Nyní se detailněji zaměříme na výsledky pro některé měsíce, zbylé jsou součástí přílohy.

Měsíc leden patří mezi zimní měsíce, ve kterých se teploty pohybují kolem nuly. Aby si lidé udrželi životní komfort, musí kontinuálně topit, takže je zvýšená poptávka po teple. Teplo potřebné k dodávce teplé vody je minoritní vůči teplu na vytápění. Na obrázku 3.7 je znázorněna půlhodinová poptávka po teple v závislosti na denní hodině. Černé čáry představují jednotlivé scénáře, červená čára jejich střední hodnotu. Dle očekávání jsou oba kotle nepřetržitě v provozu po celý den a zbylé teplo je dodáváno soustavou plynových kotlů. Tepelný zásobník není využit.



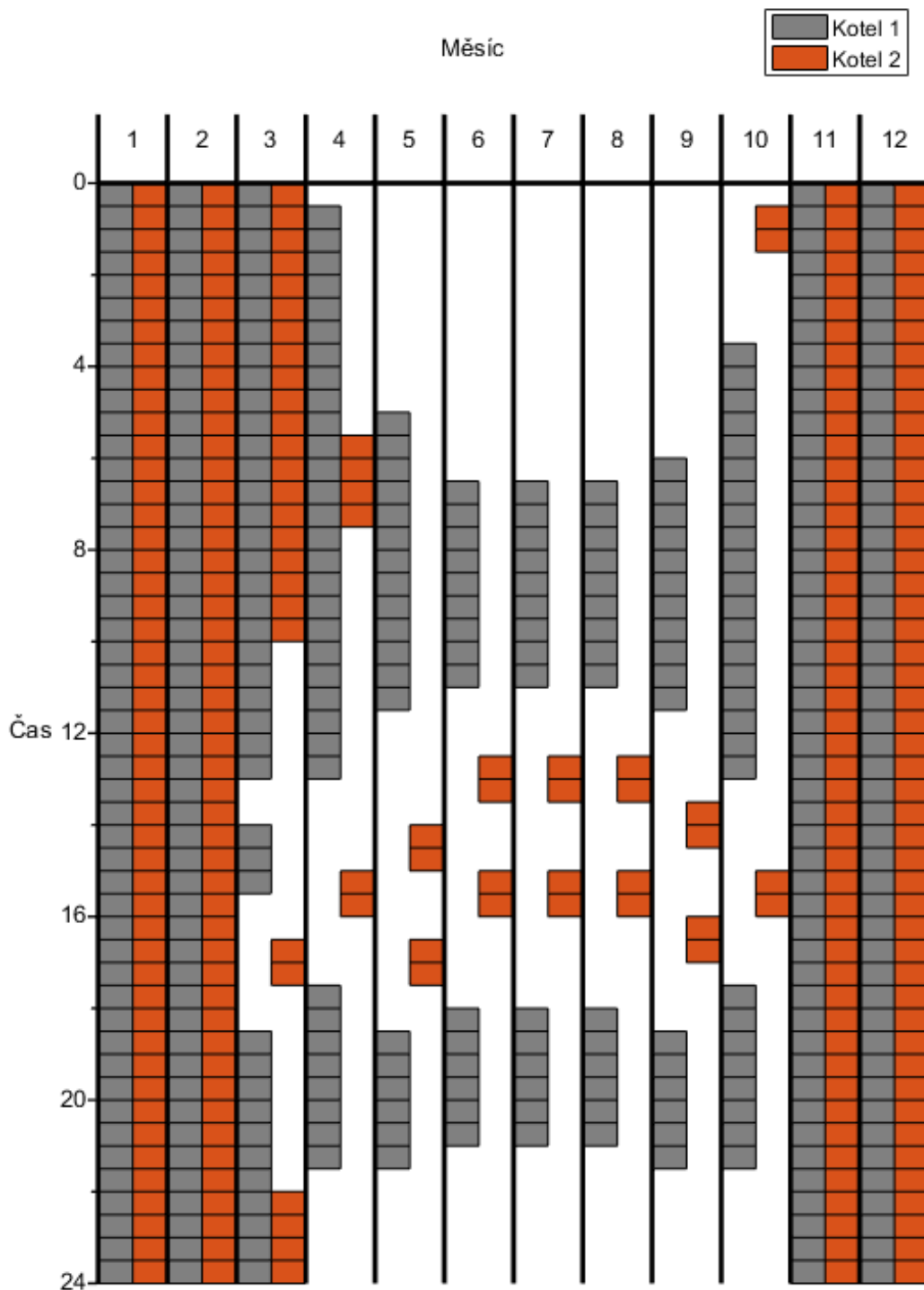
Obrázek 3.7: Plán provozu pro měsíc leden. (Vlastní tvorba.)

Jiná situace je na jaře a na podzim. Venkovní teplota je vyšší, topí se až po západu slunce. Během dne je ale stále nutné ohřívat vodu. Výše půlhodinové poptávky po teple v závislosti na čase pro měsíc duben je uvedena na obrázku 3.9. Kotle na biomasu nejsou v provozu po celý den. Menší kotel je stále v provozu větší část dne, větší kotel se používá jen pár hodin. Ve zbylých časech je teplo dodávané ze zásobníku, nebo pomocí kotlů na plyn.

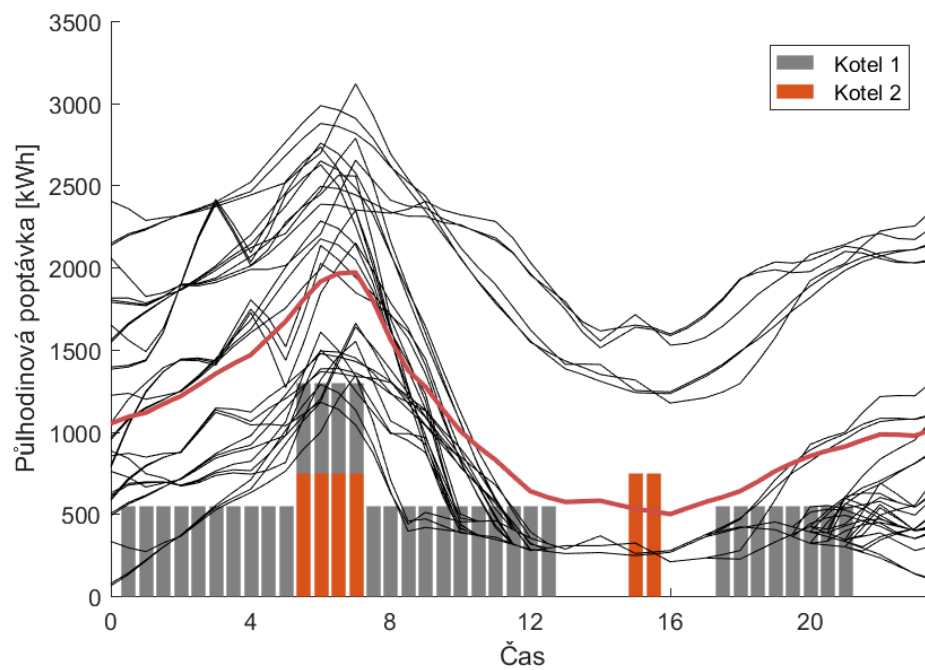
Odlišná situace je v letních měsících. Dodávané teplo je určeno (převážně) k ohřívání vody, teplo určené k vytápění je téměř zanedbatelné. Situace v červenci je zobrazena na obrázku 3.10. (Obrázky 3.9 a 3.10 mají různá měřítka na svislé ose.) Menší kotel na biomasu je v provozu pouze v době, kdy je zvýšená poptávka po teplé vodě, tj. v ranních a večerních hodinách. Větší kotel je během dne v provozu pouze 2 hodiny. Po zbytek

3.5. ŘEŠENÍ

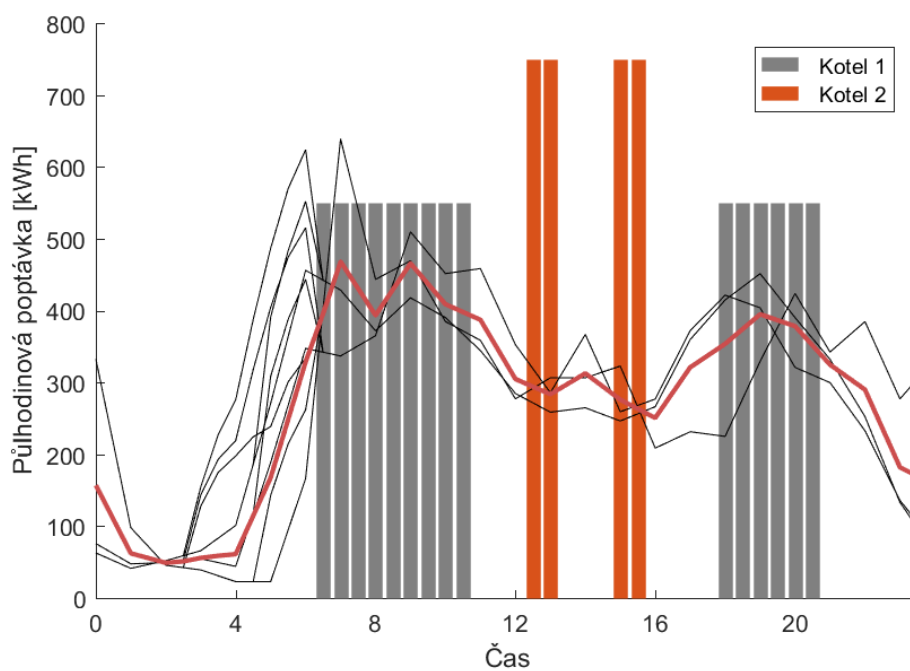
dne je poptávka menší, teplo se dodává ze zásobníku, nebo je vytvořeno pomocí soustavy plynových kotlů.



Obrázek 3.8: Plán provozu kotlů na biomasu. (Vlastní tvorba.)



Obrázek 3.9: Plán provozu pro měsíc duben. (Vlastní tvorba.)



Obrázek 3.10: Plán provozu pro měsíc červenec. (Vlastní tvorba.)

3.6. Analýza citlivosti

V modelu se vyskytují tři ekonomické vstupy, a to náklady na výrobu tepla pomocí plynu, náklady na výrobu tepla pomocí biomasy a prodejní cena tepla koncovým zákazníkům. Poslední z uvedených vstupů může teplárna do jisté míry ovlivnit, zatímco cena plynu a biomasy je ovlivněna aktuálním vývojem trhu. V následující části se budeme zabývat otázkou, jak se změní plán výroby tepla v závislosti na ceně plynu a biomasy, pokud jsou předem známe.

Nutná podmínka smysluplného využití modelu je, že náklady na výrobu tepla pomocí plynu musí být ostře větší než cena na výrobu stejného množství tepla pomocí biomasy. Pokud by výše uvedená podmínka neplatila, nebyl by důvod používat kotle na biomasu, které jsou hůře říditelné a hrozí, že vyrobené teplo nebude využito, čímž vzniknou ztráty. Teplo by se při této situaci vyrábělo pouze plynovými kotli.

Při rostoucí ceně plynu je ekonomicky výhodnější využívat více kotle na biomasu, i přestože vznikají ztráty (nespotřebované teplo, ztráty chlazením tepelného zásobníku), které jsou vykompenzovány nižší cenou dřevní štěpky. Roste-li naopak cena dřevní štěpky, kotle na biomasu jsou utlumovány a používají se jen tehdy, když je vysoká pravděpodobnost, že všechno teplo bude spotřebováno beze ztrát.

Kotle na biomasu mohou při plné vytíženosti vyrobit denně až 62 400 kWh. V základním modelu, ve kterém jsou náklady na 1 kWh plynu 1,2 Kč a biomasy 1,0 Kč, je vytíženost kotlů na biomasu přibližně 61,43 % neboli používají se jen na 61,43 % své kapacity. Model budeme dále řešit pro různé ceny plynu a biomasy.

Pro opakované řešení modelu s různými vstupy si vytvoříme v MATLABu strukturu *subs*, která bude mít následující atributy:

Označení	Význam atributu
name	název datové struktury v GAMSu
type	typ datové struktury
uels	množina názvů indexů
form	forma zápisu
val	hodnoty datové struktury

Pomocí příkazu *wgdx* vytvoříme GDX soubor pro předávání dat mezi programy. Na závěr příkazem *system* spustíme GAMS, který model vyřeší a výsledné hodnoty zapíše do příslušného GDX souboru.

```
subs.name = 'substitute';
subs.type = 'parameter';
subs.uels = {'plyn', 'biomasa'};
subs.form = 'full';
subs.val = [cena_plyn, cena_biomsa];
wgdx('MATLABtoGDX', subs);
system('gams .\analiza_citlivosti.gms lo=3 gdx=GAMStoGDX');
```

Do původního GAMS kódu přidáme parametr *substitute*, který bude sloužit k načítání cen plynu a biomasy.

```
Parameter
    substitute                substitute hodnot    ;
```

Data importujeme pomocí příkazů *\$GDXIN* a *\$LOAD*

```
$GDXIN .\MATLABtoGDX.gdx
$LOAD substitute
$GDXIN
```

a původní hodnoty nahradíme novými.

```
cena_plyn = substitute('plyn');
cena_biomasa = substitute('biomasa');
```

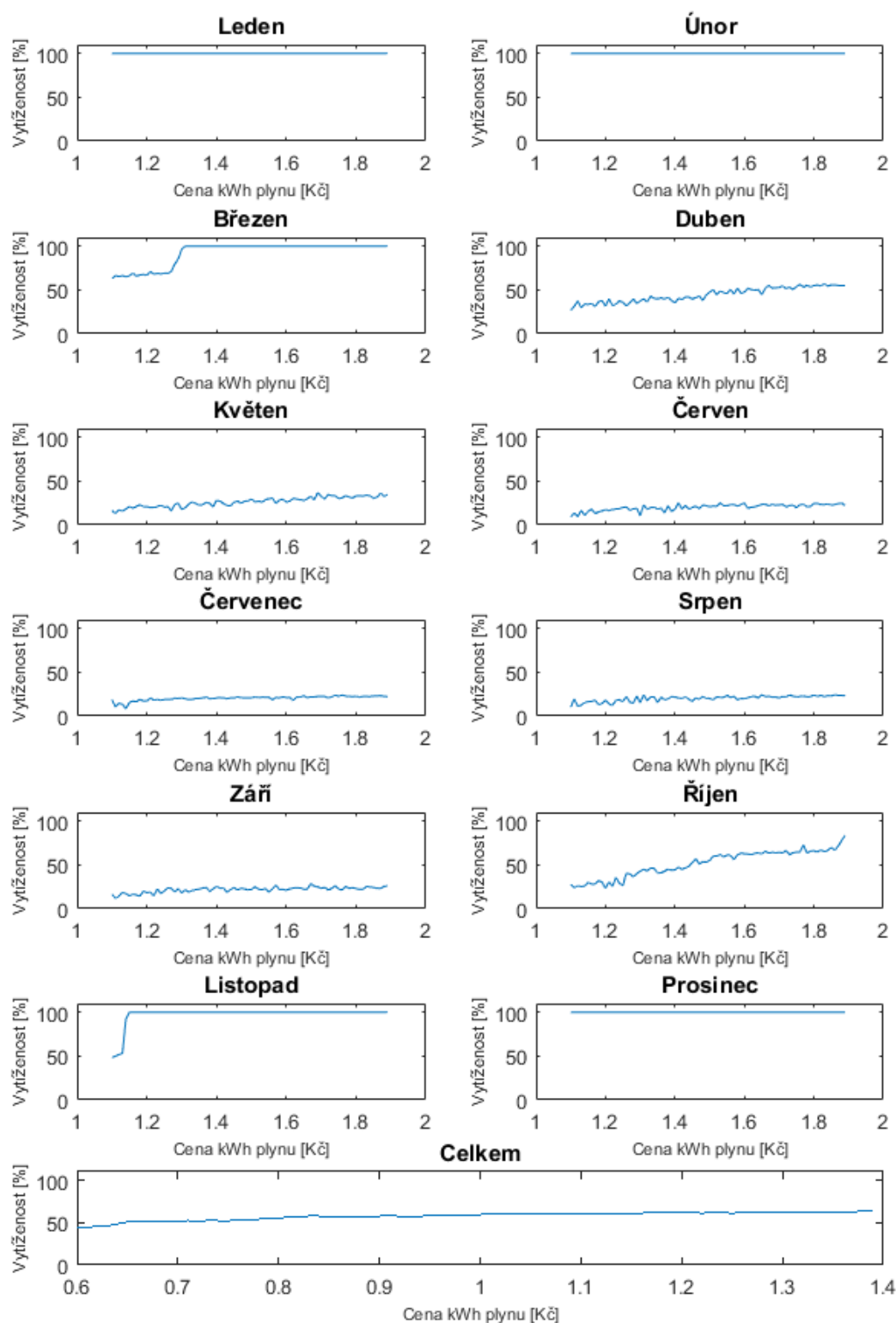
Jedná se opět o úlohu lineárního celočíselného programování (MIP), kterou opět vyřešíme pomocí solveru GUROBI. V tomto případě z důvodu výpočetní náročnosti požadujeme relativní přesnost pouze 0,005, tudíž výsledky mohou být lehce nepřesné, viz kapitola 2.4.

```
model planovani_provozu /all/;
Option MIP = gurobi;
Option optcr = 0.005 ;
solve planovani_provozu maximizing zisk using MIP;
```

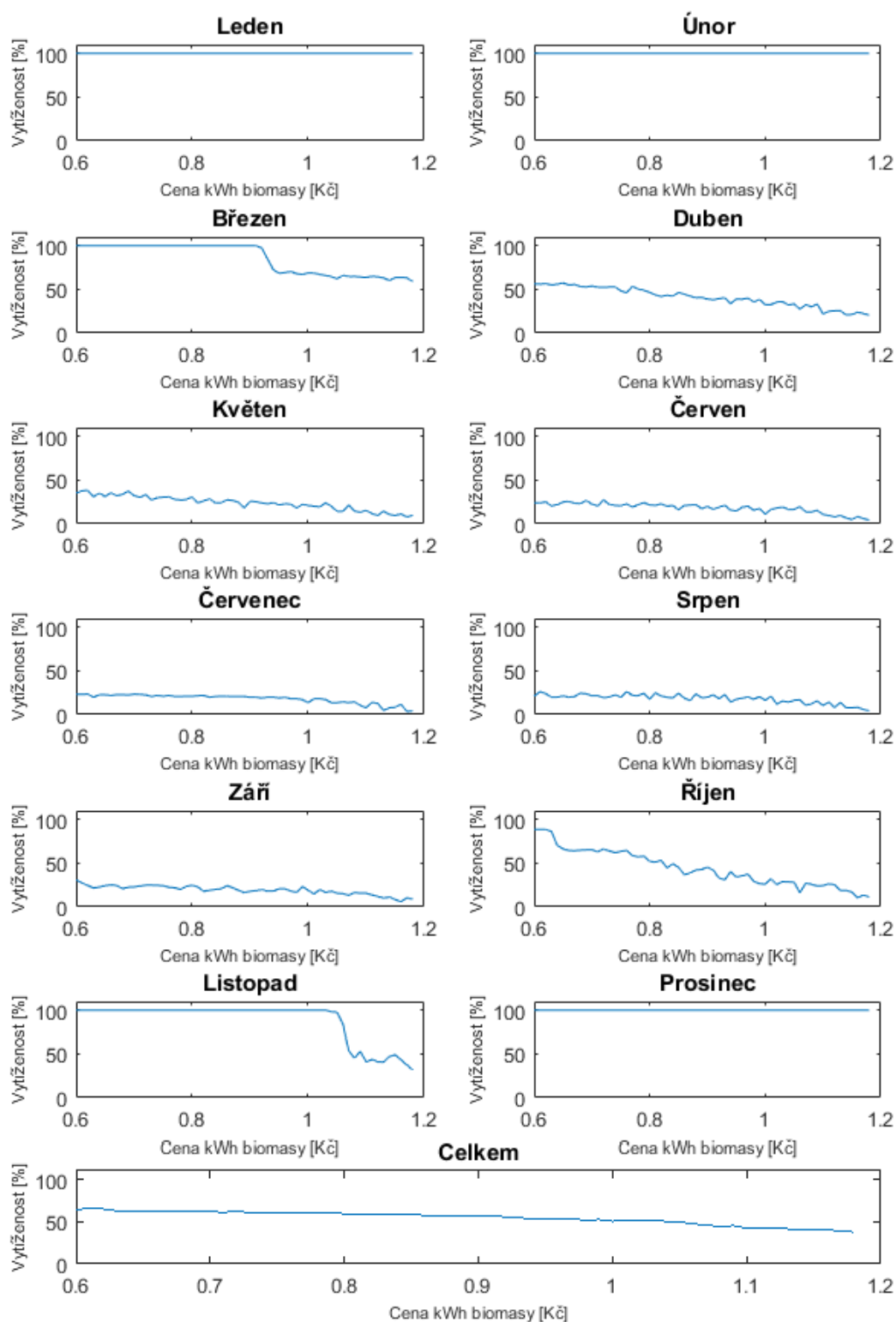
Na obrázcích 3.11, 3.12 a 3.13 je znázorněna vytíženost kotlů na biomasu v závislosti na cenách plynu a biomasy. První obrázek zachycuje růst vytíženosti kotlů na biomasu při rostoucí ceně plynu, druhý zachycuje pokles vytíženosti při rostoucí ceně biomasy. Poslední z uvedených obrázků popisuje vytíženost pro různé ceny plynu a biomasy.

V měsících prosinec, leden a únor je poptávka po teple vysoká pro každý scénář, takže kotle na biomasu jsou plně vytížené bez ohledu na ceny. Ve zbylých měsících platí již zmíněný uvedený trend, tj. při klesající ceně biomasy a rostoucí ceně plynu vytíženost roste (přesněji neklesá), dokud nedosáhne plné kapacity.

3.6. ANALÝZA CITLIVOSTI

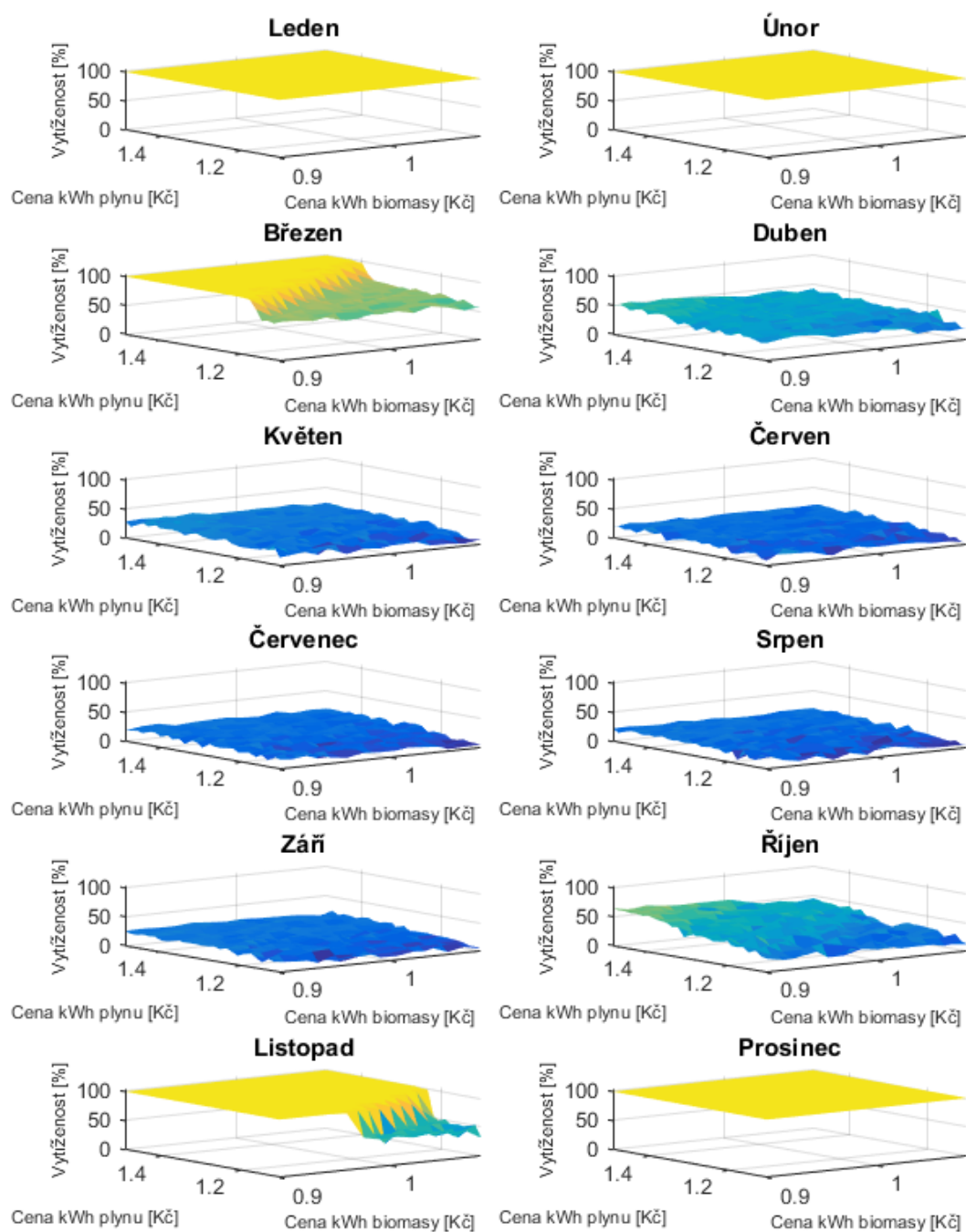


Obrázek 3.11: Vytíženost kotlů na biomasu při rostoucí ceně plynu (biomasa fixovaná na 1,0 Kč/kWh). (Vlastní tvorba.)



Obrázek 3.12: Vytíženost kotlů na biomase při rostoucí ceně biomasy (plyn fixován na 1,2 Kč/kWh). (Vlastní tvorba.)

3.6. ANALÝZA CITLIVOSTI



Obrázek 3.13: Vytíženost kotlů na biomasu. (Vlastní tvorba.)

3.7. Návrhy na vylepšení

V základním modelu kotle na biomasu produkují méně než polovinu dodaného tepla. Za předpokladu, že se očekává stále vyšší cena plynu než biomasy, bylo by rozumné rozšířit soustavu kotlů na biomasu. Dále stojí za zmínku, že Česká republika je závislá na dovozu plynu ze zahraničí. Domácí produkce dokáže pokrýt necelá 3 % spotřeby [32].

Představme si situaci, že máme původní model, do kterého chceme přidat třetí kotel na biomasu. Pokud zvolíme kotel s nízkým výkonem, najde své uplatnění v letních měsících k ohřevu vody, naopak kotel s vysokým výkonem bude vhodný při silných mrazech v zimě. Vzniká tedy přirozená otázka, jaký by měl být výkon nového kotle, aby se dosáhlo maximální finanční úspory.

Budeme zkoumat, jaká je výše finanční úspory v závislosti na výkonu nového kotle. V modelu provedeme pár změn. Množinu I značící kotle na biomasu rozšíříme

```
Set
    i                označení kotlů na biomasu                / 1 * 3 / ;
```

a obdobně jako u analýzy citlivosti přidáme blok určený k načítání parametru.

```
Parameter
    substitute                substitute hodnot                ;

$GDXIN .\MATLABtoGDX.gdx
$LOAD substitute
$GDXIN

energie('3') = substitute('energie_noveho_kotle');
```

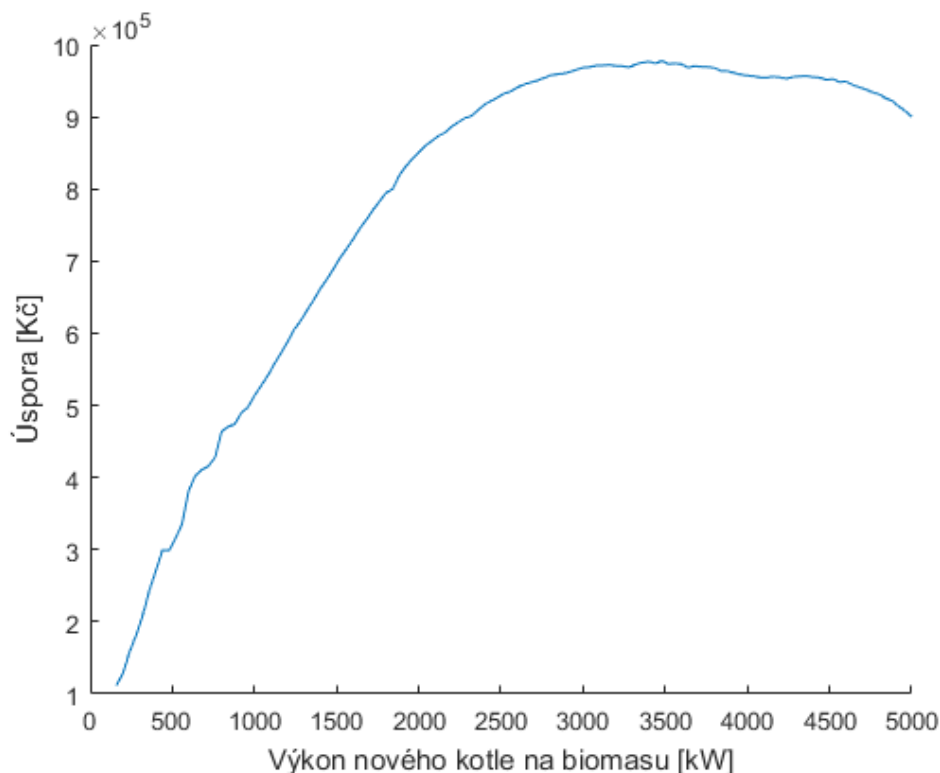
Stejně jako v minulém případě se jedná o lineární úlohu celočíselného programování (MIP), model se pouze rozrostl. Obsahuje 24 696 omezení pro 40 189 neznámých, z nichž 1 728 je binárních. Relativní přesnost nastavíme na 0,003.

```
model planovani_provozu /all/;
option MIP = gurobi;
Option optcr = 0.003 ;
solve planovani_provozu maximizing zisk using MIP;
```

Na obrázku 3.14 je znázorněna roční úspora vzniklá přidáním třetího kotle na biomasu v závislosti na jeho výkonu. Ze závislosti plyne, že nejvyšší roční úspory, téměř 1 milion Kč, lze docílit přidáním kotle o výkonu 2 500 - 5 000 kW. Pokud by byly pořizovací náklady (přepočtené na rok dle životnosti) nižší než úspora, bylo by ekonomicky výhodné kotel vybudovat.

Když už zvažujeme přestavbu teplárny, tak proč místo jednoho kotle nepostavit rovnou dva? Dle předchozích úvah by byl vhodný jeden s vyšším a druhý s nižším výkonem.

3.7. NÁVRHY NA VYLEPŠENÍ



Obrázek 3.14: Výše roční úspory vzniklé přidáním třetího kotle na biomasu v závislosti na jeho výkonu. (Vlastní tvorba.)

V modelu opět upravíme množinu I .

Set
 i označení kotlů na biomasu / 1 * 4 / ;

Výkon kotlů již není parametr, ale stává se neznámou. Aby byly zachovány původní vlastnosti, půlhodinová výše dodávky prvního (resp. druhého) kotle musí být zafixovaná na 550 (resp. 750) kWh. U třetího a čtvrtého kotle budeme požadovat, aby půlhodiny jejich provozu dodaly mezi 100 a 2 500 kWh.

Positive variables
energie(i) energie z i -tého kotle ;

energie.fx('1') = 550; energie.fx('2') = 750;
energie.lo('3') = 100; energie.up('3') = 2500;
energie.lo('4') = 100; energie.up('4') = 2500;

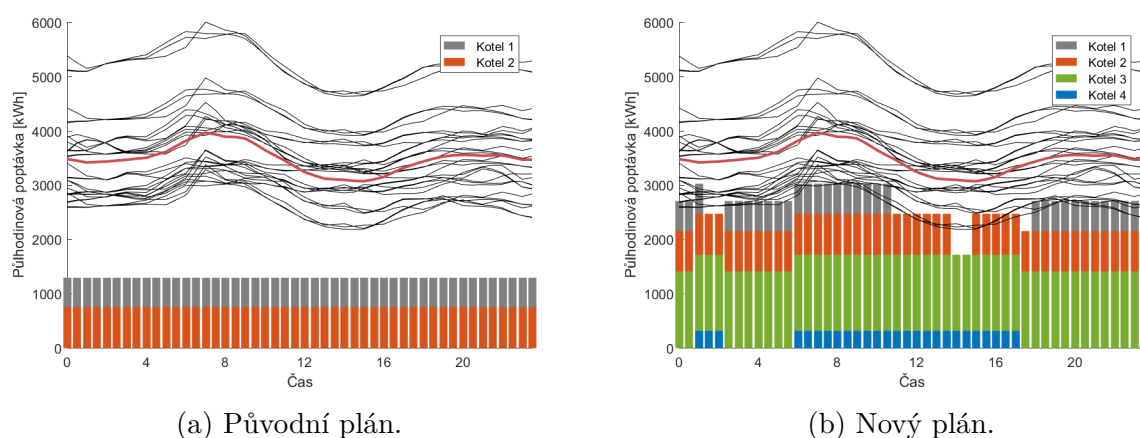
V tomto případě se již jedná o nelineární úlohu celočíselného programování, přičemž všechny nelinearity jsou kvadratické (MIQCP, *Mixed Integer Quadratically Constrained Programs*) - nelinearitu tvoří součin dodané energie kotle a proměnná značící jeho činnost. I tuto úlohu vyřešíme pomocí solveru GUROBI s relativní přesností stanovenou na 0,003.

```

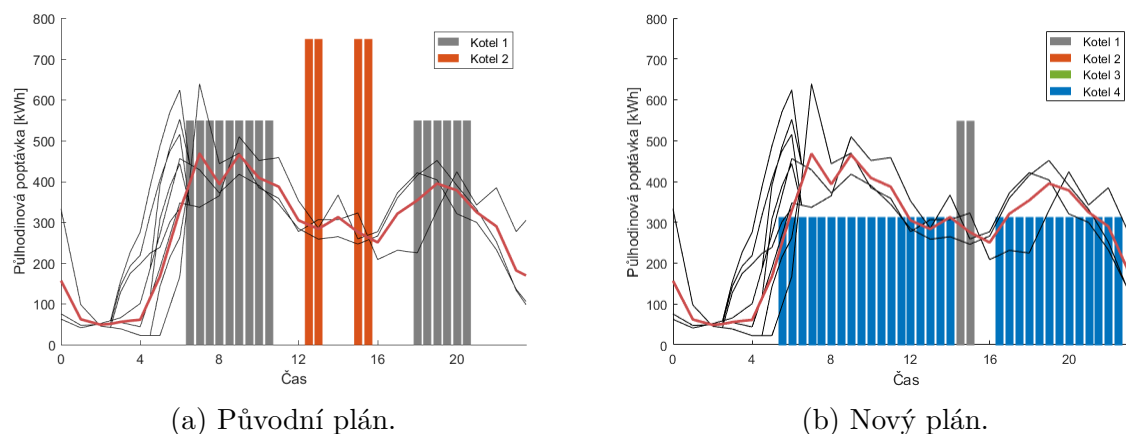
model planovani_provozu /all/;
option MIQCP = gurobi;
option optcr = 0.003;
solve planovani_provozu maximizing zisk using MIQCP;

```

Model se opět rozrostl. Skládá se ze 40 178 spojitých a 2 304 binárních proměnných, 28 116 lineárních a 17 292 kvadratických omezení. Vyřešením stanovíme výkon nových kotlů na 2 820 (resp. 630) kW. Roční úspora činí lehce nad milion korun. Na obrázku 3.17 je přepracovaný plán provozu se čtyřmi kotli na biomasu. Dle očekávání se nový kotel s vyšším výkonem (označený zeleně) používá v zimě a kotel s nižším výkonem (označený modře) pro ohřev teplé vody, viz srovnání s původním plánem 3.8. Na obrázcích 3.15 a 3.16 je porovnání plánů výroby pro měsíc leden a červenec. Všechny detailní měsíční plány jsou uvedeny v příloze.

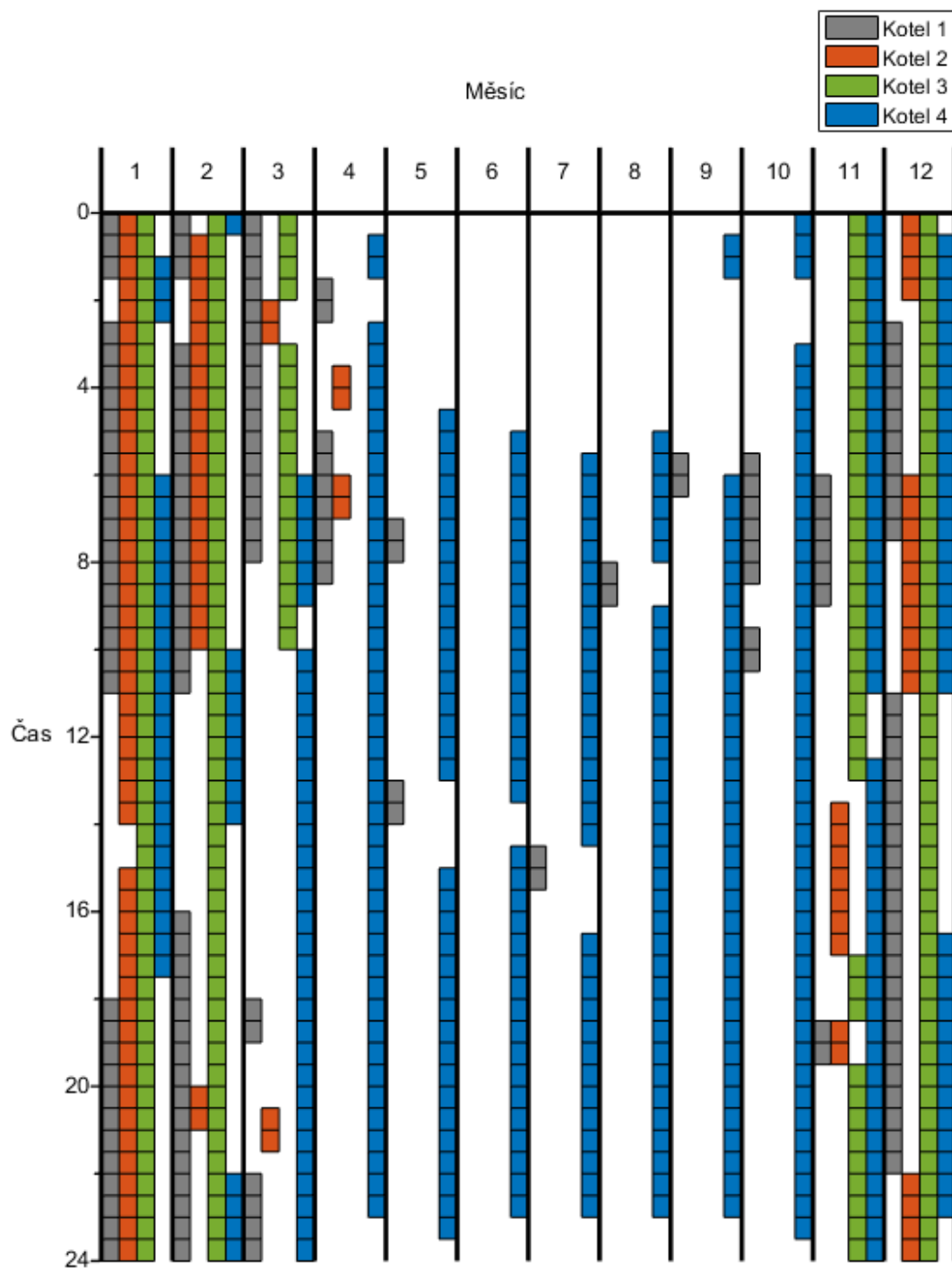


Obrázek 3.15: Plán provozu po přidání dvou kotlů na biomasu pro měsíc leden. (Vlastní tvorba.)



Obrázek 3.16: Plán provozu po přidání dvou kotlů na biomasu pro měsíc červenec. (Vlastní tvorba.)

3.7. NÁVRHY NA VYLEPŠENÍ



Obrázek 3.17: Plán provozu po přidání dvou kotlů na biomasu. (Vlastní tvorba.)

4. Závěr

Tato práce se zabývá stochastickým modelováním v oblasti energetiky. Za tímto účelem byl v první části vybudován teoretický úvod popisující matematické, lineární, nelineární a stochastické programování. V práci jsou zkoumány popisy jednotlivých typů úloh, principy řešení včetně uvedení vhodných algoritmů. U stochastického programování byl kladen důraz na popis dvoustupňové úlohy stochastické optimalizace a na scénářový přístup. Tímto byl dle autora naplněn jeden z cílů práce.

Dalším z cílů práce bylo studium, tvorba a řešení stochastických modelů, kterým je věnována druhá část. Byl úspěšně vytvořen model popisující plán výroby tepla v teplárně. Teplo v teplárně je vyráběno pomocí kotlů na biomasu a plyn, přičemž činnost kotlů na biomasu je nutné plánovat s předstihem. Jedná se o stochastický model, neboť budoucí poptávka po teple je náhodná veličina, jejíž hodnota není v době plánování známá. Model věrně zachycuje všechny fyzikální, technické, demografické i finanční aspekty související s produkcí tepla. Cílem modelu bylo vytvořit plán provozu kotlů při maximalizaci zisku. Vytvořený model se následně povedlo naprogramovat a vyřešit pomocí softwarů GAMS a MATLAB, čímž jsme získali plán provozů kotlů na biomasu.

V další části jsme se zabývali otázkou, jak by se změnila vytíženost kotlů na biomasu, při změně situace na trhu s plynem a biomasou. Z výsledků lze vyzkoušet, že při klesající ceně biomasy (resp. rostoucí ceně plynu) je ekonomicky výhodnější více využívat kotle na biomasu i při riziku ztrát způsobených nízkou poptávkou.

Poslední sekce byla věnována návrhům, jaké investice do nových kotlů na biomasu by bylo vhodné v teplárně provést, aby se v budoucnu dosahovalo vyššího zisku. Základní model byl rozšířen o jeden nový kotel na biomasu a zkoumali jsme výši roční finanční úspory v závislosti na výkonu nového kotle. Ze závislosti vyplynulo, že je vhodné postavit kotel s velmi vysokým výkonem, který bude mít své využití v zimních měsících, kdy je vysoká poptávka po teple.

Dále jsme zkoumali, zda místo přidání jednoho kotle s velmi vysokým výkonem nebude vhodnější postavit dva menší kotle. Model byl opět upraven a následně vyřešen. Z výsledku plyne, že největší úspory se dosáhne přidáním jednoho kotle s menším a druhého kotle s vyšším výkonem. Pro nově vypočtenou konfiguraci kotlů jsme vypracovali nový plán provozu kotlů na biomasu.

Práce dle názoru autora splňuje všechny body, které si kladla za cíl. Mezi její hlavní přínosy můžeme zařadit vytvořené modely, které lze použít i v budoucnu při změně vstupních dat, popřípadě je lze lehce modifikovat k řešení úloh podobného charakteru.

Diplomová práce vycházela z problematiky studované v rámci projektu ComSi („Výpočtové simulace pro efektivní nízkoemisní energetiku“ reg. č.: CZ.02.1.01/0.0/0.0/16_026/0008392 financovaného z OP VVV, Prioritní osy 1: Posilování kapacit pro kvalitní výzkum) a v práci prezentovaný originální model a dosažené výsledky budou v projektu dále využitelné.

Literatura

- [1] YANG, Xin-She. *Engineering optimization: an introduction with metaheuristic applications*. John Wiley & Sons, 2010.
- [2] BAZARAA, M. S., John J. JARVIS a Hanif D. SHERALI. *Linear programming and network flows*. 4th ed. Hoboken, N.J.: John Wiley, c2010. ISBN 04-704-6272-8.
- [3] POPELA, P. *Optimization I: sop, vo1p, opm*. Brno, 2017.
- [4] BAZARAA, M. S., Hanif D. SHERALI a C. M. SHETTY. *Nonlinear programming: theory and algorithms*. 3rd ed. Hoboken: John Wiley, 2006. ISBN 04-714-8600-0.
- [5] WOLSEY, Laurence A. *Integer Programming*. John Wiley, 1998. ISBN 0-471-28366-5.
- [6] BIRGE, John R. a Francois LOUVEAUX. *Introduction to stochastic programming*. New York: Springer, 1997. ISBN 03-879-8217-5.
- [7] POPELA, Pavel. *An object-oriented approach to multistage stochastic programming: Models and algorithms*. Memoria de tesis doctoral, UK MFF, Prague, 1998.
- [8] KALL, P., WALLACE, S. W. *Stochastic Programming*. New York: John Wiley & Sons, 1993. ISBN 978-0471951582.
- [9] WALLACE, S. W., KING, A. *Modeling with Stochastic Programming*. Springer Verlag, 2012. ISBN 978-0-387-87816-4.
- [10] RUSZCZYNSKI, A. et al. *Handbooks in Operations Research and Management Science*. vol. 10: Stochastic Programming. Amsterdam: Elsevier, 2003. ISBN 978-0-444-50854-6.
- [11] GAMS - The Modeling Language. GAMS [online]. [cit. 2022-03-11]. Dostupné z: <https://www.gams.com/products/gams/gams-language/>.
- [12] DUPAČ, Václav a Marie HUŠKOVÁ. *Pravděpodobnost a matematická statistika*. 2., upr. vyd. Praha. ISBN 978-80-246-2208-8.
- [13] TETOUR, D. *Optimalizační modely rizik v energetických systémech*. Brno, 2020. Diplomová práce. Ústav soudního inženýrství Vysokého učení technického v Brně.
- [14] *Jaký je potenciál využití biomasy v Česku a ve světě*. Internetový portál o Energetice se zaměřuje na zpravodajství a odborné články z energetického sektoru, nabídky práce a monitoring médií. [online]. 2019 [cit. 2022-03-22]. Dostupné z: <https://oenergetice.cz/nazory/jaky-potencial-vyuziti-biomasy-cesku-ve-svete>.
- [15] *Novela vyhlášky č. 78/2013 Sb. – (264/2020 Sb.); Část 3: Nastavení faktorů primární energie z neobnovitelných zdrojů*. TZB-info - Stavebnictví. Úspory energií. Technická zařízení budov. [online]. 2020 [cit. 2022-03-22]. Dostupné z: <https://www.tzb-info.cz/energeticka-narocnost-budov/20716-novela-vyhlasky-c-78-2013-sb-cast-3-nastaveni-faktoru-primarni-energie-z-neobnovitelnych-zdroju>.

- [16] *Meteostanice - Brno - Tuřany, aktuální teplota a počasí, archiv, rekordy / In-pocasi.* In-pocasi [online]. [cit. 2022-02-22]. Dostupné z: <https://www.in-pocasi.cz/archiv/brno/>.
- [17] KAUT, Michal. *Scenario tree generation for stochastic programming: Cases from finance.* 2003.
- [18] *Kmeans. MathWorks - Makers of MATLAB and Simulink* [online]. MathWorks, 2006 [cit. 2022-03-11]. Dostupné z: <https://www.mathworks.com/help/stats/kmeans.html>.
- [19] *Fyzika: vysokoškolská učebnice obecné fyziky.* 2. přepracované vydání. Brno: VUT-IUM, 2013. Překlady vysokoškolských učebnic. ISBN 978-80-214-4123-1.
- [20] *Vyhláška č. 194/2007 Sb., kterou se stanoví pravidla pro vytápění a dodávku teplé vody, měrné ukazatele spotřeby tepelné energie pro vytápění a pro přípravu teplé vody a požadavky na vybavení vnitřních tepelných zařízení budov přístroji regulujícími dodávku tepelné energie konečným spotřebitelům.* In: Sbírka zákonů. 1. 9. 2007. ISSN 1211-1244.
- [21] DRAHOKOUPILOVÁ, Lucie. *Optimalizace provozu kotelny Brno-Bystřice.* [online]. Brno, 2010. [cit. 2022-03-14]. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/26309>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Energetický ústav. Vedoucí práce Jan Fiedler.
- [22] *Stanovení potřeby teplé vody a tepla pro její přípravu a rozvod podle nové ČSN EN 15316-3.* TZB-info - Stavebnictví. Úspory energií. Technická zařízení budov. [online]. 2012 [cit. 2022-03-11]. Dostupné z: <https://voda.tzb-info.cz/priprava-teple-vody/8850-stanoveni-potreby-teple-vody-a-tepla-pro-jeji-pripravu-a-rozvod-podle-nove-csn-en-15316-3>.
- [23] *Legislativní požadavky v oblasti přípravy teplé vody.* TZB-info - Stavebnictví. Úspory energií. Technická zařízení budov. [online]. 2016 [cit. 2022-03-11]. Dostupné z: <https://voda.tzb-info.cz/priprava-teple-vody/14837-legislativni-pozadavky-v-oblasti-pripravy-teple-vody>.
- [24] *Jaké je složení domácností v ČR?* Český statistický úřad | ČSÚ [online]. 2013 [cit. 2022-03-11]. Dostupné z: https://www.czso.cz/csu/czso/jake_je_slozeni_domacnosti_v_cr20130307.
- [25] *Efficiency, Economic and Environmental Assessment of Ground Source Heat Pumps in Central Pennsylvania.* In: 2009 42nd Hawaii International Conference on System Sciences [online]. IEEE, 2009, s. 1-7 [cit. 2022-03-11]. ISBN 978-0-7695-3450-3. Dostupné z: doi:10.1109/HICSS.2009.179.
- [26] LUTZ, J., et al. *Hot Water Draw Patterns: Findings from Field Studies.* Lawrence Berkeley National Laboratory. Berkeley, CA, 2006. 10.13140/RG.2.1.2881.4888.
- [27] *Cena tepla / Teplárny.* Teplárny [online]. 2021 [cit. 2022-03-11]. Dostupné z: <https://www.teplarny.cz/cs/cena-tepla>.

LITERATURA

- [28] GAMS. GAMS 28.2 [software]. 19. února 2019 [cit. 2022-03-11]. Dostupné z: <https://www.gams.com/28/>.
- [29] MATHWORKS. MATLAB 2016a [software]. 21. března 2016 [cit. 2022-03-11].
- [30] Microsoft. Microsoft Office 2007 [software]. 30. ledna 2007 [cit. 2022-03-11].
- [31] *GAMS and Matlab - Optimization with GAMS*. GAMS and Matlab - Optimization with GAMS [online]. 2017 [cit. 2022-03-11]. Dostupné z: <https://www.gamsoptimization.com/gams-and-matlab/>.
- [32] *Statistická ročenka České republiky - 2020*. Český statistický úřad | ČSÚ [online]. 2020 [cit. 2022-03-19]. Dostupné z: <https://www.czso.cz/csu/czso/16-energetika-cy3v64b9vw>.
- [33] *GAMS and Matlab Setup and Introduction to GDXMRW*. In: Youtube [online]. 27.02.2007 [cit. 2022-03-30]. Dostupné z: https://www.youtube.com/watch?v=HPn_q8nlktE. Kanál uživatele GAMS Lessons.

6. Seznam použitých zkratek a symbolů

Zkratky:

HN	here-and-now přístup
IP	celočíselný problém
LP	lineární programování
MIP	Celočíselné lineární programování / Mixed Integer Programming
MIQCP	Celočíselné programování s kvadratickými omezeními / Mixed Integer Quadratically Constrained Programs
MP	matematické programování
NLP	nelineární programování
RP	relaxovaný problém
TUV	teplá užitková voda
WS	wait-and-see přístup

Symboly v teoretické části:

$\mathbf{a}, \mathbf{b}, \mathbf{c}$	sloupcový vektor
$\mathbf{a}^\top, \mathbf{b}^\top, \mathbf{c}^\top$	řádkový vektor
$\mathbf{A}, \mathbf{B}, \mathbf{C}$	matice
$\mathbf{A}^{-1}, \mathbf{B}^{-1}, \mathbf{C}^{-1}$	inverzní matice
arglocmin	množina všech bodů, ve kterém funkce nabývá lokálního minima
argglobmin	množina všech bodů, ve kterém funkce nabývá globálního minima
argmax	množina všech bodů, ve kterém funkce nabývá maxima
argmin	množina všech bodů, ve kterém funkce nabývá minima
\mathbf{B}	bázová matice
\mathbf{c}	sloupcový vektor cenových koeficientů
\mathbf{d}	směr
D	množina optimálních řešení
E	střední hodnota

$f(\mathbf{x})$	účelová funkce
$g(\mathbf{x})$	omezení zadané rovnicí
$h(\mathbf{x})$	omezení zadané nerovnicí
J, J_0, J_1, J_2	množina indexů celočíselných neznámých
J_B	množina indexů sloupců tvořící bazickou matici
J_N	množina indexů sloupců tvořící nebazickou matici
\mathbb{N}	množina přirozených čísel $\{1, 2, \dots\}$
$O(\mathbf{x})$	okolí bodu \mathbf{x}
p_s	pravděpodobnost scénáře s
\mathbb{R}	množina reálných čísel
\mathbb{R}^n	reálný vektorový prostor dimenze n
S	množina přípustných řešení / označení scénářů
$t(\mathbf{x})$	účelová funkce relaxované úlohy
T	množina přípustných řešení relaxované úlohy
\mathbf{T}	matice technologií ve dvoustupňové úloze
\mathbf{W}	kompensační matice ve dvoustupňové úloze
\mathbf{x}_B	sloupcový vektor bázových neznámých
\mathbf{x}_N	sloupcový vektor nebázových neznámých
\mathbf{x}_{min}	optimální řešení
ξ	náhodný vektor
\mathbf{y}	rozhodnutí druhého stupně
z	minimální hodnota účelové funkce
\underline{z}	dolní odhad z
\bar{z}	horní odhad z
z^R	minimální hodnota relaxované účelové funkce
\mathbb{Z}	množina celých čísel
$ \cdot $	absolutní hodnota čísla / kardinalita množiny
$\ \cdot\ $	Euklidovská norma
$\lfloor \cdot \rfloor$	dolní celá část
$\lceil \cdot \rceil$	horní celá část

Symbole použité v modelu:

c	měrná tepelná kapacita
cb	cena 1 kWh vyrobené pomocí biomasy
cg	prodejní cena 1 GJ
cn	celkové roční náklady
cp	cena 1 kWh vyrobené pomocí plynu
ct	celkové roční tržby
d	počet domácností
dT	rozdíl teplot
dxm_{mit}	označení vypnutí kotle
dxp_{mit}	označení zapnutí kotle
e_i	energie získána z kotle na biomasu
ch	koeficient chlazení tepelného zásobníku
i	označení kotlů na biomasu
I	množina indexů i
j	označení časové řady
J_m	počet časových řad
J	množina indexů j
k	koeficient vytápění
m	označení měsíců
M	množina indexů m
md_m	počet dní v měsíci
mn_m	měsíční náklady
mt_m	měsíční tržby
n_{ms}	počet časových řad
nb_m	denní náklady na biomasu
nc_{msr}	celkové denní náklady
np_{msr}	denní náklady na plyn

o	počet osob na domácnost
om	omezení počtu sepnutí
pr_r	pravděpodobnost scénáře r
ps_{ms}	pravděpodobnost scénáře s
Q_{msrt}	poptávané teplo
Qt_{mst}	teplo na vytápění
Qv_{rt}	teplo na ohřev vody
r	označení scénářů pro poptávku vody
R	množina indexů r
s	označení scénářů pro průběh teploty
S	množina indexů s
t	označení půlhodinových intervalů
T	množina indexů t
Te_{mst}	teplota - scénářové hodnoty
TE_{mjt}	teplota - původní data
tr_{msr}	denní tržba
u_{mjs}	příslušnost časové řady v třídě
v_{rt}	normovaná spotřeba teplé vody
vc	denní spotřeba teplé vody
x_{mit}	stav kotle na biomasu
y_{mtsr}	energie z plynových kotlů
z	zisk
za_{mtsr}	stav tepelného zásobníku
zp_m	stav tepelného zásobníku o půlnoci

7. Seznam příloh

Tištěná příloha

- Příloha č. 1 - Model.
- Příloha č. 2 - Plán provozu kotlů na biomasu pro jednotlivé měsíce.
- Příloha č. 3 - Plán provozu po přidání dvou kotlů na biomasu pro jednotlivé měsíce.

Digitální příloha

- Knihovna pro čtení GDX souborů - složka read_gdx
- Základní model - složka model
- Analýza citlivosti - složka analyza_citlivosti
- Návrh třetího kotle - složka tri_kotle
- Návrh dvou nových kotlů - složka ctyri_kotle

Informace pro čtenáře pracující s digitální přílohou

- Druhá až poslední uvedená složka vždy obsahuje zdrojová data, GAMS kód a MATLAB kód.
- GAMS kódy lze spustit samostatně. Výsledné řešení se uloží do GDX souboru.
- Aby bylo možné ovládat GAMS pomocí MATLABu, je nutné při instalaci GAMSu zaškrtnout „Use advanced installation mode“ a „Add GAMS directory to PATH environment variable“. Pak v MATLABu mezi PATH přidat složku, ve které je GAMS nainstalován. Detailní návod je na uveden ve videu [33].
- Na běžném zařízení zaberou všechny výpočty přibližně 5 dní.

8. Přílohy

Příloha č. 1 - Model

$$\min z$$

$$\text{za podmínek } Qt_{mst} = \max\{13 - Te_{mst} ; 0\} \cdot d \cdot k ,$$

$$Qv_{rt} = v_{rt} \cdot vc \cdot d \cdot o \cdot dT \cdot 4,18/3600 ,$$

$$Q_{msrt} = Qt_{mst} + Qv_{rt} ,$$

$$Q_{msrt} \leq y_{mtsr} + \sum_{i \in I} (x_{mit} \cdot e_i) + za_{m(t-1)sr} \cdot ch - za_{mtsr} \quad , \forall t > 1 ,$$

$$Q_{msr1} \leq y_{m1sr} + \sum_{i \in I} (x_{mi1} \cdot e_i) + za_{m48sr} \cdot ch - za_{m1sr} ,$$

$$za_{m1sr} = zp_m ,$$

$$za_{m48sr} = zp_m ,$$

$$dxp_{mit} - dxm_{mit} = x_{mit} - x_{mi(t-1)} \quad , \forall t > 1 ,$$

$$\sum_{t \in T} (dxp_{mit} + dxm_{mit}) \leq om ,$$

$$dxp_{mi(t-1)} + dxm_{mit} \leq 1 \quad , \forall t > 1 ,$$

$$dxp_{mit} + dxm_{mi(t-1)} \leq 1 \quad , \forall t > 1 ,$$

$$np_{msr} = \sum_{t \in T} y_{mtsr} \cdot cp ,$$

$$nb_m = \sum_{t \in T} \sum_{i \in I} (x_{mit} \cdot e_i) \cdot cp ,$$

$$nc_{msr} = np_{msr} + nb_m ,$$

$$tr_{msr} = \sum_{t \in T} Q_{msrt} \cdot cg/277,77 ,$$

$$mn_m = md_m \cdot \sum_{s \in S} \sum_{r \in R} (nc_{msr} \cdot ps_{ms} \cdot pr_r) ,$$

$$mt_m = md_m \cdot \sum_{s \in S} \sum_{r \in R} (tr_{msr} \cdot ps_{ms} \cdot pr_r) ,$$

$$cn = \sum_{m \in M} mn_m ,$$

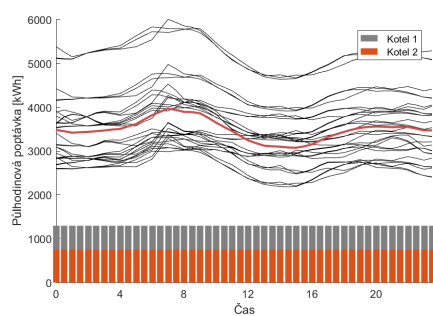
$$ct = \sum_{m \in M} mt_m ,$$

$$z = ct - cn ,$$

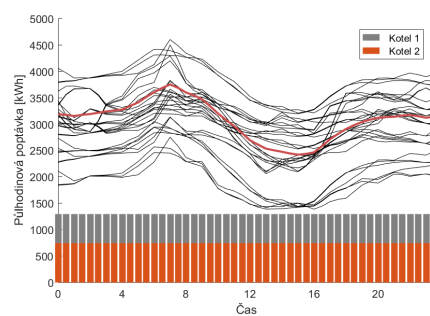
pokračování na další stránce...

$$\begin{aligned}
& y_{mtsr} \geq 0, \\
& za_{mtsr} \geq 0, \\
& zp_m \geq 0, \\
& dxp_{mit} \geq 0 \\
& dxm_{mit} \geq 0 \\
& y_{mtsr} \leq 9300 \\
& za_{mtsr} \leq 2000 \\
& np_{msr} \geq 0, \\
& nb_m \geq 0, \\
& nc_{msr} \geq 0, \\
& tr_{msr} \geq 0, \\
& mn_m \geq 0, \\
& mt_m \geq 0, \\
& cn \geq 0, \\
& ct \geq 0, \\
& x_{mit} \in \{0, 1\}, \\
& i \in \{1, 2\} = I, \\
& t \in \{1, 2, \dots, 48\} = T, \\
& s \in \{1, 2, \dots, 10\} = S, \\
& r \in \{1, 2, 3\} = R, \\
& m \in \{1, 2, \dots, 12\} = M.
\end{aligned}$$

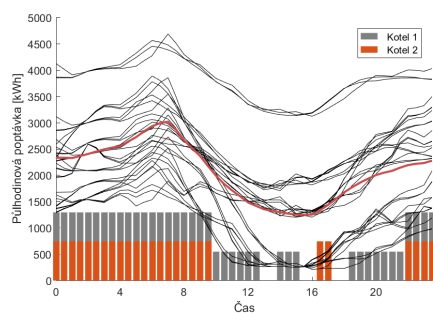
Příloha č. 2 - Plán provozu kotlů na biomasu pro jednotlivé měsíce.



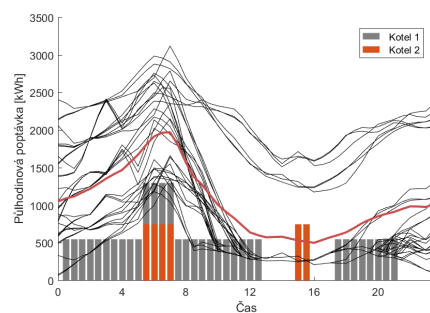
(a) Leden.



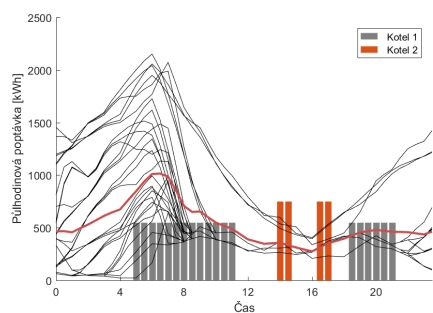
(b) Únor.



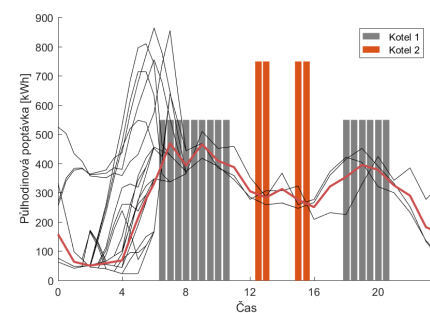
(c) Březen.



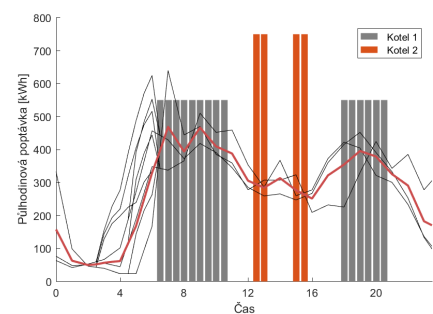
(d) Duben.



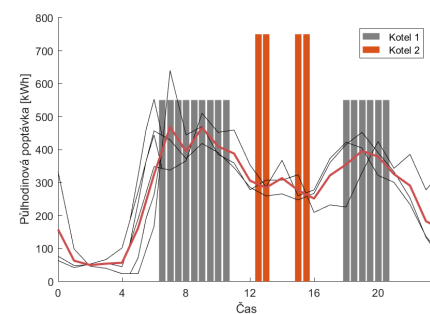
(e) Květen.



(f) Červen.

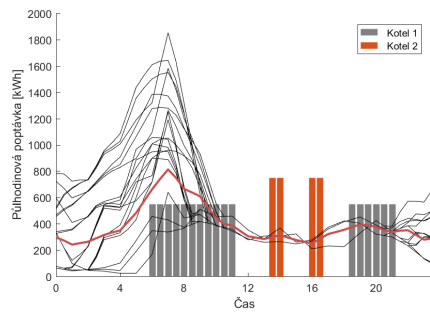


(g) Červenec.

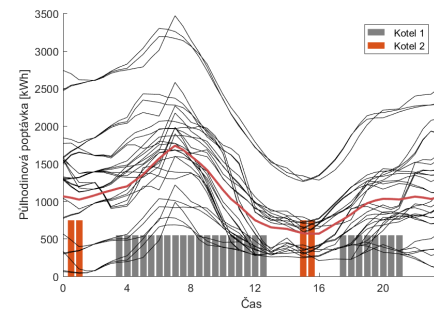


(h) Srpen.

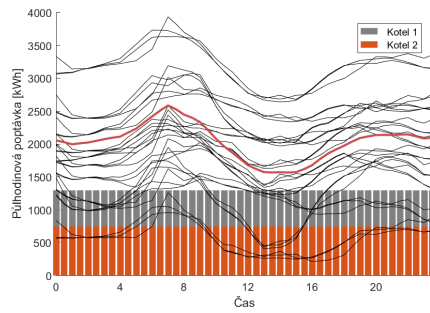
Obrázek 8.1: Plán provozu kotlů na biomasu pro jednotlivé měsíce (1. část). (Vlastní tvorba.)



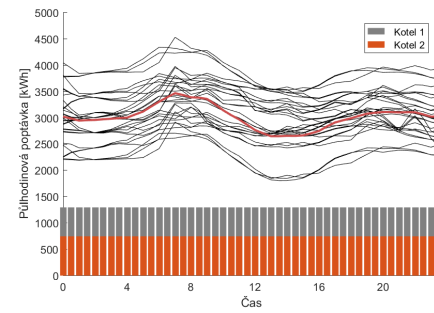
(a) Září.



(b) Říjen.



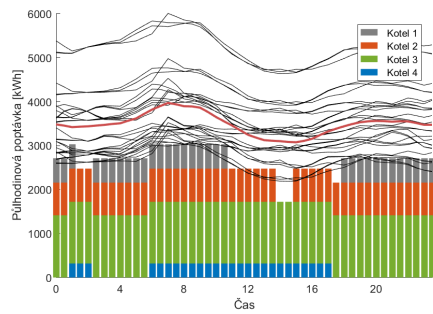
(c) Listopad.



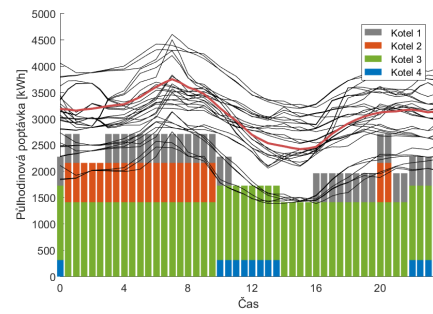
(d) Prosinec.

Obrázek 8.2: Plán provozu kotlů na biomasu pro jednotlivé měsíce (2. část). (Vlastní tvorba.)

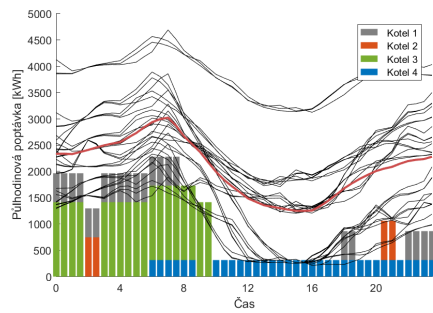
Příloha č. 3 - Plán provozu po přidání dvou kotlů na biomasu pro jednotlivé měsíce.



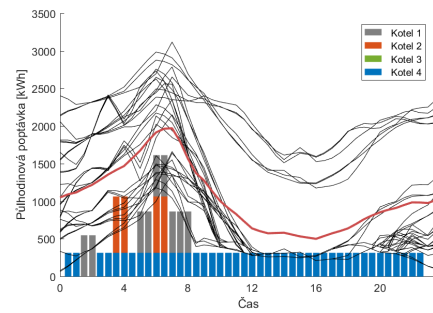
(a) Leden.



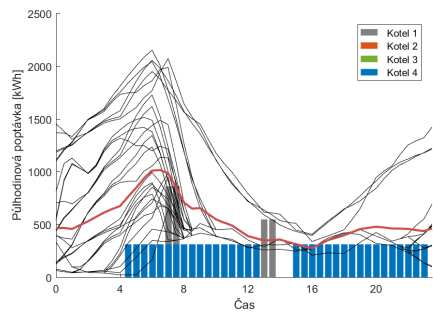
(b) Únor.



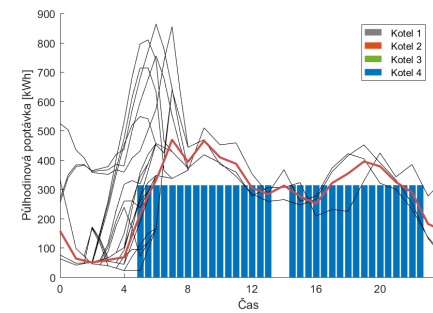
(c) Březen.



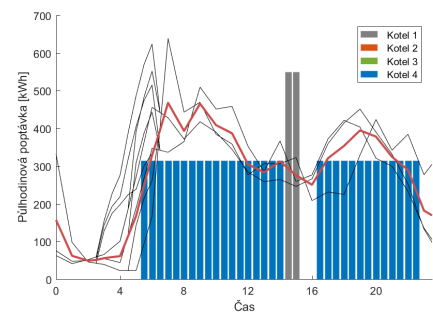
(d) Duben.



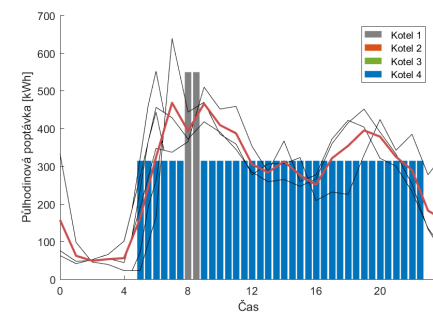
(e) Květen.



(f) Červen.

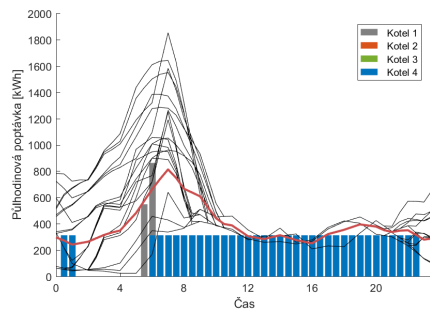


(g) Červenec.

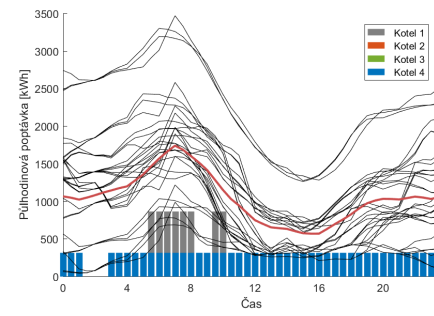


(h) Srpen.

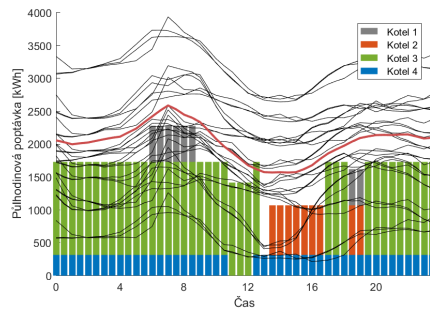
Obrázek 8.3: Plán provozu po přidání dvou kotlů na biomasu pro jednotlivé měsíce (1. část). (Vlastní tvorba.)



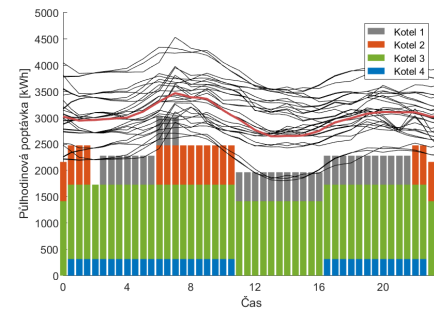
(a) Září.



(b) Říjen.



(c) Listopad.



(d) Prosinec.

Obrázek 8.4: Plán provozu po přidání dvou kotlů na biomasu pro jednotlivé měsíce (2. část). (Vlastní tvorba.)