

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA ELEKTROTECHNICKÁ



**Vývoj algoritmů pro řízení spotřeby  
v sítích smart-grids**

**Development of algorithms for demand-side  
management in smart-grids**

Praha, 2013

Autor: Bc. Josef Müller



## ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Josef Müller**

Studijní program: Kybernetika a robotika  
Obor: Systémy a řízení

Název tématu: **Vývoj algoritmů pro řízení spotřeby v sítích smart-grids**

Pokyny pro vypracování:

1. Seznamte se parametry a možnostmi platformy ARM, pro kterou budou algoritmy navrhovány.
2. Porovnejte dostupné vývojové desky s procesorem ARMv8 a ARMv9 a vyberte vhodnou desku pro vývoj algoritmů.
3. Najděte LP/MILP/QP solvery dostupné pro ARM platformu a porovnejte je z hlediska výkonnosti.
4. Navrhněte novou implementaci algoritmu řídicího systému dodaného zadavatelem s ohledem na omezený výkon ARM platformy.

Seznam odborné literatury:

- [1] J. Kondoh, N. Lu, and D. Hammerstrom, An evaluation of the water heater load potential for providing regulation service, IEEE Trans. Power Syst., vol. 26, no. 3, pp. 1309–1316, Aug. 2011.
- [2] A. Neuberg, Ripple control in the Czech Republic and Demand Side Management, in 20th Int. Conf. and Exhibition on Electricity Distribution - Part 1, 2009. CIRED 2009., 2009, pp. 1–5.
- [3] I. Knight, N. Kreutzer, M. Manning, M. Swinton, and H. Ribberink, Residential cogeneration systems: European and Canadian residential non-HVAC electric and DHW load profiles for use in simulating the performance of residential cogeneration systems, Annex 42 report, IEA ECBCS, Tech. Rep., 2007. [Online]. Available: [http://www.ecbcs.org/docs/Annex 42 Domestic Energy Profiles.pdf](http://www.ecbcs.org/docs/Annex_42_Domestic_Energy_Profiles.pdf)

Vedoucí: Ing. Ondřej Malík

Platnost zadání: do konce letního semestru 2013/2014

prof. Ing. Michael Šebek, DrSc.  
vedoucí katedry



prof. Ing. Pavel Ripka, CSc.  
děkan

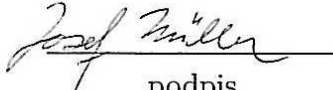
V Praze dne 18. 2. 2013



## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, dne 10.5.2013

  
podpis



## **Poděkování**

Děkuji především Ing. Ondřeji Malíkovi a Ing. Ondřeji Novákovi za pomoc, strávený čas a cenné rady při realizaci této diplomové práce.

## Abstrakt

Cílem této práce bylo posoudit možnost provozu existujícího algoritmu [12] v embedded zařízeních založených na platformě ARM a případně provést pro to potřebné úpravy. Výše uvedený algoritmus se zabývá řízením spotřeby bojlerů v domácnostech s cílem odstranit problémy s obnovitelnými zdroji elektrické energie v oblastech nízkého napětí.

Úkolem práce bylo se seznámit s parametry a možnostmi dostupných vývojových desek s procesorem ARM, porovnat tyto desky z hlediska zadaných kritérií a vybrat vhodnou desku pro implementaci řídicího algoritmu. Tento algoritmus využívá optimalizace, kterou provádí ve dvou fázích. V jedné fázi se jedná o optimalizaci typu lineární programování (LP) a ve druhé fázi typu smíšeného celočíselného lineárního programování (MILP). Dalším úkolem tedy bylo nalézt pro vybranou vývojovou desku vhodný, z cenových důvodů nekomerční, solver řešící tyto typy optimalizačních problémů. Protože ale nekomerční solvery na vybrané desce nenalezly řešení MILP problémů v dostatečně krátkém čase, bylo dalším úkolem práce vyvinout novou implementaci této části řídicího algoritmu. Vznikl tedy heuristický algoritmus řešící problém ve značně rychlejším čase a jím nalezené řešení je podobné, v jistých ohledech i lepší, než řešení získané optimalizací.



## Abstract

The aim of this thesis was to evaluate the possibility of using the existing algorithm [12] in embedded devices based on ARM and make the necessary adjustments. The algorithm deals with optimizing the charging periods of water heaters in households in order to eliminate problems with renewable energy sources in low voltage areas.

The first task was to get acquainted with the parameters and options of available development boards with ARM processor, compare these boards in terms of specified criteria and select the appropriate board for implementation of the control algorithm. This algorithm uses a two-phase optimization. The type of the optimization in the first phase is linear programming (LP) and in the second phase it is mixed integer linear programming (MILP). Therefore, the second task was to find for the selected development board a suitable non-commercial solver for these types of optimization problems. However, since the non-commercial solvers on the selected board could not find solution of MILP problems in a sufficiently short time, the final task of this thesis was to develop a new implementation of this part of the control algorithm. Thus was created a heuristic algorithm which solves the problem in a considerably shorter time and it provides a solution of similar quality, in some ways even better than the solution obtained by the original MILP optimization algorithm.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>12</b>
<b>2</b>	<b>Možnosti platformy ARM a porovnání</b>	<b>14</b>
2.1	Úvod . . . . .	14
2.2	Platforma ARM . . . . .	15
2.3	Porovnání dostupných vývojových desek . . . . .	17
<b>3</b>	<b>Solvery dostupné pro platformu ARM a porovnání</b>	<b>19</b>
3.1	Úvod . . . . .	19
3.2	Porovnání solverů . . . . .	20
<b>4</b>	<b>Heuristický algoritmus pro optimalizaci výkonové bilance</b>	<b>24</b>
4.1	Úvod . . . . .	24
4.2	Popis heuristického algoritmu . . . . .	25
4.3	Nastavení parametrů heuristického algoritmu . . . . .	29
4.4	Simulace s navrženým systémem . . . . .	33
<b>5</b>	<b>Závěr</b>	<b>37</b>



# Kapitola 1

## Úvod

Tato práce se zabývá řízením spotřeby elektrické energie v domácnostech. Zaměřuje se na oblasti, kde díky zvyšujícímu se množství lokálních fotovoltaických zdrojů dochází k výrobě elektrické energie v době nejnižší spotřeby a naopak k její nejvyšší spotřebě v době nejnižší výroby. Důsledkem je, že je energii třeba v jeden okamžik transportovat z oblasti a v jiný naopak do oblasti. To má za následek vysoké zatěžování napájecího vedení a ztráty elektrické energie při transportu. Motivací práce je snížení negativního dopadu lokální výroby elektrické energie v řízené oblasti.

Práce navazuje na dříve vytvořenou práci [12], jenž navrhuje řídicí systém, který využívá bojlerů v domácnostech k ukládání elektrické energie z fotovoltaických elektráren v době její produkce. Vhodným plánováním ohřevu vody v bojlerech se docílí snížení špiček výkonové bilance oblasti a celkového zmenšení množství elektrické energie transportované z a do oblasti. Tato práce upravuje výše zmíněný řídicí systém, tak aby ho bylo v budoucnu možné využít v praxi, tedy aby algoritmus fungoval dostatečně rychle na počítači s ARM procesorem, jenž je levný a úsporný.

Algoritmus popsany v článku [12] je implementován jako dvoufázový optimalizační problém. První fáze je zjednodušený problém lineárního programování (LP). Druhá fáze je složitější problém smíšeného celočíselného lineárního programování (MILP). Obě fáze hledají optimální rozložení příkonu bojlerů, tak aby se minimalizovaly špičky importu a exportu energie v oblasti. První fáze pracuje se zjednodušeným lineárním modelem boj-

lerů, ale s delším časovým horizontem (2 dny), tak aby bylo možné zahrnout předpovídaný vývoj počasí. První fáze se provádí pouze dvakrát denně a to jen v případě, že je k dispozici nová předpověď počasí. Druhá fáze pracuje s MILP modelem bojlerů pro přesné naplánování jejich provozu na kratším horizontu. Kritériem je co nejvíce se přiblížit optimální výkonové bilanci z první fáze, dále označované jako reference, při splnění omezení, kterým je dostatek teplé vody pro využití v domácnostech a zamezení přílišného přepínání bojlerů. Druhá fáze se provádí oproti první fázi častěji, v testovaném případě každých 10 minut, a tak bylo zapotřebí zajistit nalezení kvalitního řešení v krátkém čase.

Prvním z úkolů této práce, jenž je popsán v kapitole 2, bylo seznámení se s parametry a možnostmi platformy ARM, porovnání dostupných vývojových desek na bázi této platformy a výběr vhodné desky pro vývoj algoritmů. Cílem je, aby byl v budoucnu na této desce implementován výše zmíněný systém řízení ohřevu vody v bojlerech.

Dalším úkolem, kterým se zabývá tato práce v kapitole 3, je výběr optimalizačního solveru, jenž je:

- typu smíšeného celočíselného lineárního programování,
- z cenových důvodů nekomerční,
- funkční na vybrané desce s procesorem ARM,
- dostatečně výkonný a rychlý, aby vyřešil úlohu v řádu minut.

Tento vybraný solver by měl řešit optimalizační problém výše zmíněného systému řízení ohřevu vody.

Protože se ale nepodařilo nalézt nekomerční solver, který by řešil tento optimalizační problém v uspokojivě krátkém čase, bylo jedním z úkolů práce vytvořit heuristický algoritmus nahrazující část řídicího systému, viz kapitola 4. Vytvořený heuristický algoritmus snižuje paměťové a výkonové nároky výsledného systému. V závěru práce jsou uvedeny výsledky simulací provedených na původním řídicím systému [12] a na systému s heuristickým algoritmem.

# Kapitola 2

## Možnosti platformy ARM a porovnání

### 2.1 Úvod

Tato kapitola obsahuje přehled základních vlastností platformy ARM a jejích dostupných architektur. Dále jsou zde uvedeny nejvhodnější dostupné jednodeskové počítače pro implementaci navržených algoritmů.

## 2.2 Platforma ARM

Procesory ARM využívají redukovanou instrukční sadu (RISC), což umožňuje zrychlení práce využitím více jednoduchých instrukcí. Výhodou této architektury je nízká cena, menší zahřívání a menší spotřeba energie než u procesorů obsažených v počítačích PC, viz [1]. Redukovaná složitost a jednodušší design umožnily vytvořit nízkoenergetické systémy na jednom čipu pro vestavěné (embedded) systémy. Více informací o ARM platformě je v článku [2].

V tabulce 2.1 je vidět přehled jednotlivých vyvíjených architektur ARM procesorů a jim příslušících rodin. V současné době se dále vyvíjí architektury ARMv7 a ARMv8. A právě na architekturu ARMv7, zejména na procesory typu ARM Cortex-A8 a ARM Cortex-A9 se zaměřuje tato práce. Jsou totiž obsaženy v dostupných jednodeskových počítačích.

Rozdíly mezi procesory typu ARM Cortex-A8 a ARM Cortex-A9 jsou obsaženy

Architektura	Rodina
ARMv1	ARM1
ARMv2	ARM2, ARM3
ARMv3	ARM6, ARM7
ARMv4	StrongARM, ARM7TDMI, ARM8, ARM9TDMI
ARMv5	ARM7EJ, ARM9E, ARM10E, XScale
ARMv6	ARM11
ARMv6-M	ARM Cortex-M0, ARM Cortex-M1
ARMv7	ARM Cortex-A5, ARM Cortex-A7, <b>ARM Cortex-A8, ARM Cortex-A9,</b> ARM Cortex-A15
ARMv7-M	ARM Cortex-M3, ARM Cortex-M4
ARMv8-A	ARM Cortex-A53, ARM Cortex-A57

Tabulka 2.1: Tabulka architektur ARM procesorů

v článku [3]. Hlavní rozdílem je, že typ ARM Cortex-A9 umožňuje mít až čtyři jádra oproti ARM Cortex-A8, který existuje pouze v jedno jádrové verzi. To umožňuje typu ARM Cortex-A9 dosahovat vyšších rychlostí výpočtu.



## 2.3 Porovnání dostupných vývojových desek

Existuje mnoho jednodeskových počítačů s procesorem ARM, viz seznam uvedený v [4]. Hlavním kritériem výběru byla operační paměť alespoň 512 MB, možnost připojení přes Ethernet rozhraní, možnost použití linuxového operačního systému a podpora výrobce nebo komunitou.

Tabulka 2.2 obsahuje nejvhodnější kandidáty s jejich parametry. Jak je vidět, všichni kandidáti mají frekvenci jádra minimálně 1 GHz a obsahují HDMI výstup, čímž umožňují jednodušší práci po připojení displeje. Paměť prostřednictvím SD karet je samozřejmostí, obdobně jako USB porty pro připojení vstupních, případně výstupních zařízení.

Nakonec byli vybráni nejrozšířenější kandidáti, u typu ARM Cortex-A8 byl vybrán

Název	BeagleBoard-xM	Hackberry	IGEPv2	SBC8530	PandaBoard	PandaBoard ES	Trim-Slice
Typ jádra	A8	A8	A8	A8	A9	A9	A9
Počet jader	1	1	1	1	2	2	2
Frekvence jádra	1 GHz	1,2 GHz	1 GHz	1 GHz	1 GHz	1,2 GHz	1 GHz
Operační paměť	512 MB	1 GB	512 MB	512 MB	1 GB	1 GB	1 GB
Video výstupy							
HDMI	Ano	Ano	Ano	Ano	Ano	Ano	Ano
S-Video	Ano	Ne	Ne	Ano	Ne	Ne	Ne
Připojení k síti							
Ethernet	Ano	Ano	Ano	Ano	Ano	Ano	Ano
WiFi	Ne	Ano	Ano	Ano	Ano	Ano	Ano
Bluetooth	Ne	Ne	Ano	Ano	Ano	Ano	Ano
Úložiště							
Integrované	Ne	4 GB	512 MB	512 MB	Ne	Ne	Ne
SD karty	Ano	Ano	Ano	Ano	Ano	Ano	Ano
eSATA	Ne	Ne	Ne	Ne	Ne	Ne	Ano
USB porty							
USB	4	2	2	2	3	3	4
MicroUSB	Ne	Ne	Ne	Ne	Ne	Ne	Ano

Tabulka 2.2: Tabulka vybraných jedno deskových počítačů

BeagleBoard-xM a s typem ARM Cortex-A9 byl vybrán PandaBoard ES. Obě desky mají díky své rozšířenosti velkou podporu ze strany komunity. Více informací o těchto deskách

je dostupných v jejich manuálech, viz [5] a [6]. Další testy, zejména výběr optimalizačního solveru, viz kapitola 3, je prováděn na jednodeskovém počítači PandaBoard ES.

# Kapitola 3

## Solvery dostupné pro platformu

## ARM a porovnání

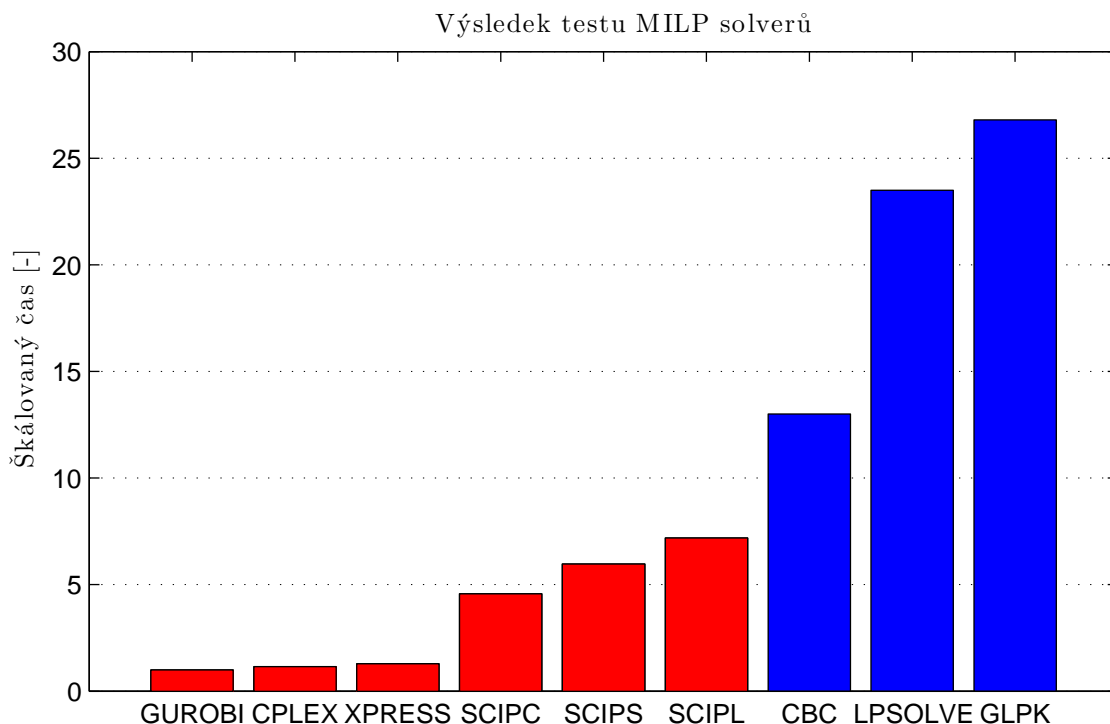
### 3.1 Úvod

Dalším úkolem práce bylo provést průzkum mezi dostupnými nekomerčními solvery a zjistit, zda je možné je využít na platformě ARM. Protože cílem dvou fází řídicího algoritmu je najít řešení smíšeného celočíselného lineárního problému (MILP) a lineárního problému (LP), hlavní pozornost byla věnována právě těmto typům solverů. Bylo provedeno jejich spuštění a porovnání výkonu na této platformě, konkrétně na vývojové desce vybrané v kapitole 2. Cílem bylo vybrat solver, který nejrychleji řeší optimalizační problém generovaný algoritmem pro řízení ohřevu vody bojlerů [12].

## 3.2 Porovnání solverů

Ukázky solverů a jejich porovnání jsou uvedeny v [7]. Příkladem je test s názvem *Mixed Integer Linear Programming Benchmark (MIPLIB2010)* z 29. 3. 2013, kde solvery měly za úkol najít optimální řešení v co nejrychlejší čas u 87 problémů. Škálované výsledky jsou na obrázku 3.1, kde čas nejrychlejšího solveru je položen roven 1 a u ostatních solverů je uveden odpovídající násobek tohoto času. Červeně jsou vyznačeny komerční a modře nekomerční solvery.

Jedná se o komerční solvery Gurobi, Cplex, Xpress a SCIP s lineárními solvery Cplex



Obrázek 3.1: Výsledky testu MILP solverů

(SCIPC), SoPlex (SCIPS) a CLP (SCIPL). Nekomerční solvery jsou CBC [8], Lpsolve [9] a GLPK [10]. Tyto nekomerční solvery se podařilo úspěšně spustit na platformě ARM, další nekomerční solvery nikoli.

Optimalizační problém algoritmu řízení ohřevu vody bojlerů se skládá ze dvou fází

(LP a MILP). Úkolem obou fází je minimalizovat zadané kritérium. Byla provedena série testů pro různé instance optimalizačních problémů. Testy byly prováděny jak na platformě ARM, konkrétně na jednodeskovém počítači PandaBoard ES, který byl vybrán v kapitole 2, tak i pro porovnání na dvoujádrovém procesoru Intel Centrino 2 o frekvenci 2,4 GHz s operační pamětí 2 GB.

Výsledky jednotlivých testů jsou v tabulkách 3.1, 3.2, 3.3 a 3.4. V tabulkách 3.1 a 3.3 jsou výsledky řešení problémů první fáze optimalizace na počítači PandaBoard ES a procesoru Intel. Protože u všech problémů první fáze byla nalezena optimální řešení, tabulky obsahují pouze solverem spotřebovaný čas. V tabulkách 3.2 a 3.4 jsou uvedeny výsledky řešení problémů druhé fáze optimalizace. Protože se nepodařilo všem solverům nalézt optimální řešení všech problémů v zadaném čase, který byl stanoven na 200 respektive 400 sekund, je v tabulkách uvedena procentuální odchylka nejlepšího nalezeného řešení od optimálního. Případ, že došlo k vyčerpání zadaného času je v tabulkách dále zdůrazněn hvězdičkou u spotřebovaného času. V dolní části tabulek jsou uvedeny hodnoty optimálních řešení.

PandaBoard ES	Problém 1	Problém 2	Problém 3
Solver	Čas (s)	Čas (s)	Čas (s)
<b>CBC</b>	1,41	1,38	1,37
<b>GLPK</b>	4,20	4,80	3,30
<b>LPSOLVE</b>	110,84	106,41	112,27

Tabulka 3.1: Tabulka výsledných časů solverů při řešení problémů první fáze optimalizace na počítači PandaBoard ES

Z tabulek 3.1 a 3.3 je vidět, že se všem solverům s výjimkou solveru Lpsolve podařilo nalézt řešení problémů první fáze ve velmi krátkém čase a to jak na procesoru Intel tak na počítači PandaBoard ES. Z tabulky 3.4 je vidět, že se pouze nekomerčnímu solveru CBC podařilo na procesoru Intel v zadaném čase nalézt podobně kvalitní řešení problémů druhé fáze jako komerčním solverům. Řešení, kterých dosáhly solvery na počítači Panda-

PandaBoard ES	Problém 1		Problém 2		Problém 3	
Solver	Odchylka (%)	Čas (s)	Odchylka (%)	Čas (s)	Odchylka (%)	Čas (s)
<b>CBC</b>	5,23	200,00*	9,01	200,00*	0,00	4,81
	3,00	400,00*	0,91	400,00*		
<b>GLPK</b>	37,93	200,00*	86,23	200,00*	0,00	200,00*
	37,40	400,00*	38,84	400,00*		
<b>LPSOLVE</b>	24,82	200,00*	54,00	200,00*	30,55	200,00*
	24,82	400,00*	49,73	400,00*	30,40	400,00*
<b>optimum</b>	13 563,44		16 600,78		2 360,72	

Tabulka 3.2: Tabulka porovnání výsledků solverů při řešení problémů druhé fáze optimalizace na počítači PandaBoard ES

Intel	Problém 1	Problém 2	Problém 3
Solver	Čas (s)	Čas (s)	Čas (s)
<b>CBC</b>	0,18	0,14	0,17
<b>GLPK</b>	0,50	0,70	0,40
<b>LPSOLVE</b>	22,64	15,65	17,26
<b>SCIPL</b>	15,58	0,80	0,83
<b>SCIPS</b>	2,93	0,62	1,09
<b>GUROBI</b>	0,07	0,08	0,10

Tabulka 3.3: Tabulka porovnání výsledných časů solverů při řešení problémů první fáze optimalizace na procesoru Intel

Board ES ve stejném čase (200 sekund), jsou ale výrazně horší, viz tabulka 3.2. Až při zdvojnásobení zadaného času se podařilo, a to jen nekomerčnímu solveru CBC, nalézt podobně kvalitní řešení.

Intel	Problém 1		Problém 2		Problém 3	
Solver	Odchylka (%)	Čas (s)	Odchylka (%)	Čas (s)	Odchylka (%)	Čas (s)
<b>CBC</b>	3,00	200,00*	0,84	200,00*	0,00	0,37
<b>GLPK</b>	23,14	200,00*	43,60	200,00*	0,00	27,00
<b>LPSOLVE</b>	23,85	200,00*	46,66	200,00*	30,34	200,00*
<b>SCIPL</b>	2,98	200,00*	0,36	200,00*	0,00	0,39
<b>SCIPS</b>	3,00	200,00*	0,36	200,00*	0,00	0,47
<b>GUROBI</b>	2,98	200,00*	0,36	200,00*	0,00	0,11
<b>optimum</b>	13 563,44		16 600,78		2 360,72	

Tabulka 3.4: Tabulka porovnání výsledků solverů při řešení problémů druhé fáze optimalizace na procesoru Intel

Výsledky dosažené v těchto a dalších testech vedly k názoru, že optimalizační problém druhé fáze, jak je nyní vytvořen, je příliš složitý, aby byl řešen nekomerčním solverem na platformě ARM. Z tohoto důvodu byl navržen heuristický algoritmus, který nalezne řešení problému druhé fáze na platformě ARM v řádově kratším čase. Výsledky uvedené v kapitole 4 ukazují, že kvalita řešení je srovnatelná s MILP optimalizací.

# Kapitola 4

## Heuristický algoritmus pro optimalizaci výkonové bilance

### 4.1 Úvod

Jak bylo ukázáno v kapitole 3, nalézt řešení optimalizačních LP problémů první fáze se na desce s procesorem ARM podařilo nekomerčním solverům ve velmi krátkém čase. Oproti tomu se jim nepodařilo v krátkém čase nalézt kvalitní řešení MILP problémů druhé fáze. Proto byl vytvořen heuristický algoritmus nahrazující MILP optimalizaci druhé fáze. Heuristický algoritmus bude popsán v následující části této kapitoly.



## 4.2 Popis heuristického algoritmu

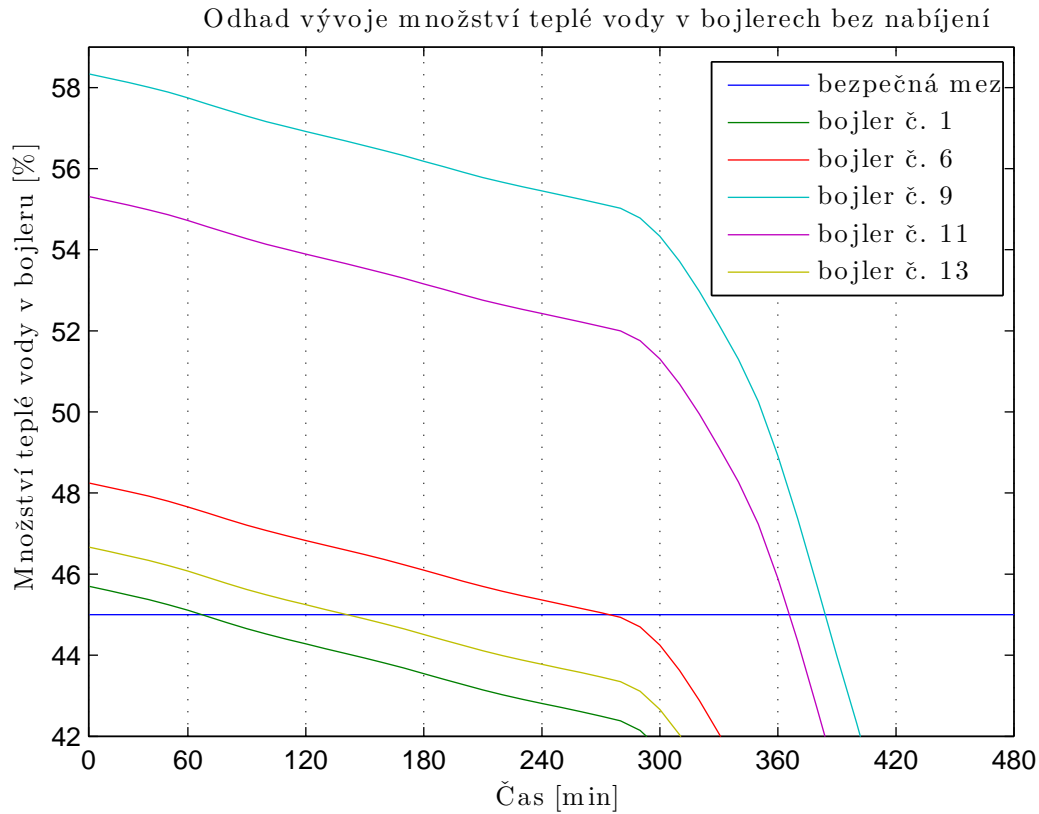
Následující část popisuje, jak byla pro účely simulace a případného dalšího použití systému změněna druhá fáze optimalizace. Heuristický algoritmus vychází ze stejných principů jako výše zmíněný MILP problém. Snaží se tedy při splnění stejných omezení nalézt řešení, které se blíží optimální výkonové bilanci z první fáze.

Optimalizační LP a MILP problém pracují s predikcí vývoje výkonové bilance, oproti tomu heuristický algoritmus využívá jen jeho aktuální změřenou hodnotu. Stejně jako MILP optimalizace tak i heuristický algoritmus závisí na správné funkci předcházející LP optimalizace. Proto pro správnou funkci heuristického algoritmu byly provedeny testy s parametrem popisujícím nejistotu predikce vývoje výkonové bilance, který ovlivňuje LP optimalizaci. Více je v další části této kapitoly.

Protože jsou k dispozici pouze bojler, které neobsahují žádné senzory k měření aktuální teploty vody, z nichž by mohla být čerpána informace o jejich stavech, je určení stavů bojlerů založeno na odhadu. Jediná komunikace, která s bojlerem probíhá, je informace předávaná bojleru o tom zda se má nabíjet a informace od něj získávaná o jeho aktuálním příkonu. K přesnému určení stavu bojleru, tj. množství teplé vody v něm, dochází pouze v okamžiku, kdy je nabíjení bojleru ukončeno jeho vnitřním termostatem. Jeho příkon poklesne na nulu, přestože není nabíjení blokováno řídicím systémem. Odhad stavu bojleru je založen na obvyklých průbězích spotřeby teplé vody v domácnostech. Tento systém využívá průběhů spotřeby vody z [13], z nichž byly vybrány průběhy typické pro Německo jako České republiky nejpodobnější.

Rozdíl mezi skutečnou spotřebou teplé vody a jejím typickým průběhem vyvolává chybu v odhadu stavu. Tato chyba narůstá s každým krokem od posledního plného nabití. Jako ochranné opatření byla zavedena tzv. bezpečná mez, tedy procentuální hodnota celkového možného nabití bojleru, pod kterou není dovoleno odhadu jeho stavu klesnout. Dalším ochranným opatřením je zavedení parametru maximální doby od posledního plného nabití. Systém tedy nedovolí, aby uplynula doba od posledního plného nabití delší než je tento parametr. Vliv těchto parametrů a volba jejich velikosti jsou popsány v další části této kapitoly.

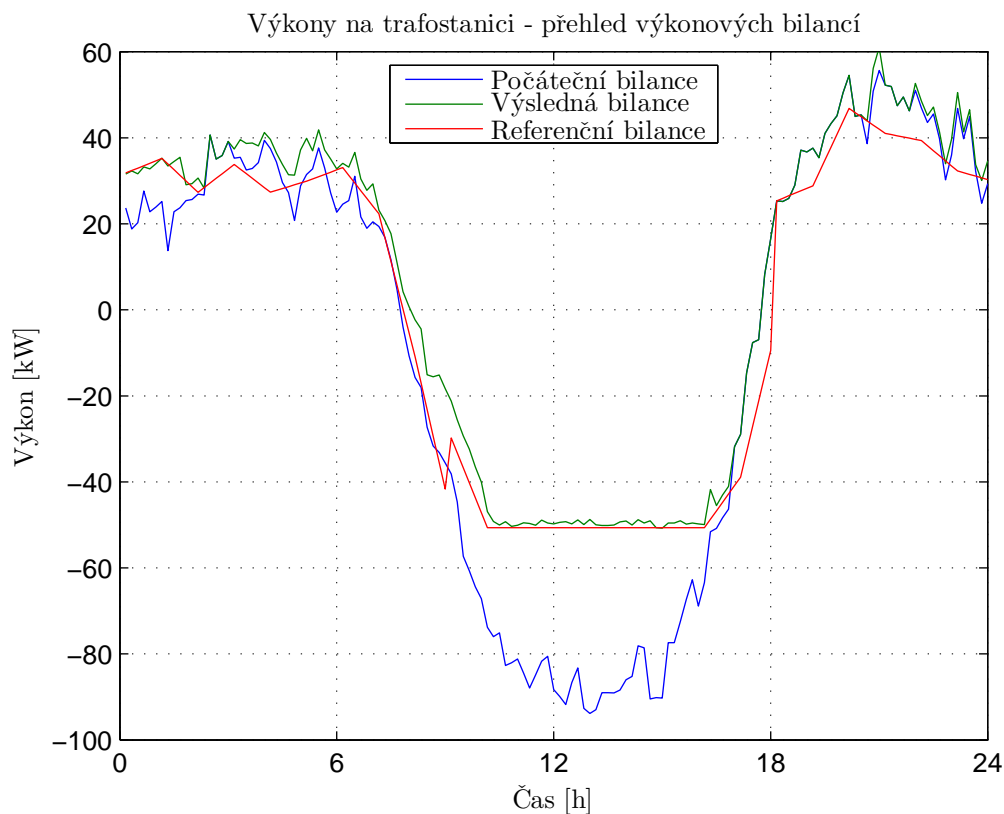
Podobně jako se určuje odhad aktuálního stavu bojleru, tak heuristický algoritmus



Obrázek 4.1: Odhad vývoje množství teplé vody v bojlerech bez nabíjení

stanoví pro každý řízený bojler odhad vývoje jeho stavu bez nabíjení. Ukázka u vybraných bojlerů je na obrázku 4.1. Dále se stanoví očekávané časy vybití bojlerů, kdy podle odhadu vývoje jejich stavu dojde bez nabíjení k jeho vybití pod bezpečnou mez, která je rovněž uvedena na obrázku 4.1 (v uvedeném příkladu je bezpečná mez 45 %). Jak je vidět, nejdříve se vybijí bojler č. 1 za přibližně 70 minut, potom bojler č. 6 za 140 minut a tak dále.

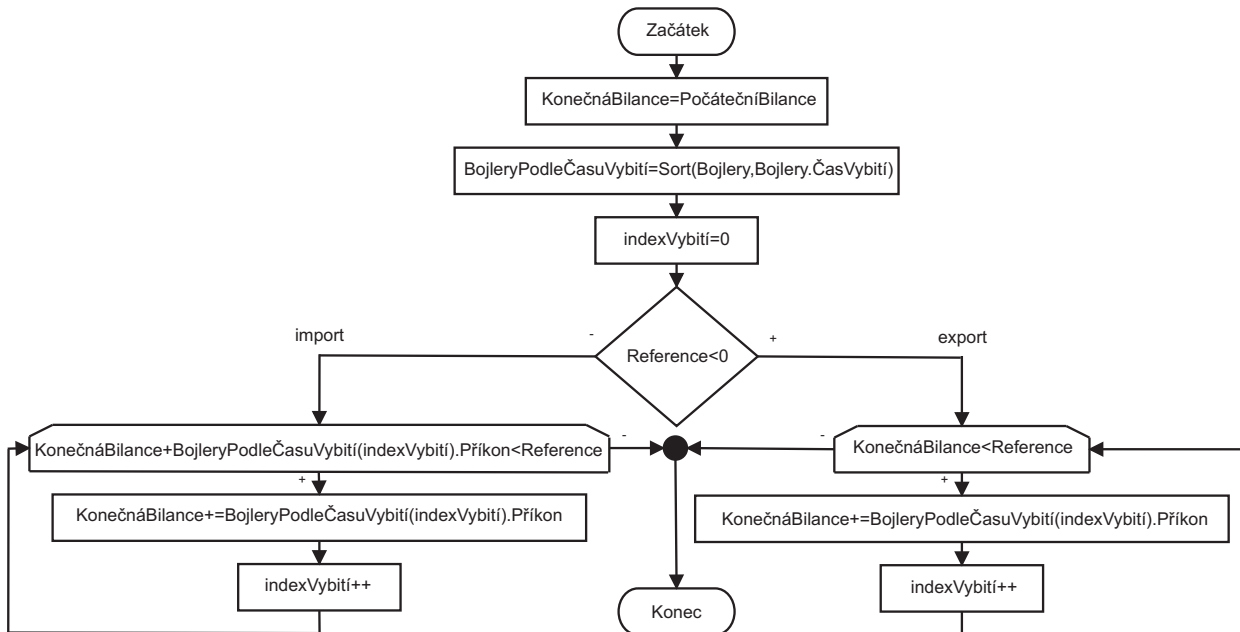
Algoritmus postupně aktivuje bojler, které se podle odhadu vývoje jejich stavu vybijí nejdříve. V časech importu energie do oblasti, kdy je výkonová bilance kladná, algoritmus aktivuje bojler, tak aby byl součet změřené výkonové bilance bez bojlerů, dále označované jako počáteční výkonová bilance, a celkového příkonu spuštěných bojlerů menší než referenční výkonová bilance získaná první fází optimalizace. V případě exportu



Obrázek 4.2: Výkony na trafostanici - přehled výkonových bilancí

energie z oblasti, tj. záporné výkonové bilance, se aktivují bojler postupně tak, aby byl součet počáteční výkonové bilance a celkového příkonu spuštěných bojlerů větší než referenční výkonová bilance, ale jen minimálně, protože úkolem je se výslednou výkonovou bilancí blížit referenční. V časech importu energie se bojler přestanou spouštět v případě, že by spuštění dalšího bojleru změnilo hodnotu konečné výkonové bilance na větší než je referenční. V dobách exportu energie se přestanou aktivovat další bojler v případě, že spuštění posledního způsobilo, že hodnota výsledné bilance překročila referenci, další zvyšování výsledné bilance není dovoleno. Je-li bojler aktivován, zůstane aktivní po minimální dobu nabíjení, tj. 30 minut, tím se zamezí přílišnému přepínání bojlerů. Ukázka výkonových bilancí je na obrázku 4.2.

Postup heuristického algoritmu je znázorněn vývojovým diagramem na obrázku 4.3. Nejprve se do proměnné *KonečnáBilance* uloží hodnota aktuální počáteční výkonové

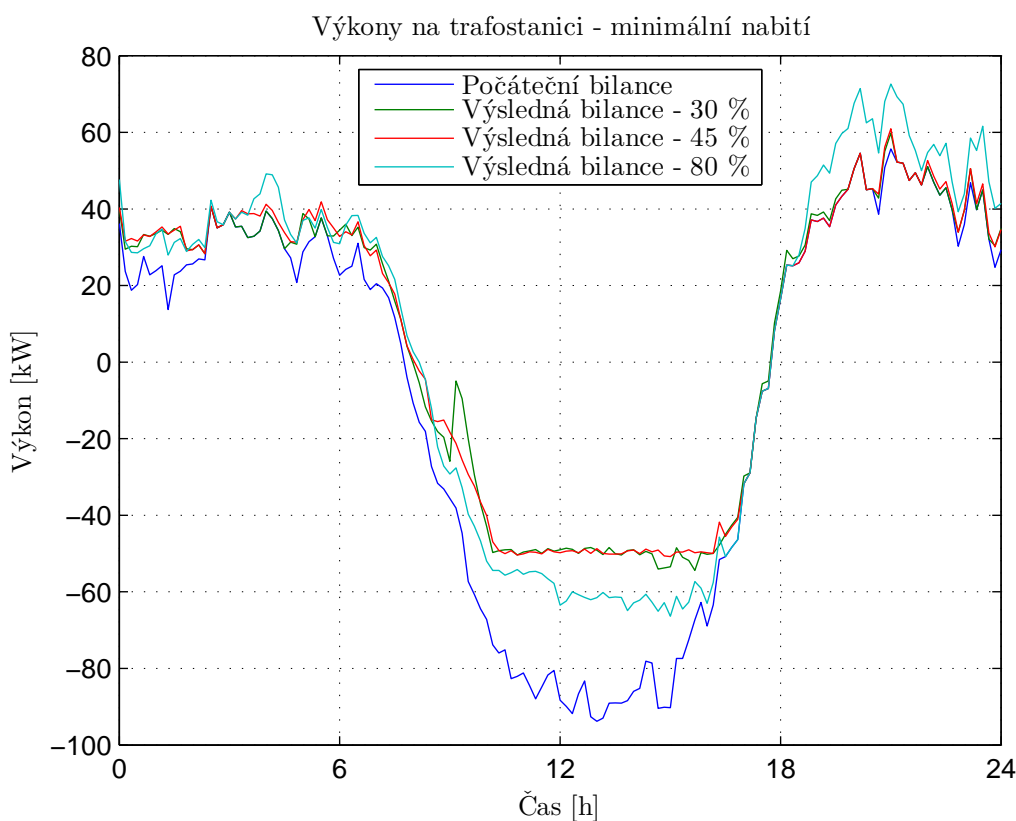


Obrázek 4.3: Vývojový diagram heuristického algoritmu

bilance *PočátečníBilance* a bojlerů se seřadí podle odhadu času jejich vybití. Poté se odděleně řeší případy, kdy se energie transportuje z a do oblasti, tj. referenční výkonová bilance *Reference* je záporná, respektive nezáporná. V případě importu energie do oblasti se posuzuje, zda spuštění dalšího bojleru nezpůsobí nárůst výsledné výkonové bilance *KonečnáBilance* nad referenční výkonovou bilanci *Reference*. Po splnění této podmínky dojde k navýšení výsledné výkonové bilance o jeho příkon. Bude postupně spuštěno tolik bojlerů, u kolika vyjde výsledná výkonová bilance menší než referenční. V případě exportu energie z oblasti se bojlerů spouští podle toho, zda je aktuální výsledná výkonová bilance *KonečnáBilance* menší než referenční *Reference*. Obdobně dochází k postupnému posuzování bojlerů a zvyšování výsledné bilance, dokud není podmínka porušena.

### 4.3 Nastavení parametrů heuristického algoritmu

Pro správnou funkci heuristického algoritmu bylo třeba vhodně zvolit některé parametry. Jedním z těchto parametrů je bezpečná mez nabití, pro kterou byly provedeny tři testy s hodnotou nastavenou na 30 %, 45 % a 80 %. Na obrázku 4.4 jsou zobrazeny průběhy naměřené během jednoho dne. Jsou zde uvedeny výsledné výkonové bilance pro



Obrázek 4.4: Výkony na trafostanici - minimální nabití

různé hodnoty tohoto parametru a průběh počáteční výkonové bilance bez příkonu bojlerů. V tabulce 4.1 je uvedeno u kolika bojlerů z celkového počtu 43 došlo během jednoho dne k vybití a průměrná doba jednoho vybití. Při nastavení tohoto parametru na příliš vysokou hodnotu, tj. 80 %, jsou bojlerů nuceny se velmi často nabíjet a není tak dán velký prostor pro plánování, což podtrhuje obrázek 4.4, kde tomu odpovídá nejhorší výsledek. Naopak malá hodnota, tj. 30 %, tento prostor zvětšuje, ale také zvyšuje riziko možného vybití bojleru. Z tabulky 4.4 tomu odpovídá během jednoho dne úplné vybití 5 bojlerů

Parametr	Výsledná bilance minimální nabití		
	30 %	45 %	80 %
Počet vybitých bojlerů [-]	5	1	0
Průměrná délka vybití [min]	100	10	0

Tabulka 4.1: Tabulka počtu vybití bojlerů a jejich průměrné délky při různých hodnotách parametru minimálního nabití

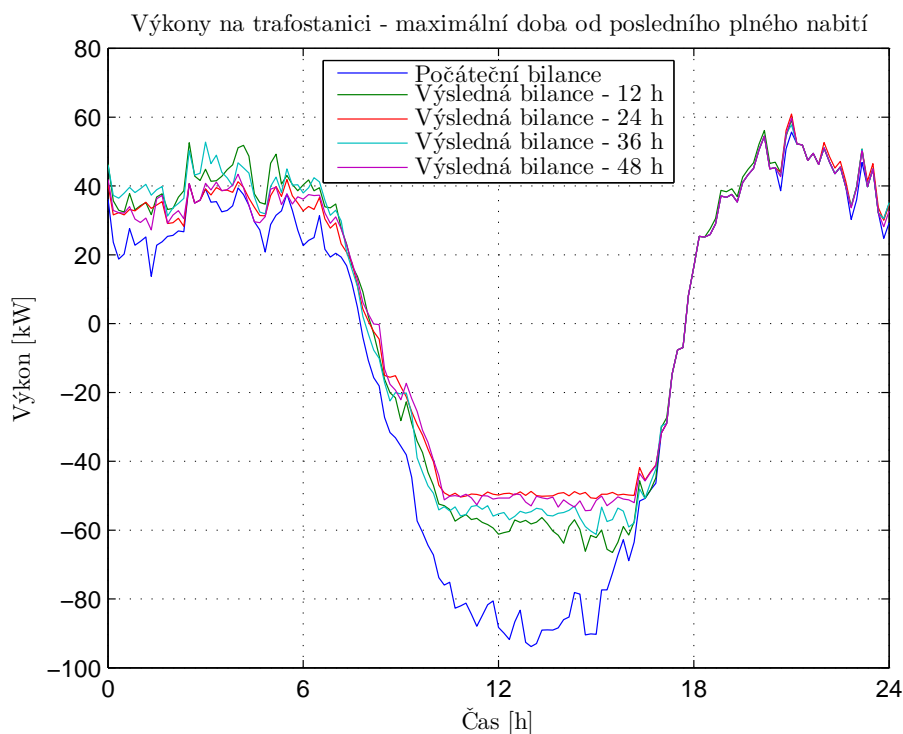
s průměrnou délkou 100 minut. Aby došlo k vyvážení obou situací bylo třeba hodnotu parametru minimálního nabití vhodně stanovit mezi těmito extrémy, tj. na 45 %.

Dalším ochranným opatřením, aby nedocházelo k přílišnému nárůstu chyby v odhadu stavu bojleru je zavedení maximálního povoleného času od posledního plného nabití. Byly provedeny čtyři s hodnotou tohoto parametru nastavenou na 12, 24, 36 a 48 hodin. Na obrázku 4.5 jsou zobrazeny průběhy naměřené během jednoho dne. Jsou zde uvedeny výsledné výkonové bilance pro různé hodnoty tohoto parametru a průběh počáteční výkonové bilance bez příkonu bojlerů. V tabulce 4.2 je uvedeno u kolika bojlerů z celkového počtu 43 došlo během jednoho dne k vybití a průměrná doba jednoho vybití. Zvolením ne-

Parametr	Výsledná bilance maximální doba od nabití			
	12 h	24 h	36 h	48 h
Počet vybitých bojlerů [-]	1	1	1	5
Průměrná délka vybití [min]	110	10	20	74

Tabulka 4.2: Tabulka počtu vybití bojlerů a jejich průměrné délky při různých hodnotách parametru maximální doby od posledního plného nabití

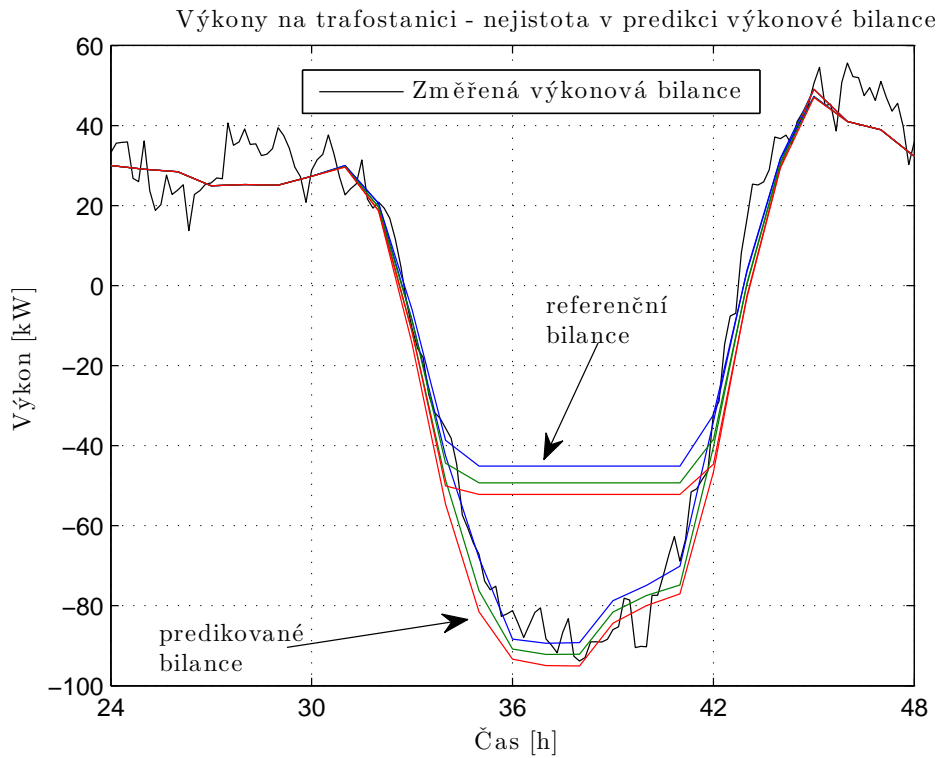
vhodně malého parametru, tj. 12 hodin, bude docházet k příliš častému nucenému nabíjení bojlerů a nebude dán prostor pro dodržení reference získané první fází optimalizace, tomu



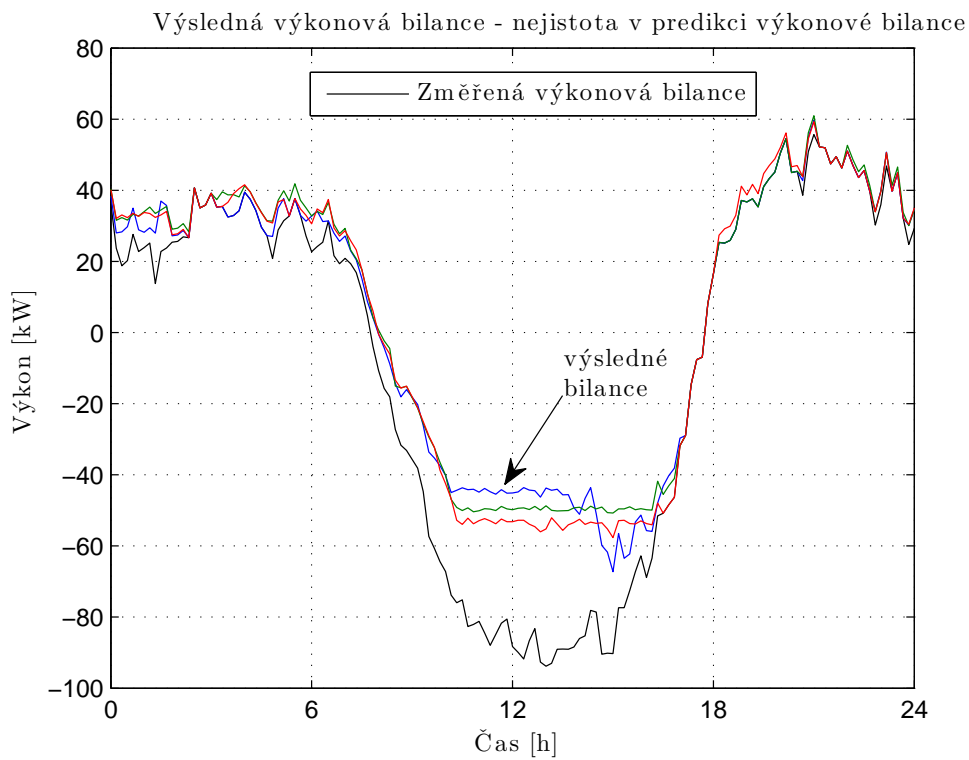
Obrázek 4.5: Výkony na trafostanici - maximální doba od posledního plného nabití

na obrázku 4.5 odpovídá průběh výsledné bilance s největšími špičkami. Naproti tomu zvolením nevhodně velké hodnoty, tj. 48 hodin, se zvyšuje riziko častého úplného vybití, viz tabulka 4.2, kde tomu odpovídá 5 vybitých bojlerů s průměrnou dobou vybití 74 minut. Z výsledků se jeví jako nejlepší nastavení tohoto parametru hodnota 24 hodin.

Posledním parametrem, se kterým byly provedeny testy pro jeho správné nastavení, je parametrem nejistoty predikce výkonové bilance. Na obrázku 4.6 je vidět průběh změřené výkonové bilance, tři její predikce pro různé hodnoty parametru nejistoty a jím odpovídající referenční bilance získané druhou fází optimalizace. Na obrázku 4.7 jsou průběhy výsledné výkonové bilance získané heuristickým algoritmem. Nejlepšího výsledku, tedy nejmenší špičky a nejmenšího objemu transportované energie, bylo dosaženo výslednou bilancí odpovídající zelené křivce. Parametr nejistoty predikce výkonové bilance byl nastaven na tomu odpovídající hodnotu.



Obrázek 4.6: Nejistota v predikci výkonové bilance - referenční výkonová bilance



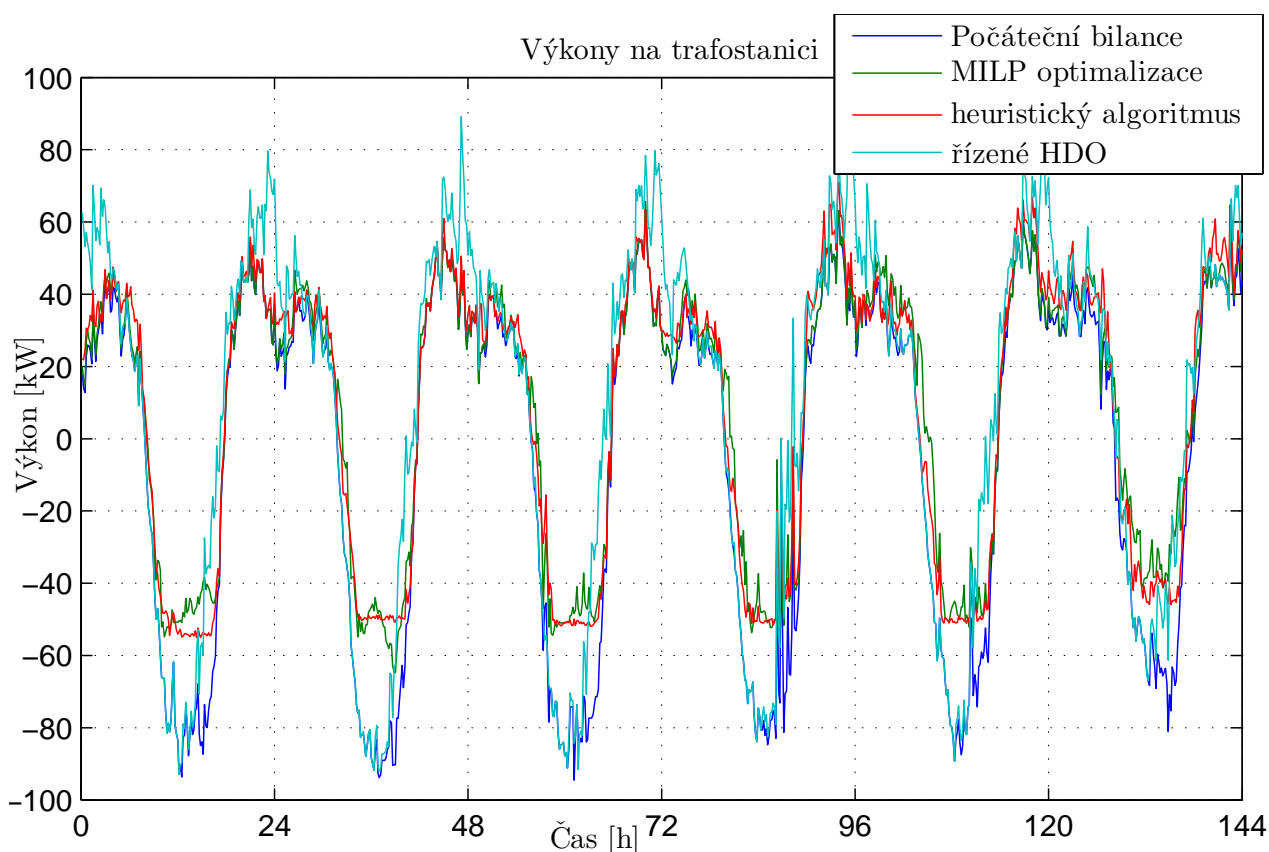
Obrázek 4.7: Nejistota v predikci výkonové bilance - výsledná výkonová bilance



## 4.4 Simulace s navrženým systémem

Simulace byly prováděny pro oblast nacházející se v České republice, kde byla změřena výkonová bilance na jaře 2011. Oblast obsahuje celkem 43 řízených bojlerů. Simulováno bylo 6 dnů, od pondělí do soboty. Minimální doba nabíjení bojleru byla stanovena na 30 minut. První fáze optimalizace se prováděla dvakrát denně v 6 a 18 hodin s 48 hodinovým horizontem a s vzorkovací periodou 1 hodina. Druhá fáze se prováděla každých deset minut a horizontem u MILP optimalizace 6 hodin. Simulace byly provedeny na počítači s 6-jádrovým procesorem o frekvenci 3,2 GHz a operační paměti 32 GB.

V současné době se pro ohřev vody bojleru používá systém hromadného dálkového

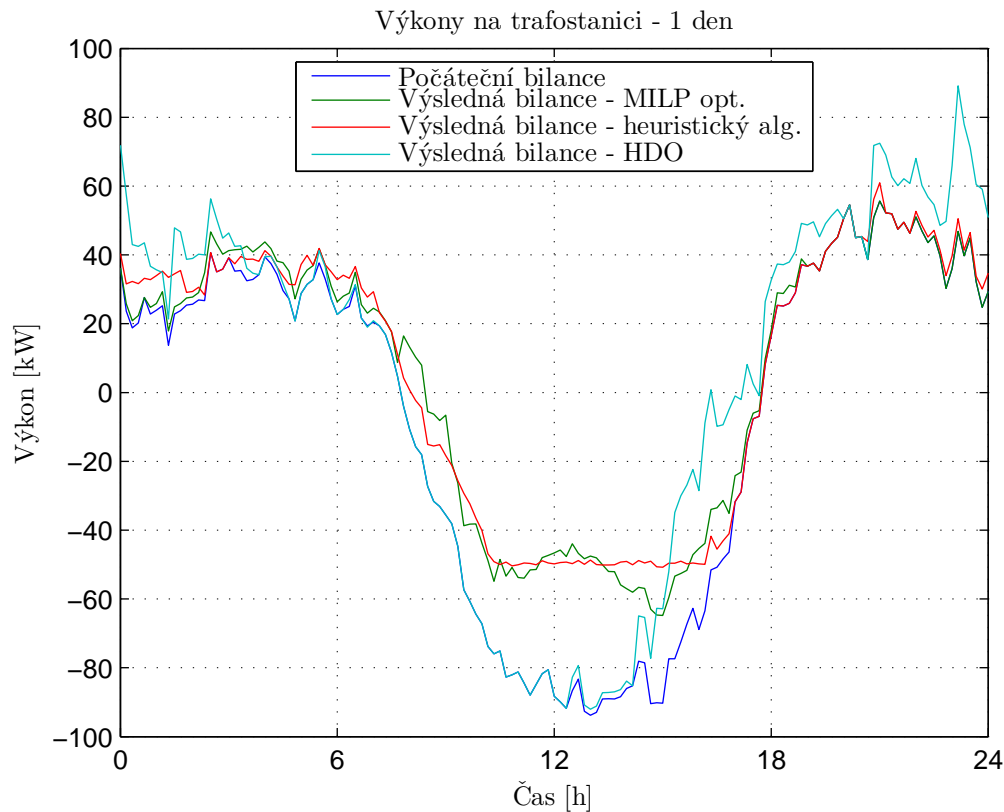


Obrázek 4.8: Výkony na trafostanici

ovládání (HDO). Systém HDO neumožňuje individuální řízení jednotlivých bojlerů, ale jen větší skupiny. Není tedy vhodný k řešení výše zmíněného problému. Více o tomto

systemu je v článku [11]. Pro názornost jsou uvedeny výsledky, kterých bylo dosaženo tímto systémem.

Na obrázku 4.8 je výsledek všech 6 dnů, a na obrázku 4.9 je pro názornost pouze 2. den



Obrázek 4.9: Výkony na trafostanici - 1 den

simulace. Na obrázcích je porovnání výsledné výkonové bilance současného řízení bojlerů systémem HDO, simulace systému s druhou fází řízenou MILP optimalizací a navrženým heuristickým algoritmem, dále obrázky obsahují počáteční výkonovou bilanci bez příkonu bojlerů. V tabulce 4.3 jsou shrnuty průměrné denní kvalitativní parametry a celkové vybití za 6 dnů simulace.

Na obrázcích a v tabulce je vidět, že v dobách importu energie do oblasti si nejlépe vede systém řízený MILP optimalizací, průměrná denní špička je nejmenší a v této době se v průměru nejméně energie transportuje. Naopak systém HDO v těchto časech velmi významně zvyšuje průměrnou denní špičku i průměrný objem importované energie. Objem

importované energie v případě řízení heuristickým algoritmem a MILP optimalizátorem je velmi podobný, špička v případě heuristického algoritmu je mírně vyšší.

V dobách transportu energie z oblasti je nejmenší průměrné denní špičky dosaženo

Parametr	Počáteční bilance	Výsledná bilance algoritmus		
		MILP	Heuristika	HDO
Průměrná špička exportu [kW]	89,49	53,52	51,05	86,12
Průměrná špička importu [kW]	58,41	59,40	63,77	82,65
Průměrná vzdálenost špiček [kW]	147,90	112,92	114,82	168,77
Průměrná exportovaná energie [MWh]	33,87	18,47	21,45	24,88
Průměrná importovaná energie [MWh]	39,33	46,34	46,54	56,32
Průměrná celková transportovaná e. [MWh]	73,20	64,81	67,98	81,20
Celkový počet vybití bojlerů [-]	-	11	7	0
Průměrná délka vybití [min]	-	100	29	0
Doba výpočtu jednoho cyklu [s]	-	65,64	0,52	-

Tabulka 4.3: Tabulka porovnání kvalitativních parametrů výsledné bilance pro různé algoritmy

heuristickým algoritmem, ale nejmenšího průměrného denního transportu energie z oblasti je dosaženo MILP optimalizací. Naopak systém HDO průměrnou špičku exportu energie zmenší jen málo a objem exportované energie je nejvyšší. Nejvyhlazenější výsledné výkonové bilance, zejména v době exportu energie, bylo dosaženo heuristickým algoritmem, ale nejmenšího objemu transportované energie dosáhl systém řízený MILP optimalizací. Během simulace těchto 6 dnů se systémem HDO nedošlo k žádnému vybití bojleru, během řízení heuristickým algoritmem k celkem 7 s průměrnou délkou 29 minut a u MILP optimalizace k 11 s průměrnou délkou 100 minut. Při porovnání průměrné doby výpočtu druhé fáze optimalizace v jednom cyklu je vidět, že spotřebovaný čas MILP optimalizací je oproti heuristickému algoritmu výrazně větší.

Nedá se tedy říci, že by MILP optimalizace vyprodukovala nějak výrazně lepší řešení

než heuristický algoritmus. V jistých ohledech, jak bylo řečeno, je dokonce heuristika lepší a navíc její časové nároky jsou výrazně menší. Obě řešení dosáhla lepších výsledků než systém HDO, který výrazně nezlepšuje tvar výkonové bilance.

# Kapitola 5

## Závěr

Prvním z úkolů této práce bylo seznámení se s parametry a možnostmi platformy ARM, porovnání dostupných vývojových desek na bázi této platformy a výběr vhodné desky pro vývoj algoritmů. Tímto úkolem se zabývala kapitola 2. Cílem je, aby byl v budoucnu na této desce implementován algoritmus řízení ohřevu vody v bojlerech. Kritériem výběru byla dostatečně velká operační paměť, možnost komunikace po Ethernet rozhraní, možnost použití linuxového systému a podpora výrobcem nebo komunitou. Jako zástupci dvou dostupných typů těchto desek ARM Cortex-A8 a ARM Cortex-A9, byly vybrány dle kritérií desky BeagleBoard-xM a PandaBoard ES.

Dalším úkolem, kterým se zabývala tato práce v kapitole 3, bylo porovnání dostupných solverů pro řešení optimalizačních problémů dvou fází řídicího algoritmu. Jednalo se v první fázi o problémy typu lineárního programování (LP) a ve druhé fázi o problémy smíšeného celočíselného lineárního programování (MILP). Cílem byl výběr vhodného, z důvodu celkové ceny nekomerčního, solveru funkčního na vybraných deskách s procesorem ARM. Z nekomerčních solverů, které bylo možné spustit na těchto deskách, si nejlépe vedl solver CBC. Výsledky testů ale ukázaly, že časy řešení MILP problémů dosažené solverem na ARM platformě nebyly dostatečně krátké pro reálné použití.

Z tohoto důvodu byl navržen heuristický algoritmus, který nahrazuje část algoritmu řízení ohřevu vody. Výsledky uvedené v kapitole 4 ukazují, že využití navrženého heuristického algoritmu je výrazně rychlejší než hledání řešení za pomoci MILP optimalizace.

Z výsledků je dále vidět, že kvalita řešení heuristického algoritmu je srovnatelná s MILP optimalizací.

# Literatura

- [1] What makes ARM-based chips relatively power efficient?: and what is the trade-off for lower power consumption?. In: *Quora* [online]. Last modified on 25. 5. 2011 [cit. 2013-04-07]. Dostupné z:  
<http://www.quora.com/What-makes-ARM-based-chips-relatively-power-efficient>
- [2] ARM architecture. In: *Wikipedia: the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, 11. 12. 2006, last modified on 6. 4. 2013 [cit. 2013-04-07]. Dostupné z:  
[https://en.wikipedia.org/wiki/ARM\\_architecture](https://en.wikipedia.org/wiki/ARM_architecture)
- [3] ARM Holdings, *Cortex-A series processors* [online]. Last modified on 7. 5. 2010 [cit. 2013-04-07]. Dostupné z:  
<http://infocenter.arm.com/help/index.jsp>
- [4] List of single-board computers. In: *Wikipedia: the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, 11. 12. 2006, last modified on 31. 3. 2013 [cit. 2013-04-07]. Dostupné z:  
[http://en.wikipedia.org/wiki/List\\_of\\_single-board\\_computers](http://en.wikipedia.org/wiki/List_of_single-board_computers)
- [5] G. Coley. *BeagleBoard-xM Rev C System Reference Manual*, BeagleBoard.org, 2010. [online]. Dostupné z:  
[http://beagleboard.org/static/BBxMSRM\\_latest.pdf](http://beagleboard.org/static/BBxMSRM_latest.pdf)

- [6] *OMAP4460 Pandaboard ES System Reference Manual*, pandaboard.org, 2011. [online]. Dostupné z:  
[http://pandaboard.org/sites/default/files/board\\_reference/ES/Panda\\_Board\\_Spec\\_DOC-21054\\_REVO\\_1.pdf](http://pandaboard.org/sites/default/files/board_reference/ES/Panda_Board_Spec_DOC-21054_REVO_1.pdf)
- [7] H. Mittelmann. *Benchmarks for Optimization Software* [online]. Last modified on 3. 4. 2013 [cit. 2013-04-07]. Dostupné z:  
<http://plato.asu.edu/bench.html>
- [8] COIN-OR Branch-and-Cut MIP Solver. In: *Trac Open Source Project* [online]. Last modified on 21. 11. 2012 [cit. 2013-04-07]. Dostupné z:  
<https://projects.coin-or.org/Cbc>
- [9] lp\_solve reference guide. In: *SourceForge* [online]. [cit. 2013-04-07]. Dostupné z:  
<http://lpsolve.sourceforge.net/5.5/>
- [10] GLPK (GNU Linear Programming Kit). In: *GNU* [online]. Last modified on 23. 6. 2012 [cit. 2013-04-07]. Dostupné z:  
<http://www.gnu.org/software/glpk/>
- [11] A. Neuberger. *Ripple control in the Czech Republic and Demand Side Management*, 20th International Conference and Exhibition on Electricity Distribution - Part 1, CIRED 2009, 2009. Dostupné z:  
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5255405>
- [12] O. Malík, P. Havel. *Active Demand-side Management System to Facilitate Integration of RES in Low Voltage Distribution Networks*, submitted for publication in IEEE Transactions on Sustainable Energy, 2013.
- [13] I. Knight, N. Kreutzer, M. Manning, M. Swinton a H. Ribberink. *Residential cogeneration systems: European and Canadian residential non-HVAC electric and DHW load profiles for use in simulating the performance of residential cogeneration systems*, Annex 42 report, IEA ECBCS, 2007. [online].  
[http://www.ecbcs.org/docs/Annex\\_42\\_Domestic\\_Energy\\_Profiles.pdf](http://www.ecbcs.org/docs/Annex_42_Domestic_Energy_Profiles.pdf)



# Obsah přiloženého CD

Na přiloženém CD je tato diplomová práce ve formátu PDF.