

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



Hyperbolická miera efektívnosti v obálkovej analýze dát

DIPLOMOVÁ PRÁCA

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Hyperbolická miera efektívnosti v obáľkovej analýze dát

DIPLOMOVÁ PRÁCA

Študijný program: Ekonomicko-finančná matematika a modelovanie
Študijný odbor: 1114 Aplikovaná matematika
Školiace pracovisko: Katedra aplikovanej matematiky a štatistiky
Vedúci práce: doc. RNDr. Margaréta Halická, CSc.



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Denisa Jančovičová
Študijný program: ekonomicko-finančná matematika a modelovanie
(Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: aplikovaná matematika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Hyperbolická miera efektívnosti v obálkovej analýze dát.
Hyperbolic efficiency measure in data envelopment analysis.

Cieľ: Väčšina modelov v obálkovej analýze dát (DEA) sa formuluje pomocou lineárneho programovania. Hyperbolický model je však úlohou nelineárneho programovania a spoľahlivé metódy na jeho riešenie neboli až donedávna známe. Cieľom diplomovej práce je popísať vlastnosti tohoto modelu a preskúmať možnosti jeho riešenia pomocou heuristického algoritmu predstaveného v článku [1] a metódy semidefinitného programovania navrhnutého v článku [2]. Literatúra: [1] Färe, R.S., Margaritis, D., Rouse, P., Roshdi, I., 2016. Estimating the hyperbolic distance function: A directional distance function approach. *European Journal of Operational Research* 254, 312-319 [2] Halická M., Trnovská M.: A novel approach to nonlinear DEA: application to Russell measure, working paper.

Vedúci: doc. RNDr. Margaréta Halická, CSc.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: prof. RNDr. Daniel Ševčovič, CSc.
Dátum zadania: 25.01.2017

Dátum schválenia: 27.01.2017

prof. RNDr. Daniel Ševčovič, CSc.
garant študijného programu

.....
študent

.....
vedúci práce

Pod'akovanie Touto cestou by som chcela vyjdríť úprimnú vďaku svojej vedúcej diplomovej práce doc. RNDr. Margaréte Halickej, CSc. najmä za podnetné rady a cenné pripomienky poskytnuté pri písaní tejto práce, ale taktiež za množstvo času, ochoty a trpezlivosti, ktoré mi venovala. Zároveň by som rada poďakovala aj svojej rodine a blízkym za ich podporu.

Abstrakt v štátnom jazyku

JANČOVIČOVÁ, Denisa: Hyperbolická miera efektívnosti v obálkovej analýze dát [Diplomová práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: doc. RNDr. Margáreta Halická, CSc., Bratislava, 2018

V tejto diplomovej práci sa zaoberáme určitým, špecifickým modelom obálkovej analýzy dát (Data Envelopment Analysis), presnejšie hyperbolickým modelom, ktorý je na rozdiel od väčšiny modelov obálkovej analýzy dát úlohou nelineárneho programovania. Keďže spoľahlivé metódy riešenia tohto modelu neboli dlho známe, kľúčovou časťou tejto práce je preskúmanie a porovnanie možností jeho riešenia pomocou vybraných algoritmov na dátových súboroch rôznych veľkostí. Prvou z metód je heuristický algoritmus z článku [10], ktorý porovnávame s metódou semidefinitného programovania predstavenou v článku [16]. Ďalej navrhujeme našu vlastnú modifikáciu uvedeného heuristického algoritmu s cieľom zjednodušiť a zefektívniť výpočty. V praktickej časti práce využívame nami naprogramované modely v softvéri MATLAB.

Kľúčové slová: Data Envelopment Analysis (DEA), Hyperbolický model, nelineárne programovanie

Abstract

JANČOVIČOVÁ, Denisa: Hyperbolic efficiency measure in data envelopment analysis [Diploma Thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: doc. RNDr. Margaréta Halická, CSc., Bratislava, 2018

In this Diploma thesis, we focus on the specific DEA model, precisely speaking, on Hyperbolic model which is, contrary to the majority of DEA models, problem of non-linear programming. Since reliable methods of solving this model have been undiscovered for a long time, the key part of this thesis is to explore and compare different possibilities of solving this model using particular algorithms on datasets of various dimensions. The first method is the heuristic algorithm from the article [10] which is compared with the method of semi-definite programming introduced in the article [16]. Furthermore, we suggest our own modification of the above mentioned heuristic algorithm with the aim to simplify and to increase the efficiency of calculations. In the practical part of the thesis we use models programmed by ourselves in the solver MATLAB.

Keywords: Data Envelopment Analysis (DEA), Hyperbolic model, non-linear programming

Obsah

Úvod	9
1 Úvod do DEA modelov	11
1.1 Symbolika a základné pojmy	11
1.2 Hyperbolický model	16
1.3 DDF model	18
1.4 Environmentálne modifikácie	20
2 Metódy riešenia hyperbolického modelu	25
2.1 Metóda I	26
2.1.1 Popis algoritmu	26
2.1.2 Ilustračný príklad	31
2.1.3 Konvergencia algoritmu	32
2.2 Metóda II	34
2.3 Metóda III	38
2.4 Environmentálne modifikácie metód	39
2.4.1 Metóda I v prípade nežiadúcich výstupov	40
2.4.2 Metóda II v prípade nežiadúcich výstupov	42
2.4.3 Metóda III v prípade nežiadúcich výstupov	44
3 Praktická časť	45
3.1 Základné verzie metód	46
3.2 Environmentálne verzie	47
3.3 Veľké dátové súbory	56
Záver	63
Zoznam použitej literatúry	65
Príloha A	68
A Zdrojové kódy programov pre MATLAB	68
A.1 Metóda I	68

A.2 Metóda II	76
A.3 Metóda III	83
Príloha B	86
B Vybrané výsledky príkladov	86

Úvod

Momentálne sme svedkami markantne narastajúcej pozornosti venovanej zefektívňovaniu procesov či už pri narábaní s verejnými zdrojmi, pri obsluhu zákazníkov v supermarkete či pri iných výrobných a nevýrobných procesoch. Snaha o minimalizáciu nákladov a maximalizáciu príjmov, napríklad prostredníctvom zefektívnenia procesov, stavia do popredia práve vedecké odbory zaoberajúce sa optimalizáciou a riešením rôznych optimalizačných problémov. Jedným z nich je práve obáľková analýza dát (Data Envelopment Analysis, DEA), v ktorej kľúčovým pojmom je práve efektivita.

DEA modely sa využívajú najmä na vyhodnocovanie miery efektivity konkrétnej rozhodovacej jednotky (*Decision Making Unit*) vzhľadom na určitú, homogénnu skupinu producentov. Súčasne však môžu pomáhať aj pri identifikovaní nedostatkov v premieňaní vstupov na výstupy voči efektívnemu vzoru, ktorý vznikol istou kombináciou efektívnych útvarov danej skupiny. Zvyčajne sú DEA modely spájané s lineárnym programovaním, avšak, ako neskôr uvidíme, existujú modely obáľkovej analýzy dát definované aj pomocou nelineárneho programovania.

K teórii DEA modelov už v súčasnosti nájdeme mnohé knihy či články, ktoré predstavujú a hlbšie objasňujú problematiku DEA, avšak azda jedinou publikáciou v slovenskom jazyku je [13], ktorá spolu s [2], [7] a množstvom ďalších článkov, tvorili podstatný zdroj teoretických poznatkov nevyhnutných k tejto práci. Využitelnosť a strategický význam DEA modelov v riadiacich a rozhodovacích procesoch podčiarkuje množstvo článkov, v ktorých sa môžeme stretnúť s praktickou aplikáciou DEA modelov v rôznych situáciach.

K základným modelom lineárneho programovania radíme zaužívané CCR a BCC modely. Tieto patria do kategórie *orientovaných* modelov, v prípade ktorých tvorí stredobod nášho záujmu hľadanie nedostatkov buď len vo vstupoch alebo iba vo výstupoch. Úlohou je teda minimalizácia vstupov pri konštantných výstupoch alebo maximalizácia výstupov pri konštantných vstupoch. Na druhej strane sa môžeme stretnúť aj s takzvanými *neorientovanými* modelmi, ktoré neuprednostňujú výstupy pred vstupmi ani vstupy pred výstupmi. Do tejto kategórie patria napríklad aditívny alebo SBM model zo skupiny modelov lineárneho programovania, avšak, ako sa spomína v [16], tiež modely nelineárneho programovania. Príkladom môže byť Russelov model či hyperbolický

model, pričom na druhý spomenutý sa zameriame v našej práci.

Hyperbolický model, resp. hyperbolická miera efektívnosti ako sa zvykne často uvádzať v literatúre, bol uvedený autormi Färom, Grosskopfom a Lovellom v publikácii [7] vydannej v roku 1985. Na rozdiel od väčšiny DEA modelov, ide o model nelineárneho programovania, a preto je nutné nájsť spôsob, akým by sa dalo vysporiadať s výpočtovým aspektom jeho riešenia. Z toho dôvodu vzniklo niekoľko článkov, ako napríklad [9], [11] alebo [21], avšak my sa budeme primárne venovať algoritmom prestaveným v článkoch [10] a [16].

Cieľom našej práce je teda popísať vlastnosti tohto modelu a empiricky porovnať možnosti jeho riešenia prostredníctvom oboch uvedených algoritmov. V práci taktiež rozoberáme určitú modifikáciu hyperbolického modelu, ktorá berie do úvahy aj tzv. *nežiadúce* výstupy. Takýto typ modelov je často využívaný v environmentálnej problematike, a preto je práca doplnená o verzie modelov, ktoré dokážu hodnotiť mieru efektivity aj na základe vyprodukovaných nežiadúcich výstupov. Tieto verzie modelov budeme často označovať aj ako *environmentálne verzie* alebo *environmentálne modifikácie* hyperbolického modelu. V tejto súvislosti upravíme aj uvedené algoritmy riešenia modelov. Použitie oboch metód ilustrujeme na konkrétnych súboroch dát rôznej veľkosti, čomu predchádza naprogramovanie jednotlivých algoritmov v softvéri MATLAB. Ďalej v práci navrhujeme miernu modifikáciu metódy z článku [10] s cieľom zefektívnenia výpočtov a následne analyzujeme výsledky pozorovaní.

Naša práca je rozdelená na teoretickú a praktickú časť. Teoretická časť práce pozostáva z dvoch kapitol. V prvej kapitole zavedieme symboliku, stručne objasníme centrálné pojmy týkajúce sa teórie DEA a bližšie popíšeme hyperbolický model a tzv. *Funkciu smerovej vzdialenosti*, na ktorej je postavená metóda z článku [10] a aj naša vlastná modifikácia tejto metódy. V uvedenej kapitole sa tiež venujeme environmentálnym verziám oboch uvedených modelov. V nasledujúcej kapitole sa zaoberáme popisom jednotlivých metód riešenia hyperbolického modelu, na ktoré sme sa v práci zamerali, a zároveň uvádzame aj nami navrhnutú modifikáciu algoritmu z článku [10]. V druhej, praktickej časti práce demonštrujeme rozdiely vybraných postupov riešenia hyperbolického modelu. Na príkladoch s rôzne veľkými dátovými súbormi ilustrujeme rozdiely v hodnotách efektívít a časovej efektívnosti jednotlivých prístupov.

1 Úvod do DEA modelov

Cieľom našej práce je porovnať rôzne metódy výpočtu jedného z netypických modelov obálkovej analýzy dát, konkrétne hyperbolického modelu. Najprv však musíme predstaviť základnú symboliku a definície potrebné k ďalšiemu využitiu teórie DEA, a preto úvodná kapitola pojednáva najmä o elementárnych pojmoch a spôsoboch značenia používaných v tejto práci. Ako sa uvádza vyššie, pri spracovaní nasledujúcej časti čerpáme najmä z učebného textu [13], avšak práca je obohatená aj o poznatky z publikácií [2], [7] a zo stránky [5].

1.1 Symbolika a základné pojmy

V dnešnej dobe sa stále viac stavia do popredia problematika efektívneho využívania zdrojov či už v podnikateľskej sfére, vo verejných inštitúciách alebo v mnohých iných oblastiach. Práve vyhodnocovanie efektívnosti a efektivity je primárnym záujmom obálkovej analýzy dát, ktorej poznatky dokážu prostredníctvom rôznych DEA modelov prispieť k hospodárnejšiemu využívaniu prostriedkov daného subjektu. Základným pilierom DEA modelov je teória lineárneho programovania (LP), avšak postupne medzi nimi nachádzajú svoje miesto aj modely formulované ako úlohy nelineárneho programovania (NLP).

Tieto metódy operačného výskumu využívajú iba dáta o vstupoch a výstupoch sledovaných útvarov, pomocou ktorých dokážu posudzovať efektivitu jednotlivých producentov v porovnaní s ostatnými v danej skupine. Podstatu DEA modelov teda tvorí určovanie relatívnej efektívnosti individuálnych subjektov v rámci určitej skupiny producentov, pričom všetci producenti tejto skupiny sa riadia tou istou *technológiou*. Zároveň podotýkame, že technológiou rozumieme spôsob pretvárania svojich zdrojov na výsledné produkty. Z uvedeného je prirodzené vyžadovať homogénnosť danej skupiny subjektov zabezpečujúcu rovnorodosť výrobnéj či nevýrobnéj aktivity, ktorou sa subjekty zaoberajú. Inými slovami, nemôžeme porovnávať nemocnice s pobočkami obchodných reťazcov alebo so školami či fabrikami. V našej práci sa napríklad bližšie pozrieme na efektívnosť aerolínií, škôl či elektrární.

Vyššie spomínané rozhodovacie jednotky alebo tiež nazývané *DMU*, z anglického De-

cision Making Unit, ktoré tvoria základnú skupinu n porovnávaných subjektov, označujeme indexom $j \in \{1, \dots, n\}$. Avšak pri upriamení sa na konkrétny organizačný útvar potrebujeme tento odlíšiť od zvyšných v skupine, a z toho dôvodu ho označujeme ako DMU_o , pre $o = 1, \dots, n$. Prirodzene môžeme dedukovať, že technológiu všetkých DMU_j charakterizujeme hodnotami každého z m vstupov a s výstupov danej rozhodovacej jednotky. Množstvo zužitkovaného i -teho vstupu j -teho subjektu reprezentuje označenie x_{ij} a symbolom y_{rj} rozumieme vyprodukované množstvo r -tého výstupu j -teho subjektu, pre $i \in \{1, \dots, m\}$ a $r \in \{1, \dots, s\}$. V našej práci sa zaoberáme len prípadmi, keď subjekty spotrebujú určité nezáporné množstvo vstupných faktorov k vytvoreniu istého nezáporného množstva výstupných faktorov. Zároveň predpokladáme, že organizačné útvary minú kladné množstvo aspoň jedného vstupu a súčasne vyprodukovujú kladné množstvo aspoň jedného výstupu. Týmto spôsobom definované vektory vstupov $x_j \in \mathbb{R}_+^m$ a vektory výstupov $y_j \in \mathbb{R}_+^s$ matematicky vyjadríme nasledovne:

$$x_j = \begin{pmatrix} x_{1j} \\ \vdots \\ x_{ij} \\ \vdots \\ x_{mj} \end{pmatrix}, \quad y_j = \begin{pmatrix} y_{1j} \\ \vdots \\ y_{rj} \\ \vdots \\ y_{sj} \end{pmatrix}, \quad \forall j = 1, \dots, n.$$

V súčasnosti sa stále väčšia dôležitosť pripisuje ekologickej udržateľnosti výrobných postupov a zodpovednosti voči životnému prostrediu. Z toho dôvodu sa aj v prípade DEA modelov môžeme často stretnúť s prípadom nežiadúcich výstupov, ako napríklad pri produkcii rôznych škodlivých či znečisťujúcich látok do ovzdušia. Obdobným príkladom sa zaoberáme aj neskôr v našej práci. Existujú rôzne spôsoby, akými sa dá s nežiadúcimi výstupmi v jednotlivých modeloch vysporiadať a my ich ďalej rozoberieme konkrétne pre hyperbolický model v podkapitole 1.4.

Z hľadiska lepšej prehľadnosti ďalších definícií sa inšpirujeme [13, str. 61] a údaje reprezentujúce vybranú skupinu pozorovaných rozhodovacích útvarov označíme ako množinu

$$Z = \{(x_j, y_j) \mid j = 1, \dots, n\}.$$

V empirickej časti našej práce sa snažíme aplikovať teóriu DEA predstavenú v prvých dvoch kapitolách na konkrétnych príkladoch. Nevyhnutnú súčasť tejto časti teda tvorí

naprogramovanie uvedených modelov, pri ktorom využívame takzvané matice vstupov X a matice výstupov Y . Konkrétne vektory vstupov x_j , resp. vektory výstupov y_j , danej jednotky DMU_j , pre $\forall j = 1, \dots, n$ tvoria stĺpce spomínaných matíc, a teda:

$$X = \begin{pmatrix} x_{11} & \dots & x_{1j} & \dots & x_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i1} & \dots & x_{ij} & \dots & x_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{m1} & \dots & x_{mj} & \dots & x_{mn} \end{pmatrix}, \quad Y = \begin{pmatrix} y_{11} & \dots & y_{1j} & \dots & y_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{r1} & \dots & y_{rj} & \dots & y_{rn} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{s1} & \dots & y_{sj} & \dots & y_{sn} \end{pmatrix}.$$

Nemenej dôležitým pojmom v súvislosti s meraním efektivity je aj mikroekonomický pojem *množiny produkčných možností* G , ktorá predstavuje všetky rôzne možnosti, akými môžu jednotlivé útvary DMU_j spracovať x_j vstupov na tvorbu y_j výstupov, pre $\forall j = 1, \dots, n$. V ďalšom ale využívame iba jej aproximáciu M , ktorá má niekoľko tvarov. V prvom rade objasníme dve základné podoby tejto množiny, a teda prípad s variabilnými a s konštantnými výnosmi z rozsahu, často označované ako M_{VRS} , resp. M_{CRS} . Aproximácie M_{CRS} a M_{VRS} sú, podľa [13, kap. 5], tvorené najmenšími množinami generovanými množinou Z a disponujú nasledovnými kľúčovými vlastnosťami množiny G :

- (A1) $Z \subset M$. Všetky body zo Z , reprezentujúce konkrétne subjekty, patria do M .
- (A2) M je konvexná, čiže všetky konvexné kombinácie bodov patriacich do M patria tiež do množiny M .
- (A3) Ak bod $(x_A, y_A) \in M$ vyjadruje útvar A , potom do množiny M patria aj všetky body $(x, y) \in \mathbb{R}^{m+s}$, pre ktoré platí $x_A \leq x \quad \vee \quad y \leq y_A$.

Vlastnosť (A3) predstavuje vlastnosť *voľnej disponovateľnosti*, ktorá znamená, že možno plytvať vstupmi, resp. vyrobiť primálo výstupov. V prípade nežiadúcich výstupov však nemôžeme znížiť produkciu nežiadúcich výstupov bez dodatočných nákladov, a teda v tomto prípade uvažujeme iný typ disponovateľnosti. Ako sa uvádza napríklad v [11], [15] alebo [8], existuje viacero spôsobov, ktorými sa dá vysporiadať s nežiadúcimi výstupmi. Bližšie sa však problematike technologickej množiny v prípade nežiadúcich výstupov venujeme v časti 1.4.

Ešte je potrebné spomenúť jednu axiómu typickú pre konštantné výnosy z rozsahu. Charakteristickou črtou tohto typu výnosov je, že ak c -násobne zväčšíme, resp. zmenšíme, množstvo spotrebovaných prostriedkov, c -násobne sa zväčší, resp. zmenší, aj množstvo vytvorených produktov, čiže:

(A4) Ak $(x_A, y_A) \in M$, potom $(cx_A, cy_A) \in M$, pre ľubovoľné $c > 0$.

Podľa [13, kap. 5] sa dá množina M_{CRS} zapísať analyticky v nasledovnom tvare:

$$M_{CRS} = \{(x, y) \in \mathbb{R}^{m+s} \mid X\lambda \leq x, \quad Y\lambda \geq y, \quad \lambda \geq 0\}. \quad (1)$$

Analytický zápis množiny M_{VRS} je analogický ako vzťah (1), avšak je nutné ešte dodať podmienku

$$e_n^T \lambda = 1, \quad (2)$$

kde e_n predstavuje stĺpcový vektor samých jednotiek rozmeru n . Takto skonštruované množiny M_{CRS} a M_{VRS} tvoria takzvanú *obálku* dát daných producentov, z čoho vyplýva aj názov celej problematiky DEA.

Ak chceme presnejšie vymedziť pojmy efektívnosť a efektivita, je potrebné vychádzať z ústredného pojmu viackriteriálnej optimalizácie tzv. *Pareto optimality*, ktorá je pre DEA modely sformulovaná v [13, str. 171]. *Efektívnymi* sa nazývajú len tie útvary, pre ktoré sa v množine M nenachádza žiadny iný útvar, ktorý by bol schopný z nižšieho alebo rovnakého množstva vstupných prostriedkov vyprodukovať vyššie alebo rovnaké množstvo výstupných produktov. Matematicky vyjadrené, sa za efektívne považujú len tie body $(\hat{x}, \hat{y}) \in M$, ku ktorým nedokážeme nájsť v množine M žiadne iné body $(x, y) \neq (\hat{x}, \hat{y})$ spĺňajúce $x \leq \hat{x}$ a $y \geq \hat{y}$. Takzvaná *hranica efektívnosti*, označovaná ako H_E , pozostáva práve z týchto efektívnych útvarov (\hat{x}, \hat{y}) . V opačnom prípade, ak má útvar rezervy či už v príliš veľkom množstve minutých zdrojov alebo v nedostatočnom množstve vyrobených produktov, hovoríme o *neefektívnom* útvere. Aby sme boli presní, je nutné spomenúť, že hranica ∂M množiny M sa skladá okrem hranice efektívnosti, aj z *hranice pseudoeфективности*, označovanej ako H_P . Táto môže niektorým typom DEA modelov spôsobovať ťažkosti, pretože je tvorená bodmi z hranice ∂M , ktoré síce môžu v jednej veličine dosahovať rovnaké hodnoty ako efektívni producenti, ale v ostatných veličinách pozorujeme isté rezervy, a preto sa nedajú pokladať za efektívne. Týmto nedostatkom disponuje aj hyperbolický model, čo ďalej rozvážame v kapitole 1.2.

V prípade s nežiadúcimi výstupmi je však situácia mierne odlišná, pretože naším cieľom je minimalizovať nežiadúce výstupy. Za efektívne teda budeme považovať tie útvary z množiny M , pre ktoré nebudú v množine M existovať také útvary, ktoré by z menšieho alebo rovnakého množstva vstupov dokázali vyrobiť väčšie alebo rovnaké množstvo žiadúcich výstupov, ale zároveň menšie alebo rovnaké množstvo nežiadúcich výstupov. Vzhľadom k charakteru úlohy je, pochopiteľne, potrebné nejakým spôsobom odlíšiť vektor výstupov pre prípad žiadúcich a nežiadúcich, aby sme boli schopní medzi nimi rozlišovať. Keďže v prvej kapitole je naším cieľom len uviesť čitateľa do základnej problematiky DEA, túto časť sme kvôli prehľadnosti zaradili do kapitoly 1.4.

Ako sme už avizovali, objektom nášho záujmu nebude len odlíšenie efektívnych a neefektívnych útvarov, ale zameriame našu pozornosť aj na zistenie ich miery efektívnosti, resp. efektivity. Efektivita, ako bezrozmerná veličina z intervalu $[0, 1]$, vyjadruje určitú relatívnu efektivitu, resp. mieru, do akej pracuje daný subjekt efektívne vzhľadom k ostatným organizačným jednotkám v skupine. Spôsoby výpočtu efektivity jednotlivých DMU sa však v rôznych modeloch môžu od seba líšiť a nie vždy môže takto definovaná miera efektívnosti priamo vyplývať z riešenia zodpovedajúcej úlohy. V prípade hyperbolického modelu však vieme explicitne odčítať efektivitu v typickom, ekonomicky interpretovateľnom tvare, a preto sa ďalej touto otázkou nezaobráame.

V súvislosti s problematikou miery efektivity je ešte vhodné spomenúť a objasniť pojem *čistá technická miera efektivity*, s ktorou sa neskôr stretneme. Podľa [13, Poznámka 8.5.] alebo knihy [2] dokáže táto miera efektivity zachytávať iba tú neefektivitu, ktorú sa dá odstrániť znížením všetkých vstupov alebo zvýšením všetkých výstupov prostredníctvom určitého koeficientu bez ohľadu na potrebu zmeny pomerov jednotlivých vstupov, resp. výstupov. Pri dosahovaní efektivity však zohrávajú úlohu aj vzájomné proporcie jednotlivých vstupov, prípadne výstupov, z toho dôvodu čistá technická miera efektivity nedokáže zachytiť všetku neefektívnosť a, ako už bolo vyššie načrtnuté, môže daný útvar označiť za efektívny, hoci má určité nedostatky. Takáto situácia nastáva vtedy, keď sa útvar nachádza na hranici pseudoeфективности.

1.2 Hyperbolický model

V úvode našej práce už bolo avizované, že sa budeme venovať najmä jednej miere efektivity - hyperbolickej miere efektivity - ktorá bola prvýkrát sformulovaná v článku [7] ako miera čistej technickej efektivity kombinujúca Farellovu vstupne a výstupne orientovanú mieru efektivity. Tieto miery efektivity sú definované v článku [6] z roku 1957. Podľa knihy [7] alebo článkov [14] či [9] je úlohou hyperbolického modelu simultánne, ekviporcionálne skracovanie vstupov a predlžovanie výstupov prostredníctvom jednej skalárnej veličiny θ .

Vzhľadom k tomu, že sa v našej práci zaoberáme spôsobmi riešenia hyperbolického modelu výlučne pri variabilných výnosoch z rozsahu, uvádzame aj v ďalšom definície modelov v prípade VRS. Formulácia úlohy nelineárneho programovania zodpovedajúca hyperbolickému modelu pre vybraný DMU_o je teda v prípade VRS nasledovná:

$$\min_{\theta, \lambda} \theta \quad (3)$$

$$s.t. \quad X\lambda \leq \theta x_o,$$

$$Y\lambda \geq \frac{1}{\theta} y_o, \quad (4)$$

$$e_n^T \lambda = 1,$$

$$\lambda \geq 0,$$

kde λ predstavuje n -rozmerný vektor a θ predstavuje koeficient maximálneho skrátenia vstupov, respektíve predĺženia výstupov. V prípade konštantných výnosov z rozsahu je podmienky úlohy NLP len vypustené obmedzenie (2).

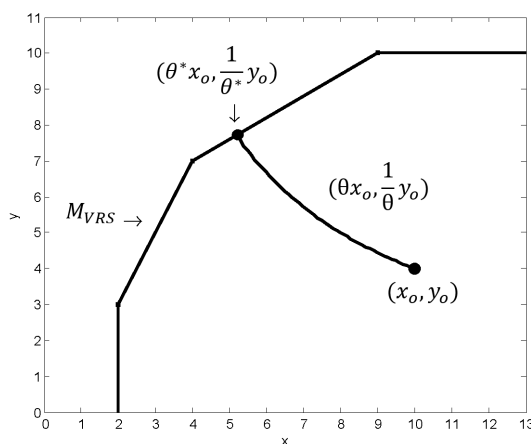
V článku [14] je hyperbolický model popísaný ako radiálny, neorientovaný model, ktorý je invariantný na zmenu jednotiek. Ako sme už uviedli, z riešenia úlohy modelu je možné odčítať efektivitu v typickom, ekonomicky interpretovateľnom tvare ako veličinu z $[0, 1]$. Konkrétne mierou efektivity rozumieme optimálnu hodnotu účelovej funkcie θ^* úlohy (3)-(4). Môžeme si všimnúť, že naozaj ak je $\theta^* = 1$, útvar leží na hranici množiny M a v prípade, že $\theta^* < 1$, sa útvar nachádza vnútri množiny M .

Ako už bolo vyššie spomenuté a uvádza sa aj v článkoch [15] alebo [10], hyperbolický model je čistou technickou mierou efektivity, a teda nezachytáva všetku neefektívnosť. Z toho dôvodu môže hyperbolická miera efektívnosti označiť za efektívne aj útvary, ktoré sú v skutočnosti len pseudoeфекtívne, čiže sa nachádzajú na hranici

pseudoefektívnosti. Vo vyššie uvedených článkoch sa dá nájsť aj spôsob identifikovania pseudoefektivity, a to implementovaním ďalšej fázy riešenia tzv. *MaxSlack* úlohy, ktorá je detailnejšie popísaná v [13].

V súvislosti s hyperbolickým modelom sa často hovorí aj o jeho geometrickej interpretácii, ktorou sa zaoberajú napríklad autorky článku [14] alebo je spomenutá aj v článku [10]. Geometricky sa teda dá úloha interpretovať ako projektovanie útvaru DMU_o daného dvojicou bodov (x_o, y_o) na hranicu ∂M pozdĺž hyperbolickej krivky $(\theta x_o, \frac{1}{\theta} y_o)$. Projektujeme teda bod (x_o, y_o) takým spôsobom, aby sa jeho projekcia stále nachádzala v aproximácii technologickej množiny M . Hodnota parametra θ , pri ktorej krivka $(\theta x_o, \frac{1}{\theta} y_o)$ pretne hranicu ∂M je optimálne riešenie úlohy NLP danej vzťahmi (3)-(4). Takýmto spôsobom dostávame projekciu $(\theta^* x_o, \frac{1}{\theta^*} y_o)$ bodu (x_o, y_o) na hranici množiny M a optimálnu hodnotu θ^* , ktorú chápeme ako mieru efektivity.

Pre lepšiu predstavu geometrickej interpretácie hyperbolického modelu doplníme túto kapitolu ešte o Obrázok 1, na ktorom sa snažíme ilustrovať vyššie uvedené úvahy.



Obr. 1: Ilustračné znázornenie geometrickej interpretácie hyperbolického modelu.

Zdroj: Vlastné spracovanie

Aj z uvedenej geometrickej interpretácie vyplýva, že optimálne hodnoty θ^* ležia určite v intervale $(0, 1]$. Na podmienky (4) úlohy NLP prislúchajúcej hyperbolickému modelu sa dá tým pádom hľadiť aj cez analytický zápis množiny M_{VRS} uvedený v (1). Alternatívny zápis danej úlohy teda môže vyzeráť aj nasledovne:

$$\min_{\theta, \lambda} \{ \theta : (\theta x_o, \frac{1}{\theta} y_o) \in M_{VRS} \}. \quad (5)$$

1.3 DDF model

Kľúčovú úlohu v našej práci zohráva aj v úvode spomínaná Funkcia smerovej vzdialenosti, častejšie uvádzaná po anglicky ako *Directional Distance Function* (DDF), ktorá bola predstavená v článku [3]. V našej práci sa spomína aj ako *DDF model* a na základe vyššie spomenutého článku ho v prípade VRS definujeme ako:

$$\begin{aligned}
 & \max_{\beta, \lambda} \beta \\
 & s.t. \quad X\lambda \leq x_o - \beta g_x, \\
 & \quad Y\lambda \geq y_o + \beta g_y, \\
 & \quad e_n^T \lambda = 1, \\
 & \quad \lambda \geq 0,
 \end{aligned} \tag{6}$$

kde $(g_x, g_y) \in \mathbb{R}_+^m \times \mathbb{R}_+^s$. DDF interpretujeme ako maximálny posun sledovaného útvaru (x_o, y_o) na hranicu ∂M vo vopred určenom smere (g_x, g_y) . Ako si môžeme poľahky všimnúť, pri riešení DDF modelu popri skracovaní vstupov pomocou člena $-\beta g_x$ simultánne predlžujeme výstupy pomocou člena $+\beta g_y$. Z toho dôvodu v skutočnosti posúvame pozorovaný subjekt DMU_o na hranicu technologickej množiny smerom $(-g_x, g_y)$. Za efektívny v zmysle DDF modelu považujeme útvar, pre ktorý vyjde optimálna hodnota premennej β rovná nule, čo implikuje, že útvar nedokážeme viac posunúť smerom (g_x, g_y) , a teda leží na hranici množiny M . Ani DDF model, tak ako hyperbolický model, však nedokáže zachytiť všetku neefektívnosť, pretože meria len čistú technickú efektivitu, a preto môže aj pseudoefektívne útvary označiť za efektívne.

Dôvodom pre zaoberanie sa DDF modelom v našej práci je práve jeho súvislosť s hyperbolickým modelom, ktorého metódami riešenia sa v práci primárne venujeme. Túto súvislosť využívajú nielen autori prvej skúmanej metódy riešenia hyperbolického modelu v článku [10], ale využívame ju aj v našej vlastnej metóde, ktorú opisujeme v časti 2.2. V ďalšom teda ukážeme, že v prípade $(g_x, g_y) = (x_o, y_o)$ možno DDF model chápať ako lineárnu aproximáciu hyperbolického modelu.

V nasledujúcej časti, týkajúcej sa súvisu hyperbolického a DDF modelu, vychádzame primárne z článku [10], avšak mierne odlišný spôsob linearizácie hyperbolického modelu je popísaný napríklad aj v článkoch [14] a [15]. Pre spresnenie situácie, sú výsledné úlohy LP ekvivalentné a líšia sa len v spôsobe odvodenia.

Uvažujúc definíciu hyperbolického modelu pomocou vzťahov (3)-(4), prvým krokom je reparametrizácia $\theta = e^{-\beta}$, pričom $\beta \in [0, \infty)$. Krivka $(\theta x_o, \frac{1}{\theta} y_o)$, pozdĺž ktorej projektujeme bod (x_o, y_o) reprezentujúci útvar DMU_o , sa týmto spôsobom reparametrizuje na $(e^{-\beta} x_o, e^{\beta} y_o)$. Tým pádom sa v bode (x_o, y_o) parameter $\beta = 0$, pretože $(x_o, y_o) = (e^0 x_o, e^0 y_o)$. Ďalej využijeme Taylorovu aproximáciu exponenciálnej funkcie pre argument x „blízko“ 0, konkrétne $e^x \doteq 1 + x$. V našom prípade dostávame $e^{-\beta} x_o \doteq (1 - \beta)x_o$ a $e^{\beta} y_o \doteq (1 + \beta)y_o$, pre β „blízko“ 0, a teda θ „blízko“ 1. Lineárna aproximácia hyperbolického modelu je preto nasledovná:

$$\begin{aligned} \max_{\beta, \lambda} \quad & \beta \\ \text{s.t.} \quad & X\lambda \leq (1 - \beta)x_o, \\ & Y\lambda \geq (1 + \beta)y_o, \\ & e_n^T \lambda = 1, \\ & \lambda \geq 0. \end{aligned} \tag{7}$$

Uvedený model (7) je evidentne DDF definovanou vzťahom (6) na začiatku časti 1.3 pri špecifickej voľbe vektora $(g_x, g_y) = (x_o, y_o)$. Všimnime si, že aj tento model sa dá geometricky interpretovať ako projekcia bodu (x_o, y_o) do bodu $((1 - \beta^*)x_o, (1 + \beta^*)y_o)$. Upozorňujeme však na fakt, že týmto spôsobom dostávame iba aproximáciu hodnoty θ pre hyperbolický model, čo implikuje, že čím je β bližšie k nule a θ k jednotke, tým očakávame presnejšiu aproximáciu.

Podotýkame ešte, že v článku [14] je dokonca dokázané, že DDF model so špeciálnou voľbou vektora $(g_x, g_y) = (x_o, y_o)$ poskytuje spodné ohraničenie pre optimálnu hodnotu účelovej funkcie hyperbolického modelu (3)-(4). Autorky však v článku vychádzajú z už spomínaného ekvivalentného zápisu úlohy (7), konkrétne:

$$\min_{\delta, \lambda} \{ 1 - \delta \mid X\lambda \leq (1 - \delta)x_o, \quad Y\lambda \geq (1 + \delta)y_o, \quad e_n^T \lambda = 1, \quad \lambda \geq 0 \}.$$

Viac informácií spolu s dôkazom uvedeného tvrdenia sa dá nájsť v spomínanom článku.

V dvoch vybraných metódach riešenia hyperbolického modelu predstavených v častiach 2.1 a 2.2 hrá dôležitú úlohu aj optimálne riešenie duálnej úlohy DDF modelu,

ktorej formulácia je v prípade VRS nasledovná:

$$\begin{aligned}
& \max_{p,w,z} y_o^T p - x_o^T w - z \\
& s.t. \quad Y^T p - X^T w - e_n z \leq 0, \\
& \quad g_y^T p + g_x^T w = 1, \\
& \quad p \geq 0_s, w \geq 0_m,
\end{aligned} \tag{8}$$

kde e_n je n -rozmerný vektor jednotiek. Optimálne riešenie duálnej úlohy DDF modelu určuje gradient opornej nadroviny množiny M_{VRS} a práve tento poznatok tvorí podstatnú súčasť vyššie spomenutých metód riešenia hyperbolického modelu. Viac teoretických poznatkov ohľadom interpretácie riešenia duálneho modelu v súvislosti s opornou nadrovinou technologickej množiny sme čerpali aj z prednášok [20] k predmetu Konvexná optimalizácia, ale aj z článku [14, Remark 3], v ktorom sa autorky síce venujú interpretácii riešenia duálnej úlohy hyperbolického modelu, no rovnako sa dajú uvedené tvrdenia aplikovať aj na DDF model.

1.4 Environmentálne modifikácie

Trochu odlišnými typmi hyperbolického či DDF modelu, ktoré uvedieme, sú už spomenuté environmentálne modifikácie uvedených modelov. Motiváciou k odvodeniu týchto modelov je poskytnúť spôsob, akým by sa dalo vysporiadať s nežiadúcimi výstupmi pri zisťovaní miery efektivity.

V tejto podkapitole teda rozoberieme environmentálne modifikácie hyperbolického aj DDF modelu v prípade variabilných výnosov z rozsahu, pričom metódy riešenia prvého uvedeného porovnáваме neskôr v našej práci. Vzhľadom k tomu, že v prípade environmentálnej verzie oboch modelov posudzujeme aj vplyv produkcie nežiadúcich výstupov na mieru efektívnosti, je potrebné tieto odlíšiť od žiadúcich výstupov, od ktorých prirodzene očakávame, aby ich množstvo bolo, na rozdiel od tých nežiadúcich, čo najvyššie. Uvažujme teda útvar DMU_j , pre $j = 1, \dots, n$, ktorý mimo x_j spotrebovaných vstupov a y_j vyrobených žiadúcich výstupov vyprodukuje ešte b_j nežiadúcich výstupov. Symbolom b_{kj} , pre $k \in \{1, \dots, p\}$, označíme vyprodukované množstvo k – teho nežiadúceho výstupu subjektu j , kde $j \in \{1, \dots, n\}$ a $k \in \{1, \dots, p\}$. Týmto spôsobom dostaneme

vektor nežiadúcich výstupov $b_j \in \mathbb{R}_+^p$, ktorý môžeme zapísať v nasledovnej podobe:

$$b_j^T = (b_{1j}, \dots, b_{kj}, \dots, b_{pj}), \quad \forall j = 1, \dots, n.$$

Ostáva však otázkou, ako upraviť vlastnosti technologickej množiny tak, aby odrážala skutočnosť, že znižovanie nežiadúcich výstupov nie je možné bez dodatočných nákladov. Z odlišnej povahy úlohy vyplýva, že nemá zmysel uvažovať technologickú množinu s vlastnosťou voľnej disponovateľnosti. Ako sme avizovali aj v časti 1.1, v tejto časti sa venujeme preformulovaniu technologickej množiny pre prípad environmentálnej problematiky. V našej práci sa zaoberáme len dvomi typmi technologickej množiny, ktoré sú uvedené napríklad v článku [15] alebo [11]. Pri prvej z nich uvažujeme voľnú disponovateľnosť vstupov a žiadúcich výstupov, t.j. môžeme plytvať vstupmi a byť leniví v produkcii výstupov, rozdiel je však v spôsobe vysporiadania sa s nežiadúcimi výstupmi, pri ktorých neuvažujeme žiadnu disponovateľnosť. Uvedená skutočnosť sa teda dá zapísať v prípade uvažovania VRS ako:

$$M_0 = \{(x, y, b) \in \mathbb{R}^{m+s+p} \mid X\lambda \leq x, \quad Y\lambda \geq y, \quad B\lambda = b, \quad e_n^T \lambda = 1 \quad \lambda \geq 0\}. \quad (9)$$

Druhý typ technologickej množiny, s ktorým v práci narábame, sa líši od prechádzajúceho len v postupe zaobchádzania s nežiadúcimi výstupmi. Konkrétne v tomto prípade uvažujeme tzv. *reverznú voľnú disponovateľnosť* nežiadúcich výstupov, ktorá je intuitívne jasná zo samotného zápisu množiny:

$$M_1 = \{(x, y, b) \in \mathbb{R}^{m+s+p} \mid X\lambda \leq x, \quad Y\lambda \geq y, \quad B\lambda \leq b, \quad e_n^T \lambda = 1 \quad \lambda \geq 0\}. \quad (10)$$

Uvedený zápis znamená, že technológia pripúšťa možnosť byť neobozretní a neekologickí a zvýšiť množstvo vyprodukovaných nežiadúcich výstupov bez dodatočných výrobných nákladov.

Ďalej sa venujeme kľúčovej časti tejto kapitoly, a teda definíciám environmentálnych modifikácií hyperbolického a DDF modelu. Ako prvú rozoberáme environmentálnu verziu hyperbolického modelu definovaného vzťahmi (3)-(4). Pri tejto modifikácii opäť vychádzame z článkov [15] a [11], pričom na druhý menovaný sa odvolávajú autori metódy z článku [10] pri demonštrovaní ich algoritmu v prípade environmentálnej problematiky.

V porovnaní s verziou bez nežiadúcich výstupov sa len modifikujú podmienky (4) v závislosti od typu technologickej množiny. Uvažujúc technologickú množinu M_0 , definovanú v (9), dostávame optimalizačný problém:

$$\begin{aligned}
& \min_{\theta, \lambda} \theta \\
& s.t. \quad X\lambda \leq x_o, \\
& \quad Y\lambda \geq \frac{1}{\theta}y_o, \\
& \quad B\lambda = \theta b_o, \\
& \quad e_n^T \lambda = 1, \\
& \quad \lambda \geq 0,
\end{aligned} \tag{11}$$

na ktorý sa dá z geometrického hľadiska hľadiť aj nasledovne:

$$\min_{\theta, \lambda} \{ \theta : (x_o, \frac{1}{\theta}y_o, \theta b_o) \in M_0 \}.$$

V prípade druhého typu technologickej množiny M_1 , určenú vzťahom (10), definujeme environmentálnu verziu hyperbolického modelu ako:

$$\begin{aligned}
& \min_{\theta, \lambda} \theta \\
& s.t. \quad X\lambda \leq x_o, \\
& \quad Y\lambda \geq \frac{1}{\theta}y_o, \\
& \quad B\lambda \leq \theta b_o, \\
& \quad e_n^T \lambda = 1, \\
& \quad \lambda \geq 0,
\end{aligned} \tag{12}$$

ktorú opäť môžeme geometricky interpretovať, ako v prípade vyjadrenia (5), analogicky:

$$\min_{\theta, \lambda} \{ \theta : (x_o, \frac{1}{\theta}y_o, \theta b_o) \in M_1 \}.$$

Oba modely sa pochopiteľne dajú upraviť pre prípad CRS odstránením podmienky (2). Dávame do pozornosti, že optimálna hodnota účelovej funkcie v oboch prípadoch opäť nadobúda hodnotu z intervalu $(0, 1]$ a reprezentuje mieru efektivity. To znamená, že efektívnym útvarom je v zmysle modelov (11), resp. (12), útvar s optimálnou hodnotou $\theta^* = 1$.

Druhou modifikáciou rozoberanou v tejto podkapitole je environmentálna verzia DDF modelu, daného vzťahom (6), v ktorej spočíva podstata metódy riešenia hyperbolického modelu s nežiadúcimi výstupmi z článku [10], ale taktiež sa o ňu opiera naša vlastná metóda.

Pri definícii DDF modelu s nežiadúcimi výstupmi čerpáme najmä z článkov [15] a [10]. Avšak vzhľadom k tomu, že v zdroji [10] autori uvádzajú len definíciu DDF modelu bez nežiadúcich výstupov a v článku [15] nachádzame len určitú analógiu primárnej úlohy (6) pre prípad nežiadúcich výstupov, formulácie nižšie uvedených modelov len vychádzajú z predmetných zdrojov a nie sú z nich presne prebrané.

K nižšie uvedenej primárnej úlohe DDF modelu odvádzame aj jeho duálnu úlohu, ktorej riešenie tvorí neoddeliteľnú súčasť metódy z článku [10] a aj našej vlastnej.

Základná myšlienka modifikácie DDF modelu je analogická ako v prípade hyperbolického modelu s nežiadúcimi výstupmi. Aj v tomto prípade sa len upravujú podmienky optimalizačnej úlohy (6) v závislosti od typu uvažovanej technologickej množiny. Preto v prípade technologickej množiny M_0 , vymedzenej vzťahom (9), definujeme DDF model s nežiadúcimi výstupmi pre VRS ako:

$$\begin{aligned}
 & \max_{\beta, \lambda} \quad \beta \\
 & s.t. \quad X\lambda \leq x_o, \\
 & \quad \quad Y\lambda \geq y_o + \beta g_y, \\
 & \quad \quad B\lambda = b_o - \beta g_b, \\
 & \quad \quad e_n^T \lambda = 1, \\
 & \quad \quad \lambda \geq 0,
 \end{aligned} \tag{13}$$

kde $(g_y, g_b) \in \mathbb{R}_+^s \times \mathbb{R}_+^p$. V prípade technologickej množiny M_1 zo vzťahu (10) je miesto tretieho obmedzenia nerovnosť $B\lambda \leq b_o - \beta g_b$.

Opäť si všimnime obdobný súvis environmentálnej verzie DDF modelu s environmentálnou verziou hyperbolického modelu. Vychádzajúc z hyperbolického modelu s nežiadúcimi výstupmi z (11) alebo (12) a využijúc rovnakú reparametrizáciu ako pri jeho základnej verzii, t.j. $\theta = e^{-\beta}$, a Taylorovu aproximáciu exponenciálnej funkcie pre argument x „blízko“ 0, dostávame lineárnu aproximáciu hyperbolického modelu s nežiadúcimi výstupmi, ktorá odpovedá DDF modelu s nežiadúcimi výstupmi pri špeciálnej

voľbe vektora $\vec{g} = (g_y, g_b) = (y_o, b_o)$.

Nakoniec sa pozrime ešte na analógiu duálnej úlohy (8) v prípade environmentálnej verzie DDF modelu (13), pod ktorou rozumieme v prípade technologickej množiny M_0 nasledovný problém:

$$\begin{aligned} \max_{p, p_B, w, z} \quad & y_o^T p - x_o^T w - b_o^T p_B - z \\ \text{s.t.} \quad & Y^T p - X^T w - B^T p_B - e_n z \leq 0, \\ & g_y^T p + g_b^T p_B = 1, \\ & p \geq 0_s, w \geq 0_m, \end{aligned} \tag{14}$$

kde e_n je n -rozmerný vektor jednotiek. K duálnej úlohe prislúchajúcej DDF modelu s nežiadúcimi výstupmi pri technologickej množine M_1 je pridaná podmienka na premennú p_B , konkrétne $p_B \geq 0_p$.

V nasledujúcej kapitole sa bližšie pozrieme na možnosti riešenia hyperbolického modelu, daného vzťahmi (3)-(4), vrátane jeho environmentálnych verzií (11) a (12).

2 Metódy riešenia hyperbolického modelu

Ako sme už naznačili, hyperbolický model, reprezentovaný vzťahmi (3)-(4), spolu s jeho environmentálnymi verziami definovanými optimalizačnými problémami (11) a (12), sú modelmi nelineárneho programovania a spoľahlivé metódy ich riešenia až donedávna absentovali. V nasledujúcej časti preto popíšeme tri možné spôsoby riešenia tohto modelu. Prvým z nich je metóda prezentovaná v článku [10] založená na aproximácii hyperbolického modelu DDF modelom, čiže úlohou lineárneho programovania (LP), o ktorej hovoríme ako o *Metóde I*. Následne predstavíme našu vlastnú modifikáciu Metódy I, ktorej algoritmus je intuitívnejší a má za cieľ zefektívnenie výpočtov. Túto metódu v ďalšom uvádzame ako *Metódu II*. Poslednou rozoberanou metódou je algoritmus z článku [16], ktorý vychádza z poznatkov semidefinitného programovania (SDP), a objavuje sa v našej práci pod názvom *Metóda III*.

Pre prípad CRS je v článku [9] ukázané, že sa hyperbolický model dá transformovať na úlohu LP, konkrétne na Farellovu vstupnú alebo výstupnú mieru efektivity. My tieto pojmy poznáme aj pod názvom vstupný, prípadne výstupný CCR model. Kľúčovú úlohu v tomto prípade zohráva vlastnosť (A4) produkčnej množiny pri CRS v mierne inom označení, a teda $\theta M = M$, $\theta > 0$, t.j. ak $(x, y) \in M$ a $\theta > 0$, tak $(\theta x, \theta y) \in M$. Pomocou vstupného CCR modelu dostávame:

$$\begin{aligned}\theta^* &= \min_{\theta, \lambda} \left\{ \theta : (\theta x_o, \frac{1}{\theta} y_o) \in M_{CRS} \right\} \\ &= \min_{\theta, \lambda} \left\{ \theta : (\theta^2 x_o, y_o) \in M_{CRS} \right\} \\ &= \min_{\theta, \lambda} \left\{ \sqrt{\theta^2} : (\theta^2 x_o, y_o) \in M_{CRS} \right\} \\ &= \min_{\theta, \lambda} \left\{ \sqrt{\gamma} : (\gamma x_o, y_o) \in M_{CRS} \right\},\end{aligned}$$

pričom vzťah medzi optimálnymi hodnotami efektívít je nasledovný $\theta^* = \sqrt{\gamma^*}$. Bližšie vysvetlenie sa nachádza v spomínanom zdroji alebo v článku [14].

Ostáva však otázkou, čo v prípade variabilných výnosov z rozsahu či pri environmentálnych verziách hyperbolického modelu (11) či (12), keď obdobné úvahy využiť nemôžeme. V uvedených prípadoch totižto neboli dlhú dobu známe spoľahlivé metódy riešenia, a z toho dôvodu sa v našej práci zaoberáme dvomi vybranými spôsobmi navrhnutými len nedávno v roku 2016, konkrétne ide o už avizované články [10] a [16].

Mimo iného ešte navrhujeme našu vlastnú metódu riešenia hyperbolického modelu, pri ktorej sme vychádzali z článku [10]. Z toho dôvodu je táto kapitola rozdelená na štyri časti, pričom prvé tri sa venujú jednotlivým algoritmom predstaveným v uvedených článkoch, resp. našej vlastnej Metóde II. Posledná časť pojednáva o environmentálnych modifikáciách uvedených algoritmov, o ktorých sme čerpali informácie prevažne z článkov [11] a [15].

Opäť upozorňujeme, že v ďalšej časti práce sa zaoberáme len riešením hyperbolického modelu s variabilnými výnosmi z rozsahu či už v jeho základnej alebo environmentálnej verzii.

2.1 Metóda I

Ako prvej sa budeme venovať metóde predstavenej v článku [10], ktorá navrhuje transformáciu úlohy, definovanej vzťahmi (3)-(4), na algoritmus založený na lineárnom programovaní. Základným pilierom Metódy I je aproximácia hyperbolického modelu pomocou DDF modelu, pre ktorý existujú efektívne spôsoby riešenia, pretože vychádza z lineárneho programovania. Táto časť je obohatená aj o Obrázok 3 pochádzajúci priamo zo spomínaného zdroja, ktorým autori ilustrovali navrhnutý postup.

Ako sme už avizovali, kľúčovým pojmom pri tejto metóde je funkcia smerovej vzdialenosti alebo Directional Distance Function (DDF) opísaná v časti 1.3. V danej podkapitole je uvedený aj súvis, resp. spôsob aproximácie hyperbolického modelu, danej optimalizačnou úlohou (3)-(4), na DDF model (6) so špeciálnou voľbou vektora $\vec{g} = (g_x, g_y) = (x_o, y_o)$.

Autori teda navrhujú riešiť hyperbolický model pomocou jeho aproximácie postavenej na úlohe LP, konkrétne DDF modelom so špeciálnou voľbou vektora $\vec{g} = (g_x, g_y)$ a niekoľkých dopĺňujúcich krokov, ktoré sa pokúsime objasniť v tejto podkapitole.

V ďalšom preto opíšeme postup z článku [10], ktorý vychádza práve z primárnej a duálnej úlohy DDF modelu definovaných v časti 1.3.

2.1.1 Popis algoritmu

1. Ako prvé vyriešime primárnu aj duálnu úlohu DDF modelu, ktoré sú definované vzťahmi (6) a (8), pri počiatočnej voľbe vektora $\vec{g} = (g_x, g_y) = (x_o, y_o)$, čím dostá-

vame ich optimálne riešenia (λ^1, β^1) a (p^1, w^1, z^1) .

Dávame do pozornosti, že v prípade, ak je útvar efektívny v zmysle hyperbolického modelu definovaného v časti 1.2, je efektívny aj v zmysle modelu (6), keďže daný útvar leží priamo na hranici množiny M_{VRS} . Z toho dôvodu vieme efektívnosť v zmysle hyperbolického modelu určiť z riešenia modelu (6) hneď v prvej iterácii, presnejšie sa dívame, či je $\beta^1 = 0$. Parameter β totižto určuje efektívnosť v zmysle modelu (6) a ak je nulový, pozorujeme, že útvar DMU_o , reprezentovaný bodom (x_o, y_o) , sa už ďalej v smere \vec{g} nedá posunúť, teda leží na hranici množiny M_{VRS} . Z toho dôvodu, ak vyjde $\beta^1 = 0$, položíme $\theta^* = 1$ a algoritmus končí.

2. Ak sa však β^1 nerovná nule, čiže útvar neleží na hranici množiny M_{VRS} , využijeme súvis optimálneho riešenia (p^1, w^1, z^1) duálnej úlohy DDF modelu (8) s opornou nadrovinou množiny M_{VRS} . Ako sme už naznačili v časti 1.3, optimálne riešenie (p^1, w^1, z^1) duálnej úlohy (8) určuje opornú nadrovinu H^1 množiny M_{VRS} v bode projekcie $(x_o - \beta^1 x_o, y_o + \beta^1 y_o)$, ktorou je $H^1 : y^T p^1 - x^T w^1 - z^1 = 0$.

Nasledujúcou úlohou je teda nájsť priesečník tejto opornej nadroviny H^1 s hyperbolickou krivkou $(\theta x_o, \frac{1}{\theta} y_o)$, ktorý predstavuje aproximáciu optimálneho riešenia v danej iterácii, resp. odhad projekcie $(\theta^* x_o, \frac{1}{\theta^*} y_o)$ útvaru DMU_o na hranicu množiny M_{VRS} pre danú iteráciu. Priesečník opornej nadroviny H^1 a hyperbolickej krivky $(\theta x_o, \frac{1}{\theta} y_o)$ je bod $(\theta^1 x_o, \frac{1}{\theta^1} y_o)$, pričom hodnotu θ^1 vypočítame dosadením súradníc $(\theta^1 x_o, \frac{1}{\theta^1} y_o)$ do predpisu opornej nadroviny H^1 , čiže $H^1 : \frac{1}{\theta^1} y_o^T p^1 - \theta^1 x_o^T w^1 - z^1 = 0$. Následne prenásobíme rovnicu koeficientom θ^1 a vyjadríme hodnotu θ^1 ako koreň kvadratickej rovnice:

$$\theta^1 = \frac{-z^1 + \sqrt{(z^1)^2 + 4y_o^T p^1 x_o^T w^1}}{2x_o^T w^1}. \quad (15)$$

V tejto súvislosti je ešte dôležité podotknúť, že sa môžeme stretnúť aj s prípadom, keď skalárny súčin duálnej premennej w^1 s vektorom vstupov x_o , resp. skalárny súčin duálnej premennej p^1 s vektorom výstupov y_o , vyjde nulový. V prípade, že skalárny súčin $x_o^T w^1$ je nulový, čiže kvôli nezápornosti oboch premenných máme na mysli situáciu, keď $w_i^1 = 0$ alebo $x_{io} = 0$ pre $\forall i = 1, \dots, m$, menovateľ vo vzťahu (15) sa rovná nule. Avšak všimnime si, že rovnica vyjadrujúca priesečník opornej nadroviny H^1 a hyperbolickej krivky sa zredukovala na lineárny problém,

t.j. $H^1 : \frac{1}{\theta^1} y_o^T p^1 - z^1 = 0$. Preto θ^1 vypočítame ako $\theta^1 = \frac{y_o^T p^1}{z^1}$. Druhou možnosťou je, že skalárny súčin $y_o^T p^1$ vyjde nulový, teda pre nezápornosť oboch premenných sa $p_r^1 = 0$ alebo $y_{ro} = 0$, pre $\forall r = 1, \dots, s$. V tomto prípade sa opäť vyššie uvedená rovnica zredukuje na lineárny problém, a teda vzťah pre výpočet θ^1 je nasledovný: $\theta^1 = -\frac{z^1}{x_o^T w^1}$. Prípad, keď sa oba predmetné skalárne súčiny rovnajú nule, čiže $x_o^T w^1 = 0$ a zároveň $y_o^T p^1 = 0$, nie je potrebné bližšie analyzovať, pretože k tejto situácii nikdy nedochádza vďaka normalizačnej podmienke $g_y^T p + g_x^T w = 1$ z duálnej úlohy DDF modelu (8). Pri prvej iterácii totižto volíme vektor $(g_x, g_y) = (x_o, y_o)$, čiže sa predmetná podmienka transformuje na $y_o^T p^1 + x_o^T w^1 = 1$, čo implikuje, že ak by nastala situácia $x_o^T w^1 = 0$ a zároveň $y_o^T p^1 = 0$, uvedená podmienka by nebola splnená, pretože by sa rovnala nule a nie jednej. Pre ďalšie iterácie, keď volíme smerový vektor $(g_x^I, g_y^I) = (x_o - \theta^I, \frac{1}{\theta^I} y_o - y_o)$, dostávame po roznásobení normalizačnú podmienku $\frac{1}{\theta^I} y_o p^I - y_o^T p^I + x_o^T w^I - \theta^I x_o^T w^I = 1$, a teda ak by $x_o^T w^1 = 0$ a zároveň $y_o^T p^1 = 0$, predmetná podmienka by sa opäť nerovnala jednej, ale nule.

3. Ďalšia fáza algoritmu spočíva v preverení, či sa nájdený priesečník vyjadrujúci aproximáciu optimálneho riešenia pre danú iteráciu nachádza v množine M_{VRS} alebo nie. Ak áno, algoritmus končí a optimálnym riešením je nájdený priesečník $(\theta^1 x_o, \frac{1}{\theta^1} y_o)$. V prípade, že nie, skonštruujeme nový smerový vektor \vec{g}^1 v tvare $(g_x^1, g_y^1) = (x_o - \theta^1 x_o, \frac{1}{\theta^1} y_o - y_o)$ a opakujeme celý algoritmus dokým vypočítaný priesečník nebude ležať v množine M_{VRS} , čím sa algoritmus ukončí.

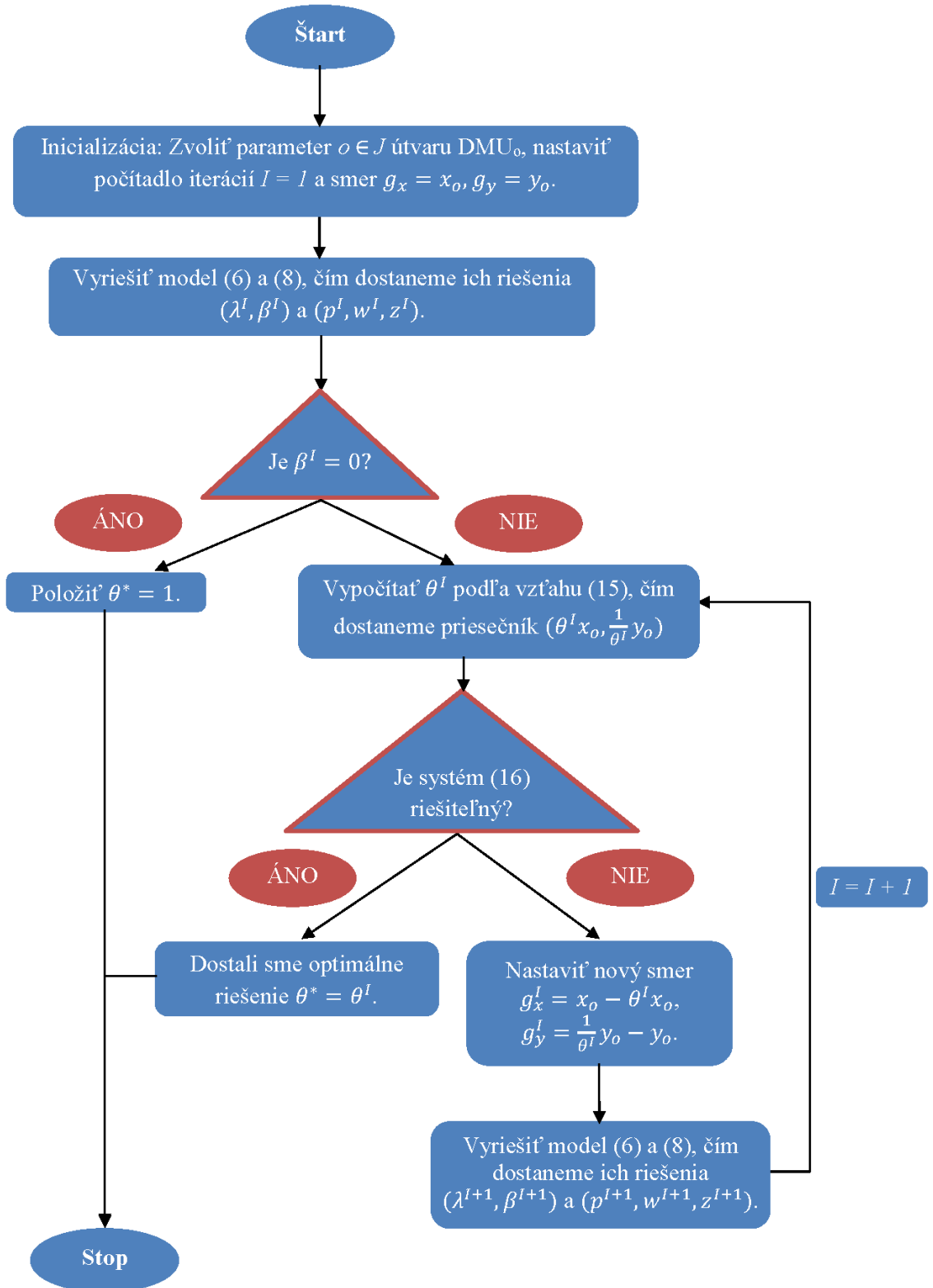
Musíme sa však ešte pozrieť na to, akým spôsobom zistiť, či nájdený priesečník leží v množine M_{VRS} . Autori článku [10] tvrdia, že uvedená úloha je ekvivalentná so zisťovaním, či priesečník patrí prieniku $H^1 \cap M_{VRS}$. Bližšie ozrejmenie sa nachádza v spomínanom článku. Dodávame však, že spomínaná úvaha vlastne vedie k preskúmaniu, či sú nasledovné podmienky prípustné, teda či existuje nejaký vektor λ ,

ktorý spĺňa nasledovné podmienky:

$$\begin{aligned}
s.t. \quad & \sum_{j \in R^1} \lambda_j x_j \leq \theta^1 x_o, \\
& \sum_{j \in R^1} \lambda_j y_j \geq \frac{1}{\theta^1} y_o, \\
& \sum_{j \in R^1} \lambda_j = 1, \\
& \lambda_j \geq 0, j \in R^1, \\
& R^1 = \{j \in J : y_j^T p^1 - x_j^T w^1 - z^1 = 0\}.
\end{aligned} \tag{16}$$

Inak povedané, množina R^1 je množina indexov tých útvarov, ktoré ležia na nadrovine H^1 . Upozorňujeme, že vzhľadom k definícii množiny R^1 nie je vektor λ v tomto prípade n -rozmerný, ale len $|R^1|$ -rozmerný.

Na koniec tejto časti ešte uvádzame Obrázok 2, ktorý predstavuje schematický prehľad algoritmu Metódy I inšpirovaný samotným článkom [10], avšak upravený v našom vlastnom spracovaní.



Obr. 2: Schematické znázornenie algoritmu Metódy I.

Zdroj: [10] a vlastné spracovanie

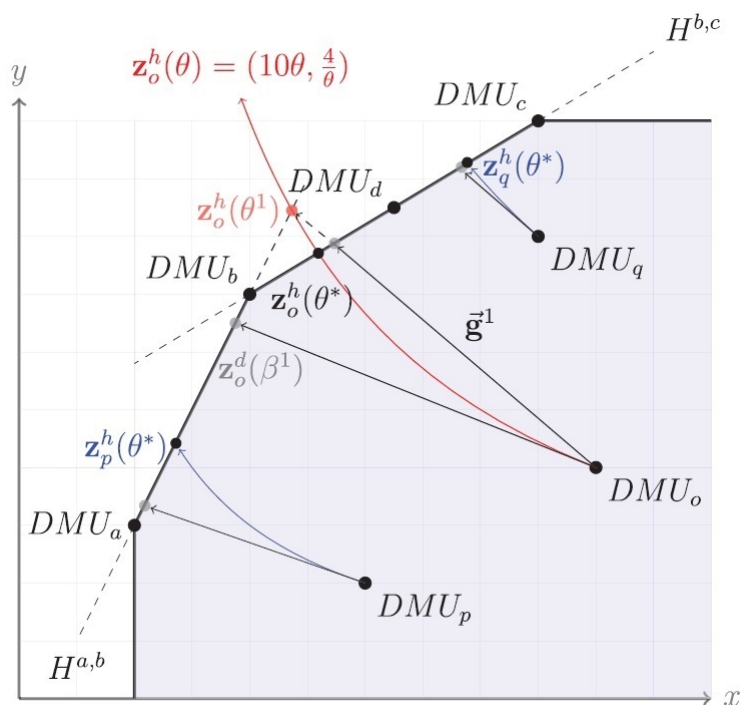
2.1.2 Ilustračný príklad

Uvedený algoritmus ozrejmujeme Obrázok 3, ktorý sme čerpali z ilustratívneho príkladu v článku [10]. Uvažujme len jeden subjekt DMU_o . Vyriešením odpovedajúcej primárnej a duálnej úlohy, t.j. úloh (6) a (8), dostávame riešenia (λ^1, β^1) , (p^1, w^1, z^1) a súčasne projekciu bodu (x_o, y_o) , na obrázku označenú ako $z_o^d(\beta^1)$, na hranicu množiny M_{VRS} , konkrétne na opornú nadrovinu množiny M_{VRS} spájajúcej útvary DMU_a a DMU_b .

Získanú opornú nadrovinu $H^1 : y^T p^1 - x^T w^1 - z^1 = 0$ označujeme aj $H^{a,b}$ podľa útvarov z hranice množiny M_{VRS} , ktoré spája. Spomínaný priesečník $(\theta^1 x_o, \frac{1}{\theta^1} y_o)$ opornej nadroviny H^1 a hyperbolickej krivky $(\theta x_o, \frac{1}{\theta} y_o)$, na obrázku vyznačená červenou, je reprezentovaný bodom $z_o^h(\theta^1)$. Parameter θ^1 získame použitím vzťahu (15).

Následne zistíme, či uvedený priesečník leží v množine M_{VRS} . Keďže tomu tak nie je, algoritmus opakujeme s upraveným vektorom $\bar{g}^1 = (g_x^1, g_y^1) = (x_o - \theta^1 x_o, \frac{1}{\theta^1} y_o - y_o)$. Ľahko si všimneme, že vektor $(-g_x^1, g_y^1)$ vlastne reprezentuje smer z útvaru DMU_o do nájdeného priesečníka.

Celý algoritmus opakujeme s novým smerovým vektorom, čiže riešime úlohy (6) a (8) s upraveným vektorom \bar{g}^1 , pričom dostávame riešenia (λ^2, β^2) a (p^2, w^2, z^2) . Nový priesečník opornej nadroviny $H^{b,c} = H^2 : y^T p^2 - x^T w^2 - z^2 = 0$ a hyperbolickej krivky $(\theta x_o, \frac{1}{\theta} y_o)$ je bod $(\theta^2 x_o, \frac{1}{\theta^2} y_o)$, pričom hodnotu θ^2 opäť vypočítame pomocou vzťahu (15). Keďže systém (16) pre úlohu z Obrázka 3 je už tentoraz prípustný, dostávame optimálne riešenie $z_o^h(\theta^*) = (\theta^2 x_o, \frac{1}{\theta^2} y_o)$ a výslednú mieru efektivity reprezentuje hodnota θ^2 , čím algoritmus končí.



Obr. 3: Znázornenie Metódy I na konkrétnom príklade.

Zdroj: [10]

2.1.3 Konvergenca algoritmu

V tejto časti bližšie rozoberieme konvergenciu Metódy I, presnejšie povedané, pozrieme sa na dôkaz konverencie Metódy I, ktorý jej autori uvádzajú v článku [10]. Ako však neskôr uvidíme, Metóda I nemusí vždy konvergovať a ani samotné argumenty autorov nie sú vždy splnené.

Autori Metódy I v článku [10] uvádzajú tvrdenie, že postupnosť hodnôt θ^I vygenerovaná Metódou I je ostro rastúca a konverguje k optimálnemu riešeniu hyperbolického modelu daného vzťahmi (3)-(4).

V naznačenom dôkaze tohto tvrdenia autori vychádzajú z konkrávnosti hranice množiny M_{VRS} a ostrej rastúcnosti hyperbolickej krivky $(\theta x_o, \frac{1}{\theta} y_o)$ vo výstupoch a ostrej klesajúcnosti vo vstupoch. Z týchto faktov autori ďalej dedukujú, že postupnosť vygenerovaných aproximácií hodnôt miery efektivity $\tilde{\theta}^I$, vypočítaná pomocou výsledkov DDF modelu ako $\tilde{\theta}^I = e^{-\beta^I}$, je ostro rastúca. Praktická časť našej práce však túto dedukciu vyvracia, pretože postupnosť hodnôt $\tilde{\theta}^I$ nevychádzala ostro rastúca. Podotýkame ešte, že ak by sme predpokladali ostrú rastúcnosť postupnosti $\tilde{\theta}^I = e^{-\beta^I}$, vyplývala

by z nej ostrá klesajúcosť postupnosti hodnôt β^I .

Autori následne tvrdia, že môžeme bez ujmy na všeobecnosti namiesto postupnosti hodnôt $\tilde{\theta}^I$ v ďalších úvahách pracovať s postupnosťou hodnôt θ^I získaných vzťahom (15), t.j. ako priesečník opornej nadroviny a hyperbolickej krivky. Ako však neskôr uvidíme, hodnoty $\tilde{\theta}^I$ a θ^I sa ale rovnať nemusia a teda uvedené postupnosti nemožno zamieňať.

Následne z týchto faktov podľa autorov priamo vyplýva, že aj postupnosť vygenerovaných hodnôt θ^I , daných vzťahom (15), je rastúca a keďže počet nadrovín definujúcich množinu M_{VRS} je konečný, aj uvedená postupnosť je konečná a navrhnutý postup dospeje k optimálnej hodnote hyperbolickeho modelu v konečnom počte krokov.

Ako uvádzame nižšie, tento dôkaz má podľa nás niekoľko nedostatkov a zároveň neskôr pozorujeme, že výsledky našich výpočtov z praktickej časti práce vyvracajú tvrdenie o konvergencii Metódy I. Dokonca, v prípade environmentálnych verzií hyperbolickeho modelu (11) a (12) ani nie je splnený predpoklad monotónnosti hyperbolickej krivky, pozdĺž ktorej projektujeme útvar na hranicu efektívnosti, resp. pseudoefektívnosti. Tejto problematike sa však podrobnejšie venujeme v časti 2.4.1.

V ďalšom uvádzame prehľad našich argumentov spochybňujúcich korektnosť dôkazu z článku [10].

1. V prvom rade opäť upozorňujeme, že predpoklad ostrej rastúcosť hodnôt $\tilde{\theta}^I$, určených ako $\tilde{\theta}^I = e^{-\beta^I}$, nebude ďalej v príkladoch splnený. Stačí sa pozrieť už len na ilustračný príklad autorov Metódy I v článku [10], ktorého výsledky autori uvádzajú v Tabuľke 2. Pre útvar DMU_o , ktorý skonvergoval až na druhú iteráciu, vychádza hodnota $\beta^1 = 0.625$ a $\beta^2 = 0.8661$. Od nich odvíjajúce sa hodnoty $\tilde{\theta}^I = e^{-\beta^I}$ sú $\tilde{\theta}^1 = 0.5353$ a $\tilde{\theta}^2 = 0.4206$, z čoho jasne vidíme, že postupnosť hodnôt $\tilde{\theta}^I = e^{-\beta^I}$ určite nie je ostro rastúca. Ak sa však pozrieme na hodnoty θ^I dopočítané zo vzťahu (15), tie už naozaj ostro rastúce sú. Predstavenú situáciu budeme pozorovať aj my v praktickej časti práce.
2. Ďalším bodom je dôležité pozorovanie, že pri takto definovanom algoritme Metódy I hodnoty $\tilde{\theta}^I = e^{-\beta^I}$ a θ^I zo vzťahu (15) nemusia byť rovnaké, a teda môžu predstavovať odlišné body na hranici množiny M_{VRS} . Prvá spomenutá hodnota $\tilde{\theta}^I$ vychádza len z výsledných hodnôt DDF modelu ako aproximácia miery efektivity $\tilde{\theta}^I = e^{-\beta^I}$,

avšak hodnota θ^I už má za cieľ predstavovať presnejší odhad miery efektivity pre danú iteráciu a je výsledkom priesečníku hyperbolickej krivky a opornej nadroviny.

Práve ostrá monotónnosť postupnosti θ^I zo vzťahu (15) je kľúčová pre konvergenciu Metódy I. Aj praktické aplikácie v časti 3 ukazujú, že ak algoritmus Metódy I konverguje, postupnosť hodnôt θ^I monotónne konverguje smerom k hodnote θ^* , hoci ju nemusí nadobúdať, ako môžeme pozorovať aj v ilustračnom príklade z časti 2.1.2. Na druhej strane postupnosť hodnôt $\tilde{\theta}^I$ aj v prípade konvergenzie Metódy I konverguje k celkom inej hodnote.

3. Autori vyvodzujú z rastúcej postupnosti hodnôt θ^I jej konvergenciu k optimálnej hodnote θ^* . Podotýkame však, že k tomu by bolo potrebné dokázať, že postupnosť θ^I je zhora ohraničená hodnotou θ^* a zároveň, že limitný bod tejto postupnosti je θ^* . Opäť poukazujeme na ilustračný príklad z časti 2.1.2, v ktorom pozorujeme, že napriek konvergencii Metódy I postupnosť θ^I nenadobudla optimálnu hodnotu θ^* .
4. Posledným argumentom voči korektnosti dôkazu konvergenzie v článku [10] je odôvodnenie konečnosti algoritmu Metódy I. Práve konečný počet oporných nadrovín je podľa autorov dôvodom konečnosti algoritmu, avšak nie je dostatočne zrejmé, že nadroviny sa nemôžu opakovať, čo by mohlo značne skomplikovať situáciu.

2.2 Metóda II

V tejto časti popíšeme a vysvetlíme nami navrhnutú modifikáciu Metódy I, ktorá má za cieľ zefektívniť a zrýchliť algoritmus, avšak zároveň je úmyslom predstavenej metódy aj jej vyššia intuitívnosť v porovnaní s Metódou I.

Aj v tomto prístupe využívame aproximáciu hyperbolickeho modelu pomocou DDF modelu (6) spolu s jeho duálnou verziou (8). Riešenie hyperbolickeho modelu teda opäť hľadáme pomocou aproximácie optimalizačného problému, daného vzťahmi (3) - (4), úlohou LP, konkrétne DDF modelom, ktorý je opísaný v časti 1.3. Na nasledujúcich riadkoch objasníme postup nami navrhnutej Metódy II so zreteľom na odlišnosti v porovnaní s Metódou I.

1. Počiatočná inicializácia metódy spolu s prvotnou fázou sú totožné ako v prípade Metódy I. Z toho dôvodu získame vyriešením modelov (6) a (8) pri špeciálnej voľbe

smerového vektora $(g_x, g_y) = (x_o, y_o)$ ich optimálne riešenia (λ^1, β^1) a (p^1, w^1, z^1) , ale taktiež už po prvej fáze disponujeme informáciou, či útvar leží na hranici množiny M_{VRS} alebo nie. Použijúc obdobné úvahy ako v časti 2.1, ak $\beta^1 = 0$, útvar sa už nedá viac posunúť smerom k hranici množiny M_{VRS} , a preto položíme $\theta^* = 1$ a algoritmus končí.

2. Druhý krok Metódy II sa taktiež nelíši od Metódy I. Z toho dôvodu v prípade, že $\beta^1 \neq 0$, čiže ak útvar nie je efektívny v zmysle DDF, a teda ani v zmysle hyperbolického modelu, dopočítame pomocou vzťahu (15) aproximáciu hodnoty θ pre danú iteráciu a označíme ju θ^1 .
3. V čom sa líši nami navrhnutá modifikácia od Metódy I je tretí krok. V Metóde I sa po vypočítaní hodnoty θ^1 preveruje, či nájdený priesečník $(\theta^1 x_o, \frac{1}{\theta^1} y_o)$ leží v množine M_{VRS} pomocou vyšetovania úlohy prípustnosti (16), ktorá v praxi znamená riešenie ďalšej úlohy lineárneho programovania. V prípade, že priesečník neleží v množine M_{VRS} , zmeníme smerový vektor \vec{g} na $\vec{g}^1 = (g_x^1, g_y^1) = (x_o - \theta^1 x_o, \frac{1}{\theta^1} y_o - y_o)$ a celý algoritmus opakujeme. Zmena smerového vektora \vec{g} vlastne znamená, že útvar DMU_o projektujeme na hranicu množiny M_{VRS} smerom k nájdenému priesečníku, presnejšie povedané smerom spojnice útvaru DMU_o a tohto priesečníka.

V Metóde II navrhujeme vynechať uvedený krok zisťovania, či daný bod leží alebo neleží v množine M_{VRS} a miesto toho rovno projektovať nájdený priesečník späť na hranicu množiny rovnakým smerovým vektorom. Predstaveným spôsobom „ušetříme“ riešenie jednej úlohy LP a použitím vzťahu (15) dostávame rovno nový priesečník $(\theta^2 x_o, \frac{1}{\theta^2} y_o)$ z ďalšej iterácie algoritmu Metódy I. Z predchádzajúceho vyplýva, že takto koncipovaná úloha sa vlastne dá sformulovať nasledovne:

$$\begin{aligned}
 & \max_{\beta, \lambda} \beta \\
 & s.t. \quad X\lambda \leq \theta^1 x_o - \beta g_x, \\
 & \quad Y\lambda \geq \frac{1}{\theta^1} y_o + \beta g_y, \\
 & \quad e_n^T \lambda = 1, \\
 & \quad \lambda \geq 0,
 \end{aligned} \tag{17}$$

kde vektor $\vec{g} = (g_x, g_y) = (x_o - \theta^1 x_o, \frac{1}{\theta^1} y_o - y_o)$. Ak optimálna hodnota β^2 úlohy

(17) vyjde nulová, znamená to, že priesečník $(\theta^1 x_o, \frac{1}{\theta^1} y_o)$ sa viac v smere \bar{g}^1 nepohol, čiže leží priamo na hranici množiny M_{VRS} . Nájdený priesečník je teda optimálnym riešením, výslednú mieru efektivity reprezentuje hodnota θ^1 a algoritmus končí.

Naopak ak je optimálna hodnota β^2 úlohy (17) nenulová, dá sa ľahko dedukovať, že pôvodný priesečník $(\theta^1 x_o, \frac{1}{\theta^1} y_o)$ leží mimo množiny M_{VRS} , a preto sa ďalej pohneme na novú opornú nadrovinu H^2 množiny M_{VRS} , ktorú by sme v prípade Metódy I dostali až v ďalšej iterácii.

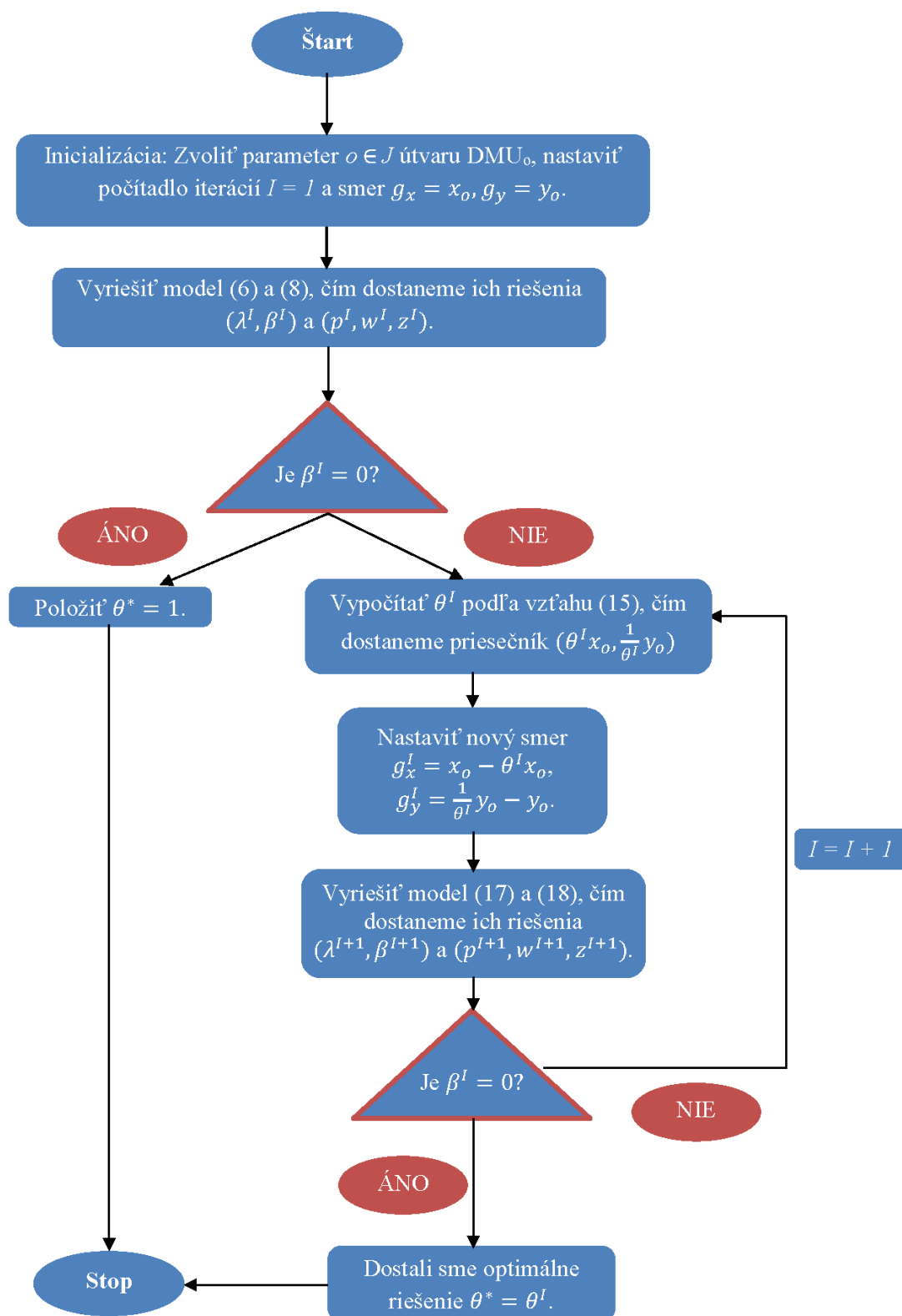
Následne opäť hľadáme priesečník $(\theta^2 x_o, \frac{1}{\theta^2} y_o)$ novonájdenej opornej nadroviny H^2 a hyperbolickej krivky $(\theta x_o, \frac{1}{\theta} y_o)$. V tejto súvislosti nepochybne treba vymedziť aj znenie odvodennej duálnej úlohy, ktorej riešenie poskytuje predpis vyššie zmienenej opornej nadroviny H^2 .

$$\begin{aligned} \max_{p,w,z} \quad & \frac{1}{\theta^1} y_o^T p - \theta^1 x_o^T w - z \\ \text{s.t.} \quad & Y^T p - X^T w - e_n z \leq 0, \\ & g_y^T p + g_x^T w = 1, \\ & p \geq 0_s, w \geq 0_m, \end{aligned} \tag{18}$$

kde e_n je n -rozmerný vektor jednotiek.

Uvažujme teraz riešenia (λ^2, β^2) a (p^2, w^2, z^2) úloh (17) a (18). Tieto následne využijeme, dosadením do vzťahu (15), na výpočet hodnoty θ^2 . Vzhľadom k charakteru modifikovanej úlohy (18) nie je potrebné meniť daný vzťah. Následne pomocou získanej hodnoty θ^2 transformujeme vektor \bar{g}^1 , v ktorého smere projektujeme útvar na hranicu množiny M_{VRS} , na $\bar{g}^2 = (g_x^2, g_y^2) = (x_o - \theta^2 x_o, \frac{1}{\theta^2} y_o - y_o)$, a opäť riešime úlohy (17) a (18). Tento cyklus opakujeme, kým nevyjde optimálna hodnota β^I úlohy (17) rovná nule, čo znamená, že sa nájdený priesečník viac nepohol, a teda ležal na hranici množiny M_{VRS} .

Aj v tomto prípade uvádzame pre lepšie pochopenie schematický prehľad rozoberaného algoritmu.



Obr. 4: Schematické znázornenie Metódy II.

Zdroj: Vlastné spracovanie

2.3 Metóda III

Poslednou sledovanou metódou, ktorou sa v tejto časti venujeme, je metóda predstavená v článku [16] využívajúca podstatne odlišný spôsob riešenia hyperbolického modelu ako v predchádzajúcich dvoch prípadoch. Uvedený článok sa venuje primárne riešeniu iného nelineárneho DEA modelu, konkrétne Russellovho modelu, avšak autorky aplikujú danú metódu aj na hyperbolický model v článku [14] alebo aj na prípad environmentálnej verzie hyperbolického modelu v článku [15].

V tejto časti ďalej povieme pár slov o Metóde III a uvidíme dôkaz z článku [14] o rovnosti optimálnej hodnoty θ^* hyperbolického modelu, definovaného vzťahmi (3)-(4), a optimálnej hodnoty $\hat{\theta}$ vypočítanej Metódou III.

Algoritmus sformulovaný v týchto článkoch sa zakladá na priamom riešení hyperbolického modelu (3)-(4) pomocou jeho SDP formulácie. Na rozdiel od predošlých postupov teda pozostáva len z jedného kroku potrebného k zisteniu optimálnej hodnoty θ bez potreby ďalších výpočtov, ktoré algoritmus môžu značne spomaliť a skomplikovať. Ako príklad spomenieme hľadanie predpisu opornej nadroviny či dopočítavanie hodnoty θ^I pomocou vzťahu (15).

V spomenutých článkoch je hyperbolický model transformovaný pomocou zavedenia novej premennej $\phi = \frac{1}{\theta}$ a relaxáciou uvedenej nekonvexnej rovnosti na hyperbolickú nerovnosť $\phi\theta \geq 1$. Všimnime si, že na túto nerovnosť sa dá hľadiť aj ako na vyšetovanie nezápornosti determinantu 2×2 matice P s prvkami ϕ, θ na diagonále a s jednotkami na zvyšných miestach. Spolu s predpokladom nezápornosti diagonálnych prvkov matice P je táto podmienka ekvivalentná s podmienkou kladnej semidefinitnosti matice P , t.j. $P \succeq 0$. Tieto úpravy preto vedú k nasledovnému optimalizačnému problému

postavenom na teórii SDP:

$$\begin{aligned}
 & \min_{\theta, \lambda, \phi} \theta \\
 & s.t. \quad X\lambda \leq \theta x_o, \\
 & \quad Y\lambda \geq \phi y_o, \\
 & \quad P := \begin{pmatrix} \theta & 1 \\ 1 & \phi \end{pmatrix} \succeq 0, \\
 & \quad e_n^T \lambda = 1, \\
 & \quad \lambda \geq 0.
 \end{aligned} \tag{19}$$

Relaxáciou pôvodného optimalizačného problému (3)-(4) úlohou (19) síce zväčšíme množinu prípustných riešení, ale na druhej strane získame úlohu konvexnej optimalizácie, pre ktorej riešenie už v dnešnej dobe existujú efektívne algoritmy. Zároveň však vyzdvihujeme ďalší prínos Metódy III, a tým je nasledujúce tvrdenie spolu s existenciou jeho explicitného dôkazu:

Tvrdenie 2.1. *Optimálna hodnota θ^* úlohy (3)-(4) sa rovná optimálnej hodnote $\hat{\theta}$ úlohy (19).*

Čerpajúc z článku [14], uvádzame na nasledujúcich riadkoch dôkaz Tvrdenia 2.1.

Dôkaz. Predpokladajme, že (θ^*, λ^*) je optimálne riešenie modelu (3)-(4) a $(\hat{\theta}, \hat{\phi}, \hat{\lambda})$ je optimálne riešenie úlohy (19). Potom nepochybne môžeme o $(\theta^*, \frac{1}{\theta^*}, \lambda^*)$ tvrdiť, že je prípustným riešením úlohy (19), a teda platí $\hat{\theta} \leq \theta^*$. Naopak, ak vezmeme optimálne riešenie modelu (19), tak z podmienok tohto modelu vyplýva, že $Y\hat{\lambda} \geq \hat{\phi}y_o \geq \frac{1}{\hat{\theta}}y_o$, pričom vychádzame z nezápornosti vektora y_o a parametra $\hat{\phi}$ a z poznatku, že $\hat{\phi} \geq \frac{1}{\hat{\theta}}$. Z predchádzajúceho sa teda dá usúdiť, že $(\hat{\theta}, \hat{\lambda})$ je prípustným riešením (3)-(4), a preto $\hat{\theta} \geq \theta^*$. Predchádzajúce závery, t.j. $\hat{\theta} \leq \theta^*$ a $\hat{\theta} \geq \theta^*$, teda implikujú, že $\hat{\theta} = \theta^*$. \square

2.4 Environmentálne modifikácie metód

V nasledujúcej podkapitole rozoberáme, akým spôsobom sa predchádzajúce tri metódy modifikujú v prípade uvažovania environmentálnej verzie hyperbolického modelu, definovaného úlohami (11), resp. (12). Podkapitola je preto rozdelená na tri časti, pričom

každá z nich sa venuje úprave práve jednej zo skúmaných metód z častí 2.1, 2.2 a 2.3 pre potreby zahrnutia nežiadúcich výstupov do hyperbolického modelu.

2.4.1 Metóda I v prípade nežiadúcich výstupov

V prvej časti tejto podkapitoly presnejšie popíšeme environmentálnu verziu algoritmu Metódy I na riešenie hyperbolického modelu v kontexte environmentálnej problematiky.

Ako sme už uviedli, čerpáme najmä z článkov [15] a [11], pričom na druhý menovaný sa odvolávajú autori Metódy I v článku [10]. Vzhľadom k tomu, že v zdroji [10] autori uvádzajú len algoritmus Metódy I bez nežiadúcich výstupov, odvodili sme zvyšné vzťahy potrebné k algoritmu.

Základná myšlienka postupu je analogická ako v prípade Metódy I z časti 2.1, a teda ústredným pojmom tejto metódy je DDF model, konkrétne jeho environmentálna verzia definovaná vzťahom (13) v prípade technologickej množiny M_0 , resp. jeho upravenej verzie v prípade technologickej množiny M_1 , ktorým sme sa venovali v časti 1.4.

1. Prvým krokom je opäť nájdenie optimálnych riešení (λ^1, β^1) a (p^1, p_B^1, w^1, z^1) primárnej a duálnej úlohy DDF modelu, ktoré sú definované vzťahmi (13) a (14), resp. berúc do úvahy technologickú množinu M_1 riešime ich upravené verzie.

Upozorňujeme však, že v uvedených modeloch volíme najprv špeciálny smerový vektor $\vec{g} = (g_y, g_b) = (y_o, b_o)$. Keďže metóda má za cieľ riešiť hyperbolický model definovaný vzťahmi (11), resp. (12), v x -ovom smere sa vôbec nehýbeme. Z uvedeného vyplýva, že sledovaný útvar DMU_o posúvame v rámci DDF modelu len v smere skracovania nežiadúcich výstupov a predlžovania žiadúcich výstupov.

Aj v prípade environmentálnej verzie hyperbolického modelu vieme z nájdeného riešenia primárnej úlohy DDF modelu určiť, či je útvar efektívny v zmysle DDF či hyperbolického modelu alebo nie. Ak teda $\beta^1 = 0$, útvar leží na hranici množiny M_0 , resp. M_1 , položíme $\theta^* = 1$ a algoritmus sa ukončí.

2. Ak $\beta^1 \neq 0$, dopočítame aproximáciu θ^1 hodnoty θ pre danú iteráciu pomocou nižšie odvodeného vzťahu (20). Pri odvodení predpisu pre výpočet θ^1 opäť vychádzame z myšlienky priesečníku opornej nadroviny H^1 a hyperbolickej krivky, ktoré však majú v environmentálnom kontexte mierne odlišný tvar. Oporná nadrovina H^1

v tvare $H^1 : y^T p^1 - x^T w^1 - b^T p_B^1 - z^1 = 0$ pretína hyperbolickú krivku $(x_o, \frac{1}{\theta} y_o, \theta b_o)$ v bode $(x_o, \frac{1}{\theta^1} y_o, \theta^1 b_o)$, ktorý zistíme vyjadrením parametra θ^1 ako koreň kvadratickej rovnice nasledovne:

$$\theta^1 = \frac{-z^1 - x_o^T w^1 + \sqrt{(z^1 + x_o^T w^1)^2 + 4y_o^T p^1 b_o^T p_B^1}}{2b_o^T p_B^1}. \quad (20)$$

Opäť je potrebné vziať do úvahy aj situácie, keď jeden zo skalárnych súčinov $y_o^T p^1$ alebo $b_o^T p_B^1$ vyjde nulový. Použijúc obdobné úvahy ako v časti 2.1 dostávame vzťah $\theta^1 = -\frac{z^1 - x_o^T w^1}{b_o^T p_B^1}$ pre prípad nulového skalárneho súčinu $y_o^T p^1$ a predpis $\theta^1 = -\frac{y_o^T p^1}{z^1 + x_o^T w^1}$, ak vyjde nulový druhý skalárny súčin $b_o^T p_B^1$. Ani v tomto prípade, vďaka normalizačnej podmienke $g_y^T p + g_b^T p_B = 1$, nikdy nedochádza k súčasnej nulovosti oboch skalárnych súčinov $y_o^T p^1$ a $b_o^T p_B^1$. Z toho dôvodu túto situáciu bližšie nerozoberáme. Dodávame ešte, že v prípade nulového skalárneho súčinu $x_o^T w^1$ nie je potrebné meniť vyššie uvedený vzťah (20).

3. Podstata nasledujúcej fázy algoritmu opäť tkvie v overení, či nájdený priesečník reprezentujúci aproximáciu riešenia leží v množine M_0 , resp. M_1 . Pre technologickú množinu M_0 , uvedené testujeme analogicky upravenou úlohou prípustnosti:

$$\begin{aligned} s.t. \quad & \sum_{j \in R^1} \lambda_j x_j \leq x_o, \\ & \sum_{j \in R^1} \lambda_j y_j \geq \frac{1}{\theta^1} y_o, \\ & \sum_{j \in R^1} \lambda_j b_j = \theta^1 b_o, \\ & \sum_{j \in R^1} \lambda_j = 1, \\ & \lambda_j \geq 0, j \in R^1, \\ & R^1 = \{j \in J : y_j^T p^1 - x_j^T w^1 - b_j^T p_B^1 - z^1 = 0\} \end{aligned} \quad (21)$$

Vyšetrujeme teda, či existuje nejaký $|R^1|$ -rozmerný vektor λ , ktorý by spĺňal podmienky (21). V prípade technologickej množiny M_1 je miesto tretieho obmedzenia nerovnosť $\sum_{j \in R^1} \lambda_j b_j \leq \theta^1 b_o$.

Ak je uvedená úloha prípustná, a teda vypočítaný priesečník leží v množine M_0 , resp. M_1 , algoritmus končí a optimálnym riešením je práve tento priesečník

$(x_o, \frac{1}{\theta^1}y_o, \theta^1b_o)$ s mierou efektivity θ^1 . V opačnom prípade však musíme skonštruovať nový smerový vektor $\vec{g}^1 = (g_y^1, g_b^1) = (\frac{1}{\theta^1}y_o - y_o, b_o - \theta^1b_o)$ a opakujeme celý algoritmus dokým vypočítaný priesečník nebude ležať v množine M_0 , resp. M_1 , čím sa algoritmus ukončí a my dostávame výslednú aproximáciu optimálneho riešenia.

V súvislosti s Metódou I v prípade nežiadúcich výstupov považujeme za nutné ešte spomenúť, že v prípade environmentálnych verzií hyperbolického modelu (11) a (12) už nie sú splnené všetky argumenty, ktoré autori článku [10] využívajú v dôkaze konverencie ich algoritmu. Predpokladajú totižto, že krivka $z_o^h(\theta) = (\theta x_o, \frac{1}{\theta}y_o)$ je rýdzo monotónna, avšak v prípade nežiadúcich výstupov pracujeme s modelmi (11), resp. (12), z ktorých vyplýva, že predmetná krivka má tvar $z_o^h(\theta) = (x_o, \frac{1}{\theta}y_o, \theta b_o)$, a teda nie je rýdzo monotónna v premennej x . Ako neskôr uvidíme, Metóda I v prípade nežiadúcich výstupov bude mať naozaj aj v menších dátových súboroch problém s konvergenciou, čo môže byť dôsledok aj tejto skutočnosti.

2.4.2 Metóda II v prípade nežiadúcich výstupov

Vzhľadom k environmentálnej problematike, ktorej sa v našej práci venujeme, predstavíme taktiež upravenú verziu Metódy II práve pre prípad nežiadúcich výstupov. Keďže v časti 2.4.1 obsérne popisujeme úpravu Metódy I na použitie v environmentálnej problematike a tiež z dôvodu, že odvodenie environmentálnej verzie Metódy II z jej základnej podoby definovanej v časti 2.2 je analogické, nevenujeme tejto časti toľko priestoru ako pri predchádzajúcich metódach.

1. Po inicializácii metódy a vyriešení úlohy (13) so špeciálnou voľbou vektora \vec{g} , konkrétne $\vec{g} = (g_y, g_b) = (y_o, b_o)$ v prípade uvažovania technologickej množiny M_0 , resp. v prípade technologickej množiny M_1 berúc do úvahy odpovedajúcu úpravu danej úlohy, sa dívame na to, či subjekt leží na hranici množiny alebo nie. V prípade ak áno, položíme $\theta^* = 1$ a algoritmus ukončíme.
2. Ak tomu tak nie je, vypočítame pomocou riešenia duálnej úlohy (14), resp. jej odpovedajúcej verzie v prípade technologickej množiny M_1 , a vzťahu (20) priesečník opornej nadroviny H^1 a hyperbolickej krivky $(x_o, \frac{1}{\theta}y_o, \theta b_o)$. Dodávame, že v prípade

nulovosti jedného zo skalárnych súčinov $y_o^T p^1$ alebo $b_o^T p_B^1$, platia obdobné úvahy ako v časti 2.4.1.

3. Vypočítaný priesečník $(x_o, \frac{1}{\theta^1} y_o, \theta^1 b_o)$ projektujeme späť na hranicu množiny M_0 , definovanej vzťahom (9), v smere $(0, \frac{1}{\theta^1} y_o - y_o, b_o - \theta^1 b_o)$, čiže riešime nasledovnú optimalizačnú úlohu:

$$\begin{aligned} \max_{\beta, \lambda} \quad & \beta \\ \text{s.t.} \quad & X\lambda \leq x_o, \\ & Y\lambda \geq \frac{1}{\theta^1} y_o + \beta g_y, \\ & B\lambda = \theta^1 b_o - \beta g_b, \\ & e_n^T \lambda = 1, \\ & \lambda \geq 0, \end{aligned} \tag{22}$$

kde vektor $\vec{g} = (g_y^1, g_b^1) = (\frac{1}{\theta^1} y_o - y_o, b_o - \theta^1 b_o)$. V prípade technologickej množiny M_1 nahrádzame tretie obmedzenie nerovnosťou $B\lambda \leq \theta^1 b_o - \beta g_b$. Všimnime si, že sa opäť jedná o mierne modifikovaný DDF model, pri ktorom neprojektujeme útvar DMU_o , ale nájdený priesečník na hranicu efektívnosti, resp. na hranicu pseudoefektívnosti.

Ak je optimálna hodnota β^2 úlohy (22) rovná nule, nájdený priesečník leží na hranici množiny, a teda už jeho ďalší posun nie je možný. Z čoho poľahky vyplýva, že tento predstavuje optimálne riešenie a algoritmus končí.

Analogicky so základnou verziou Metódy II, ak hodnota β^2 úlohy (22) je nenulová, nájdený priesečník $(x_o, \frac{1}{\theta^1} y_o, \theta^1 b_o)$ leží mimo technologickej množiny, a teda sa posunie na opornú nadrovinu H^2 množiny M_0 , resp. M_1 . Predpis H^2 zohráva dôležitú úlohu v hľadaní nového priesečníka $(x_o, \frac{1}{\theta^2} y_o, \theta^2 b_o)$ a získame ho z riešenia duálnej úlohy v tvare:

$$\begin{aligned} \max_{p, w, z} \quad & \frac{1}{\theta^1} y_o^T p - x_o^T w - \theta^1 b_o^T p_B - z \\ \text{s.t.} \quad & Y^T p - X^T w - B^T p_B - e_n z \leq 0, \\ & g_y^T p + g_b^T p_B = 1, \\ & p \geq 0_s, w \geq 0_m, \end{aligned} \tag{23}$$

kde e_n je n -rozmerný vektor jednotiek. Uvedenú úlohu riešime v prípade, ak uvažujeme technologickú množinu M_0 , v prípade technologickej množiny M_1 pridávame podmienku $p_B \geq 0_p$.

Využijúc vzťah (20), dostávame novú hodnotu θ^2 a s ňou aj nový priesečník. Následne modifikujeme smerový vektor $\bar{g}^2 = (g_y^2, g_b^2) = (\frac{1}{\theta^2}y_o - y_o, b_o - \theta^2 b_o)$ a s takto upraveným vektorom opäť riešime úlohy (22) a (23). Uvedený postup opakujeme do momentu, kým optimálna hodnota β^I úlohy (22) nevyjde nulová, čo bude signalizovať, že novonájdenný priesečník $(x_o, \frac{1}{\theta^I}y_o, \theta^I b_o)$ leží na hranici množiny M_0 , resp. M_1 , a teda sa viac nepohol smerom k hranici množiny. V danom prípade dostávame optimálne riešenie s výslednou mierou efektivity a algoritmus sa ukončí.

2.4.3 Metóda III v prípade nežiadúcich výstupov

Posledná časť tejto kapitoly je venovaná environmentálnej verzii Metódy III. Vychádzajúc z hyperbolického modelu s nežiadúcimi výstupmi, daného úlohami (11) pre technologickú množinu M_0 , prípadne (12) pre M_1 , modifikujeme Metódu III podľa práve uvažovanej technologickej množiny. Aplikácii Metódy III v kontexte environmentálnej problematiky sa venujú napríklad aj autorky článku [15], z ktorého vychádzame aj my pri odvodení environmentálnej verzie Metódy III.

Zahrnutím nežiadúcich výstupov do analýzy sa zmení úloha SDP Metódy III v prípade technologickej množiny M_0 definovanej vzťahom (9) nasledovným spôsobom:

$$\begin{aligned}
 & \min_{\theta, \lambda, \phi} \theta \\
 & s.t. \quad X\lambda \leq x_o, \\
 & \quad Y\lambda \geq \phi y_o, \\
 & \quad B\lambda = \theta b_o, \\
 & \quad P := \begin{pmatrix} \theta & 1 \\ 1 & \phi \end{pmatrix} \succeq 0, \\
 & \quad e_n^T \lambda = 1, \\
 & \quad \lambda \geq 0,
 \end{aligned} \tag{24}$$

Ak uvažujeme technologickú množinu M_1 danú vzťahom (10), miesto tretieho obmedzenia použijeme nerovnosť $B\lambda \leq \theta b_o$.

3 Praktická časť

V poslednej kapitole sa venujeme predovšetkým empirickému porovnávaniu jednotlivých metód riešení hyperbolického modelu na dátových súboroch rôznej veľkosti. Primárne pozorujeme odlišnosti v hodnotách mier efektívít, ale aj samotnú rýchlosť výpočtov jednotlivých metód. Výsledky sme získali vlastnými výpočtami v softvéri MATLAB, konkrétne pomocou knižnice CVX, v ktorom sme naprogramovali jednotlivé metódy podľa ich definícií uvedených v teoretickej časti. V prílohách A.1, A.2 a A.3 sú uvedené zdrojové kódy použité pre výpočty.

Pre upresnenie dodávame, že rýchlosť algoritmov sme merali pomocou funkcie

```
1 tic
2 t=toc;
```

čo znamená, že výsledné hodnoty sa nedajú interpretovať ako konkrétna časová jednotka, napr. počet sekúnd alebo minút. Uvedená funkcia meria interný čas potrebný na vykonanie daných príkazov a má zmysel jej výsledné hodnoty porovnávať relatívne medzi jednotlivými algoritmami.

V prípade Metódy I sme zdrojový kód zostavovali na základe postupu definovaného v časti 2.1 a teoretické podklady pre environmentálnu verziu sú uvedené v podkapitole 2.4.1. Našu vlastnú Metódu II sme implementovali na základe častí 2.2 a 2.4.2, v prípade nežiadúcich výstupov. Ich výsledky následne porovnávame s výsledkami získanými pomocou Metódy III, ktorú sme naprogramovali podľa vzťahov uvedených v časti 2.3, respektíve podľa časti 2.4.3 pri environmentálnej verzii modelu.

Ako sme už uviedli na začiatku Kapitoly 2, pre prípad modelu s CRS bez nežiadúcich výstupov už efektívny a spoľahlivý spôsob riešenia existuje dlhšie. Dalo by sa tým pádom pozrieť na fungovanie skúmaných metód aj pre prípad CRS s nežiadúcimi výstupmi, v našej práci však berieme do úvahy len prípad variabilných výnosov z rozsahu.

Táto kapitola je rozdelená na tri časti, pričom prvá je venovaná základnej verzii hyperbolického modelu, v druhej rozoberáme jeho environmentálnu modifikáciu a v tretej pozorujeme výsledky jednotlivých metód na veľkých dátových súboroch, v ktorých sa počet útvarov pohybuje rádovo v tisícoch.

3.1 Základné verzie metód

Ako sme už avizovali, najprv sa venujeme metódam riešenia základnej verzie hyperbolického modelu, čomu odpovedajú aj zvolené dátové súbory. Základné informácie o počte útvarov a počte vstupov a výstupov sú uvedené v Tabuľke 1. Bližšie informácie o dátach sa dajú nájsť v zdrojoch uvedených v spomínanej tabuľke.

Tabuľka 1: Súhrnné informácie o dátových súboroch prislúchajúcich k základnej verzii hyperbolického modelu s VRS

Názov dátového súboru	Aerolínie	PC spoločnosti	US školy
Zdroj	[1]	[17]	[18]
Počet DMU	19	20	70
Počet vstupov	4	3	5
Počet výstupov	2	1	3

V nasledovnej Tabuľke 2 pozorujeme základné výsledky týkajúce sa rýchlosti algoritmu v prepočte na 1 útvar a počet rozdielnych výsledkov. Poznamenajme, že sme v žiadnom z príkladov nepozorovali signifikantne odlišnú hodnotu efektivity pri jednotlivých metódach riešenia. Pre presnosť podotýkame, že pod signifikantnosťou rozdielov máme na mysli hodnoty aspoň 10^{-6} a väčšie. V našom prípade však rozdiely v hodnotách efektivity medzi Metódou I, resp. Metódou II, a Metódou III boli maximálne rádovo v 10^{-8} v prípade všetkých troch dátových súborov. Odlišnosti vo výsledkoch Metódy I a Metódy II v tomto prípade nepozorujeme.

Tabuľka 2: Základné informácie o výsledkoch prislúchajúcich k základnej verzii hyperbolického modelu s VRS

Názov dátového súboru	Aerolínie	PC spoločnosti	US školy
Čas na 1 DMU - Metóda I	0,6089	0,9214	1,1310
Čas na 1 DMU - Metóda II	0,5399	0,8192	1,1305
Čas na 1 DMU - Metóda III	0,4078	0,4319	0,4861
Počet rozdielnych výsledkov	0	0	0

Ak sa pozrieme na počet iterácií potrebných k skonvergovaní algoritmov Metódy I,

resp. Metódy II, pozorujeme, že oba algoritmy vždy skonvergovali už po prvej iterácii. Avšak ako neskôr uvidíme, nie je tomu tak vždy. Dôvodom pre takto úspešné výsledky môže byť aj malý počet útvarov a málo dátových súborov, na ktorých sme metódy testovali. Z toho dôvodu sa ďalej budeme venovať ešte pozorovaniu a porovnávaniu oboch metód pri komplexnejších a obsiahlejších dátových súboroch. Čo sa týka Metódy III, tá tiež skonvergovala pre všetky útvary pri všetkých dátových súboroch.

Všimnime si tiež časy výpočtov na jeden útvar pri Metóde I a II. Tieto časy sa od seba markantne nelíšia, avšak môže to byť spôsobené práve tým, že vo všetkých dátových súboroch skonvergovali algoritmy hneď po prvej iterácii, a tak prirodzene neočakávame veľkú časovú úsporu. V prípade, že by bolo potrebných k nájdeniu optimálneho riešenia viac iterácií, sa dá predpokladať, že by bolo zrýchlenie algoritmu Metódy II výraznejšie. Čo však vieme jednoznačne zhodnotiť, je výrazná časová úspora v prípade Metódy III, ktorú budeme pozorovať aj v ďalších príkladoch.

Nedá sa nespomenúť ešte jeden podstatný fakt. V Metóde I aj v Metóde II je potrebné určiť presnosť identifikovania nuly, napríklad pri testovaní nulovosti β . Keďže Matlab, resp. jeho nadstavba CVX, využívajú na riešenie optimalizačných úloh určité numerické metódy, výsledná hodnota nemusí byť presne rovná nule, a preto sme ako východiskovú presnosť volili 10^{-6} . Ako však neskôr uvidíme, pri tejto hladine presnosti nie sú skúmané Metódy I a II vždy schopné skonvergovať. Žiadnej zo základných verzií metód však táto úroveň presnosti nerobila problém a pri všetkých dátových súboroch metódy skonvergovali pri počiatočnom nastavení presnosti 10^{-6} .

3.2 Environmentálne verzie

V tejto podkapitole zhrnieme naše pozorovania z porovnávania environmentálnych verzií metód riešenia hyperbolického modelu. Z toho dôvodu už nasledujúce príklady zahŕňajú aj nežiadúce výstupy. Súhrnné informácie o počte útvarov a počte vstupov a výstupov, či už žiadúcich alebo nežiadúcich, uvádzame v Tabuľke 3. Pre detailnejšie informácie o dátových súboroch sa dá nahliadnuť do zdrojov z tabuľky.

Tabuľka 3: Súhrnné informácie o dátových súboroch prislúchajúcich k environmentálnej verzii hyperbolického modelu s VRS

Názov dátového súboru	Elektrárne	Školy
Zdroj	[8]	[12]
Počet DMU	92	96
Počet vstupov	3	5
Počet žiadúcich výstupov	1	2
Počet nežiadúcich výstupov	2	1

Dátový súbor o elektrárnach použili pri ilustrácii metódy aj autori v článku [10], v ktorom bola predstavená Metóda I, z toho dôvodu sme tiež porovnávali naše výsledky s uvedenými v článku. Je dôležité podotknúť, že autori používali softvér GAMS a nemerali rýchlosť ich metódy, len počet iterácií a samotnú hodnotu efektivity.

V prípade environmentálnej modifikácie hyperbolického modelu máme k dispozícii dve verzie definované modelmi (11) a (12), ktoré sa líšia v uvažovanej technologickej množine. Pozreli sme sa preto na oba varianty. Dopĺňame ešte, že autori článku [10] riešili len model (11).

V nasledovnej Tabuľke 4 pozorujeme základné výsledky týkajúce sa rýchlosti algoritmu každého z troch metód v prepočte na 1 útvar a počet rozdielnych výsledkov. Vzhľadom ku skutočnosti, že pri dátovom súbore o elektrárnach algoritmy pomocou Metódy I a II pre pár útvarov neskonvergovali, čas na 1 útvar je v týchto prípadoch vypočítaný len na 1 skonvergovaný útvar.

Za rozdielne výsledky pokladáme nielen odlišnosti výsledných mier efektív v prípade skonvergovaných útvarov, ale aj udalosť, ak útvar pre aspoň jednu z metód neskonvergoval. Presnejšie povedané, problematika neskonvergovaného algoritmu sa týka len Metód I a II, pretože Metóda III skonvergovala pri všetkých dátových súboroch pre všetky útvary.

Ďalej podotýkame, že najväčší pozorovaný rozdiel v hodnote efektivity skonvergovaných útvarov medzi Metódou I, resp. Metódou II, a Metódou III bol rádovo v hodnote 10^{-5} aj to len v prípade modelu (12) pri dátovom súbore so školami. Z toho dôvodu namerané odlišnosti mier efektív skonvergovaných útvarov nepovažujeme za relevantné,

a teda ich neberieme do úvahy pri celkovom počte odlišných výsledkov. Z toho vyplýva, že v tomto prípade sú v celkovom počte odlišných výsledkov zahrnuté len neskonvergované útvary, avšak ako neskôr uvidíme, rozdiely vo výsledných hodnotách efektívít nebudú vždy až také nízke. Neskôr však uvádzame aj detailnejšiu analýzu nameraných odlišností výsledkov jednotlivých metód.

Tabuľka 4: Základné informácie o výsledkoch prislúchajúcich k environmentálnej verzii hyperbolického modelu s VRS

Typ modelu	Model (11)		Model (12)	
Názov dátového súboru	Elektrárne	Školy	Elektrárne	Školy
Čas na 1 DMU - Metóda I	1,5268	1,463	1,503	1,4839
Čas na 1 DMU - Metóda II	1,221	1,2733	1,2138	1,0669
Čas na 1 DMU - Metóda III	0,4439	0,475	0,4339	0,4952
Počet rozdielných výsledkov	2	0	1	0

Opäť pozorujeme výraznú časovú úsporu v prípade Metódy III, avšak už o niečo miernejšiu v prípade Metódy II v porovnaní s Metódou I.

Uverejnené výsledky autorov článku [10] k dátovému súboru s elektrárnami sa týkali modelu (11) a v porovnaní s našimi výsledkami sa mimo neskonvergovaných útvarov líšili maximálne rádovo o 10^{-5} . Upozorňujeme ale na fakt, že autori uvádzali výsledky mier efektivity zaokrúhlené na 5 desatinných miest, čo mohlo mierne ovplyvniť výsledné rozdiely. V uvedených výsledkoch však autorom skonvergovali všetky útvary a pri tých útvaroch, ktoré v našom prípade neskonvergovali, sa nachádza rovnaká hodnota efektivity ako v prípade Metódy III.

Kvôli lepšej predstave čitateľa o odlišnostiach výsledkov jednotlivých metód uvádzame základnú analýzu nameraných odchýlok v Tabuľkách 5 a 6. V tabuľkách pozorujeme maximálnu, minimálnu a priemernú absolútnu odchýlku medzi jednotlivými metódami a taktiež súčet absolútnych odchýlok. Dávame do pozornosti celkovo menšie odchýlky v prípade modelu (11). Najprv uvádzame analýzu pre dátový súbor s elektrárnami a následne pre dátový súbor týkajúci sa škôl.

Tabuľka 5: Analýza rozdielov výsledkov pre dátový súbor Elektrárne - uvažujúc VRS

Porovnávané metódy		Max	Min	Avg	Sum
Model (11)	I a III	$3,75 \times 10^{-8}$	$8,26 \times 10^{-13}$	$8,55 \times 10^{-9}$	$7,69 \times 10^{-7}$
	II a III	$3,75 \times 10^{-8}$	$8,26 \times 10^{-13}$	$8,45 \times 10^{-9}$	$7,69 \times 10^{-7}$
	I a II	$1,32 \times 10^{-9}$	0	$7,20 \times 10^{-11}$	$6,48 \times 10^{-9}$
Model (12)	I a III	$2,53 \times 10^{-6}$	$9,90 \times 10^{-11}$	$6,40 \times 10^{-8}$	$5,82 \times 10^{-6}$
	II a III	$1,72 \times 10^{-6}$	$9,90 \times 10^{-11}$	$7,33 \times 10^{-8}$	$6,74 \times 10^{-6}$
	I a II	$3,29 \times 10^{-6}$	0	$7,98 \times 10^{-8}$	$7,26 \times 10^{-6}$

Vzhľadom k vyššie spomenutému faktu, že autori článku [10] pri testovaní Metódy I riešenia modelu (11) uvádzali výsledky miery efektivity zaokrúhlené na 5 desatinných miest, analýzu odchýlok v porovnaní s ich výsledkami kvôli možným skresleniam neuvádzame. Pri zaokrúhlení našich výsledkov na 5 desatinných miest, boli všetky odchýlky skonvergovaných útvarov nulové až na jednu výnimku. Odchýlky medzi našimi výsledkami všetkých troch sledovaných metód boli pre predmetný útvar síce nulové, ale od hodnoty vypočítanej autormi v článku [10] sa líšili o 5×10^{-5} .

Tabuľka 6: Analýza rozdielov výsledkov pre dátový súbor Školy - uvažujúc VRS

Porovnávané metódy		Max	Min	Avg	Sum
Model (11)	I a III	$2,98 \times 10^{-7}$	$2,94 \times 10^{-12}$	$1,12 \times 10^{-8}$	$1,07 \times 10^{-6}$
	II a III	$2,98 \times 10^{-7}$	$2,94 \times 10^{-12}$	$1,11 \times 10^{-8}$	$1,07 \times 10^{-6}$
	I a II	$1,43 \times 10^{-9}$	0	$7,11 \times 10^{-11}$	$6,83 \times 10^{-9}$
Model (12)	I a III	$1,11 \times 10^{-5}$	$2,21 \times 10^{-12}$	$2,29 \times 10^{-7}$	$2,20 \times 10^{-5}$
	II a III	$4,00 \times 10^{-6}$	$2,21 \times 10^{-12}$	$1,48 \times 10^{-7}$	$1,42 \times 10^{-5}$
	I a II	$1,25 \times 10^{-5}$	0	$1,61 \times 10^{-7}$	$1,55 \times 10^{-5}$

Pozrime sme sa ďalej na Tabuľku 7, ktorá uvádza počet iterácií potrebných ku skonvergovaniu algoritmu Metódy I. Vzhľadom k tomu, že autori článku [10] tvrdia, že pri rôznych typoch príkladov skonvergoval nimi navrhnutý algoritmus do 3 iterácií, všímame sme si teda hlavne početnosť prípadov, keď počet iterácií prekročí číslo 3. Dávame opäť do pozornosti aj fakt, že algoritmus Metódy I v prípade s elektrárnami

v zopár prípadoch neskonvergoval.

Dopĺňame ešte porovnanie našich výsledkov počtov iterácií potrebných ku skonvergovaniu algoritmu Metódy I s výsledkami, ktoré autori Metódy I uvádzajú v článku [10]. V porovnaní s našimi výsledkami Metódy I sa líšili výsledky autorov spolu pre tri útvary, avšak z toho dva boli už spomínané neskonvergované útvary. Tretí útvar líšiaci sa vo výslednom počte iterácií podľa autorov článku [10] potreboval na skonvergovanie iba jednu iteráciu, avšak v našom prípade algoritmus Metódy I skonvergoval po dvoch iteráciách.

Tabuľka 7: Informácie o počte iterácií v prípade Metódy I pre environmentálnu verziu hyperbolického modelu s VRS

Typ modelu	Model (11)		Model (12)	
Názov dátového súboru	Elektrárne	Školy	Elektrárne	Školy
Počet iterácií = 1	76	77	69	82
Počet iterácií = 2	12	15	20	9
Počet iterácií = 3	1	3	2	5
Počet iterácií > 3	1	1	0	0
Počet neskonvergovaných útvarov	2	0	1	0

Nasledovná Tabuľka 8 je analogická s predchádzajúcou, a teda udáva počet iterácií potrebných ku skonvergovaniu algoritmu Metódy II. Opäť si všímame najmä prípady, keď algoritmus potreboval viac ako 3 iterácie. V prvom rade vyzdvihujeme nižší počet neskonvergovaných útvarov ako v prípade Metódy I, ale tiež celkovo nižší počet iterácií, ktoré útvary v prípade modelu (12) ku skonvergovaniu potrebovali.

Tabuľka 8: Informácie o počte iterácií v prípade Metódy II pre environmentálnu verziu hyperbolického modelu s VRS

Typ modelu	Model (11)		Model (12)	
Názov dátového súboru	Elektrárne	Školy	Elektrárne	Školy
Počet iterácií = 1	76	77	73	87
Počet iterácií = 2	12	15	19	7
Počet iterácií = 3	2	3	0	2
Počet iterácií > 3	1	1	0	0
Počet neskonvergovaných útvarov	1	0	0	0

V prípade modelu (11) sa počet iterácií pre jednotlivé útvary pri Metódach I a II mimo neskonvergovaných útvarov nelíšil. Čo je však zaujímavé a na čo sme už upozornili, v prípade modelu (12) nastala aj situácia odlišného počtu iterácií pri niektorých útvaroch. Ako sme už predostreli, celkovo môžeme pozorovať pri modeli (12) nižší počet potrebných iterácií pri našej vlastnej Metóde II v porovnaní s Metódou I.

Dávame ešte raz do pozornosti fakt, že Metóda I a aj Metóda II pri týchto dátových súboroch v zopár prípadoch neskonvergovali a taktiež nastala situácia, keď algoritmus síce skonvergoval, avšak po viac ako 3 iteráciách. Podstatnou skutočnosťou je aj fakt, že Metóda II neskonvergovala v prípade modelu (11) len pre jeden útvar v porovnaní s dvomi neskonvergovanými útvarmi Metódy I. Konkrétne sa to týka útvaru *DMU 14*, ktorý robil problém aj Metóde I. V prípade modelu (12) však naša vlastná Metóda II skonvergovala, na rozdiel od Metódy I, pre všetky útvary.

V ďalšom rozoberieme bližšie pozorovania týkajúce sa neskonvergovaných útvarov pri dátovom súbore s Elektrárnami. Najprv sa pozrieme na výsledky modelu (11), pri ktorom Metóda I neskonvergovala pre dva útvary, konkrétne pre *DMU 14* a *DMU 21* a Metóda II iba pre útvar *DMU 14*.

V prípade útvaru *DMU 21* sa pri Metóde I hodnota θ^I síce po 3. iterácii ustálila na číse, ktoré udávajú Metódy II aj III, a teda na hodnote 0,8634, avšak úloha (21) nevyšla nikdy prípustná. Podotýkame ešte, že hodnota β^I z úlohy (13) stále vychádzala po 3. iterácii blízko 1, čiže ostro neklesala, a teda hodnota $\tilde{\theta}^I$ ostro nerástla.

Pre lepší prehľad uvádzame nižšie, v Tabuľke 9, prvých 10 hodnôt výslednej po-

stupnosti θ^I , resp. β^I a z nej vychádzajúce hodnoty $\tilde{\theta}^I$, útvaru *DMU* 21. Keďže sa od seba v ďalších iteráciách už veľmi nelíšia, nepovažujeme za prínosné ich doplniť do tabuľky. Všimnime si, že hodnoty β^I neboli v jednotlivých iteráciách presne rovné 1, ale líšili sa len minimálne.

Tabuľka 9: Niekoľko prvých členov výslednej postupnosti hodnôt θ^I , β^I a $\tilde{\theta}^I$ neskonvergovaného útvaru *DMU* 21 z dátového súboru Elektrárne pre Metódu I - pre model (11)

Typ metódy	Metóda I		
Premenná	θ^I	β^I	$\tilde{\theta}^I$
1	0,860324080969	0,153208722878	0,857950629197
2	0,863405521646	0,974331194025	0,377444706441
3	0,863408163522	0,999977568189	0,367887693466
4	0,863408163518	0,999999964815	0,367879454115
5	0,863408163518	0,999999965142	0,367879453995
6	0,863408163518	0,999999964908	0,367879454081
7	0,863408163518	0,99999996473	0,367879454146
8	0,863408163518	0,999999965138	0,367879453996
9	0,863408163518	0,999999964481	0,367879454238
10	0,863408163518	0,999999964623	0,367879454186

Dôležitým pozorovaním, na ktoré by sme upriamili čitateľovu pozornosť je fakt, že hodnoty β^I do piatej iterácie rastú a následne oscilujú okolo približne rovnakého čísla. Z naznačeného dôkazu konvergenzie Metódy I, ktorú uvádzajú autori tejto metódy vo svojom článku, a o ktorom sme hovorili na konci časti 2.1, vyplýva, že ak postupnosť hodnôt $\tilde{\theta}^I = e^{-\beta^I}$ má byť ostro rastúca, postupnosť hodnôt β^I musí byť ostro klesajúca, čo očividne v tomto prípade nie je splnené.

Ak sa však pozrieme aj na výsledné postupnosti hodnôt β^I niektorých vybraných skonvergovaných útvarov v Tabuľke 10, vidíme, že ani v tomto prípade hodnoty neklesajú, ba naopak rastú. Nahliadnime však aj na postupnosť hodnôt θ^I , ktoré boli vypočítané ako priesečník opornej nadroviny H^I a hyperbolickej krivky podľa vzťahu (15). Dá sa ľahko spozorovať, že tieto hodnoty ostro rastú na rozdiel od postupnosti hodnôt $\tilde{\theta}^I = e^{-\beta^I}$, ktorú pre lepšie porovnanie taktiež doplníme do Tabuľky 10. V Ta-

buľke 10 čitateľ môže tým pádom pozorovať jasnú nerovnosť medzi hodnotami $\tilde{\theta}^I$ a θ^I , o ktorej sme hovorili na konci časti 2.1. Práve rastúce hodnoty θ^I zo vzťahu (15) sú kľúčové pri konvergencii algoritmu Metódy I, a teda vidíme, že napriek nerastúcosti hodnôt $\tilde{\theta}^I$ algoritmus Metódy I skonvergoval. Pre upresnenie dodávame, že miera efektivity určená Metódou III vyšla 0,7916 pre *DMU* 11 a 0,7237 pre *DMU* 67.

Tabuľka 10: Výsledná postupnosť hodnôt θ^I , β^I a $\tilde{\theta}^I$ vybraných skonvergovaných útvarov *DMU* 11 a *DMU* 67 z dátového súboru Elektrárne pre Metódu I - pre model (11)

Útvar	<i>DMU</i> 11			<i>DMU</i> 67		
Premenná	θ^I	β^I	$\tilde{\theta}^I$	θ^I	β^I	$\tilde{\theta}^I$
1	0,7074	0,3523	0,703	0,7010	0,3687	0,6917
2	0,7571	0,3566	0,7001	0,7233	0,8955	0,4084
3	0,7745	0,782	0,4575	0,7237	0,9982	0,3685
4	0,7916	0,8836	0,4133			

Pozrime sa teraz na útvar *DMU* 14, pre ktorý bola situácia odlišná, pretože hodnota θ^I sa neustálila na žiadnom konkrétnom čísle, i keď sa väčšinou pohybovala okolo hodnoty efektivity určenej Metódou III, t.j. 0,88517. Čo sa týka optimálnej hodnoty parametra β^I z úlohy (13), tá vychádzala okolo 0, no občas „vystrelila“ na hodnoty okolo 3,6. V tomto prípade bol opäť problém v neprípustnosti úlohy (21) a hodnota parametra θ^I podobná tej, ktorá vyšla v Metóde III sa dosahovala najskôr v 8. iterácii.

Čo sa týka Metódy II, tá neskonvergovala len pre útvar *DMU* 14. Ani v tomto prípade sa hodnota parametra θ^I neustálila na konkrétnom čísle a hodnota parametra β^I sa pohybovala okolo -1 a taktiež občas „vystrelila“ na hodnoty okolo 2,67, čo bolo dôvodom neskonvergovania.

Pre lepšiu predstavu o situácii neskonvergovaných útvarov uvádzame v Tabuľke 11 niekoľko prvých členov výslednej postupnosti hodnôt θ^I , β^I a $\tilde{\theta}^I = e^{-\beta^I}$ pre neskonvergovaný útvar *DMU* 14 z dátového súboru Elektrárne pri uvažovaní modelu (11). Pre detailnejší obraz výslednej postupnosti hodnôt θ^I , β^I a $\tilde{\theta}^I$ prináležiacej útvaru *DMU* 14, doplníme v Prílohe B Tabuľky 18 a 19 s prvými 50 členmi týchto postupností. Upozorňujeme ešte, že hodnoty v tabuľke sú zaokrúhlené na štyri desatinné miesta, čiže hodnoty β^I sa len približne pohybovali okolo hodnôt 0 v prípade Metódy I, resp.

–1 pri Metóde II.

Tabuľka 11: Niekoľko prvých členov výslednej postupnosti hodnôt θ^I , β^I a $\tilde{\theta}^I$ neskonvergovaného útvaru *DMU* 14 z dátového súboru Elektrárne pre Metódy I a II - pre model (11)

Typ metódy	Metóda I			Metóda II		
Premenná	θ^I	β^I	$\tilde{\theta}^I$	θ^I	β^I	$\tilde{\theta}^I$
1	0,633	0,4493	0,6381	0,633	0,4493	0,6381
2	0,6968	0	1	0,6893	-1	2,7183
3	0,7628	0	1	0,7585	-1	2,7183
4	0,8442	0	1	0,8247	-1	2,7183
5	0,8809	0	1	0,8767	-1	2,7183
6	0,8844	0	1	0,8848	-1	2,7183
7	0,885	0	1	0,8852	-1	2,7183
8	0,8852	0	1	0,8271	2,6718	0,0691
9	0,8773	3,6716	0,0254	0,8727	-1	2,7183
10	0,8841	0	1	0,8841	-1	2,7183
11	0,8851	0	1	0,8851	-1	2,7183
12	0,8848	3,6712	0,0254	0,885	2,671	0,0692
13	0,8852	0	1	0,8852	-1	2,7183

Ďalej bližšie nazrieme na problémy neskonvergovaných útvarov v prípade modelu (12). Konkrétne mala s konvergenciou problém len Metóda I opäť v prípade dátového súboru s Elektrárnami pri jednom útware *DMU* 26. Situácia je podobná ako pri útware *DMU* 21 v prípade modelu (11), a teda hodnota θ^I sa síce ustálila na hodnotu, ktorú udávajú algoritmy Metód II aj III, ale úloha (21) nebola nikdy prípustná. Ešte dodávame, že hodnota β^I z úlohy (13) sa opäť po 3. iterácii pohybovala okolo 1.

Ďalšou otázkou, ktorou sa zaoberáme, je problém identifikovania nuly, ktorý v prípade modelu (12) znamenal podstatný rozdiel v počte neskonvergovaných útvarov.

Pri inicializačnej hodnote presnosti 10^{-6} v dátovom súbore s elektrárnami neskonvergovali v prípade Metódy I aj Metódy II dva útvary. V oboch prípadoch však šlo o rôzne útvary. Pri presnosti 10^{-5} už v oboch metódach neskonvergoval len jeden z pôvodných

útvarov a pri nastavení presnosti 10^{-4} skonvergovali v prípade našej Metódy II všetky útvary. Uvedené výsledky sú teda pri presnosti 10^{-4} . V prípade Metódy I stále neskonvergoval jeden útvar, avšak ďalšie znižovanie presnosti by sa minulo účinku, pretože problém bol v úlohe prípustnosti, ktorá nikdy nevyšla a zmena presnosti by výsledok neovplyvnila.

V dátovom súbore so školami pri Metóde I s východiskovou presnosťou 10^{-6} neskonvergovalo 12 útvarov, pričom v prípade našej Metódy II išlo len o 4 útvary. Pri presnosti 10^{-5} už boli výsledky pozitívnejšie a pri oboch metódach neskonvergoval len jeden, avšak odlišný, útvar. Pri nastavení presnosti na hodnotu 10^{-4} už skonvergovali všetky útvary.

V prípade modelu (11) zmeny presnosti z 10^{-6} na 10^{-5} , resp. 10^{-4} , nespôsobili odlišnosti vo výsledkoch v ani jednom dátovom súbore.

Celkovo teda hodnotíme, že algoritmus Metódy III je nielen časovo efektívnejší, ale aj spoľahlivejší z hľadiska skonvergovania.

3.3 Veľké dátové súbory

V nasledujúcej kapitole uvádzame výsledky pozorovaní jednotlivých metód pri ich použití na veľké dátové súbory. Jedná sa o simulované dáta bez nežiadúcich výstupov zo zdroja [4], ktoré použili na testovanie Metódy I aj autori tejto metódy v článku [10]. V tejto časti sa teda zameriavame na porovnanie výsledkov jednotlivých metód riešenia základného hyperbolického modelu daného vzťahmi (3)-(4).

Tabuľka 12: Súhrnné informácie o dátových súboroch prislúchajúcich k základnej verzii hyperbolického modelu s VRS pri veľkých dátových súboroch

Názov dátového súboru	Data 1	Data 2	Data 3	Data 4
Počet DMU	2 500	5 000	7 500	10 000
Počet vstupov	2	2	2	2
Počet výstupov	3	3	3	3
Počet efektívnych útvarov	25	50	75	100

Nasledovná Tabuľka 13 poskytuje prehľad rýchlostí algoritmov jednotlivých metód

v prepočte na jeden útvar. Opäť podotýkame, že pri každom dátovom súbore aspoň jedna z Metód I alebo II neskonvergovala pre niekoľko útvarov, a z toho dôvodu uvádzame prepočet na 1 skonvergovaný útvar. Na rozdiel od predošlých tabuliek tohto typu neuvádzame počet odlišných výsledkov, keďže z hľadiska obsírnosti tejto problematiky to nepovažujeme za prehľadný spôsob podania výsledkov čitateľovi. Odlišnosti výsledkov hodnôt efektívít teda rozoberáme neskôr v osobitných tabuľkách.

V tomto prípade treba opäť vyzdvihnúť významnú časovú úsporu Metódy III. Pozorujeme, že pri všetkých dátových súboroch vychádzala Metóda III viac ako 3-krát efektívnejšia v porovnaní so zvyšnými dvomi metódami. Naopak Metóda II bola síce o niečo efektívnejšia v porovnaní s Metódou I, ale vzhľadom k tomu, že počet efektívnych útvarov tvoril len 1 percento z ich celkového počtu, očakávali sme výraznejší rozdiel medzi Metódou I a II.

Čo sa týka použitej presnosti identifikovania nuly, použili sme len 10^{-4} , pretože pri počiatočnom nastavení presnosti 10^{-6} neskonvergovalo v prípade Metódy I pre dátový súbor Data 1 rádovo niekoľko stoviek útvarov. Preto sme vzhľadom k časovej náročnosti výpočtov pracovali len s presnosťou 10^{-4} .

Tabuľka 13: Základné informácie o výsledkoch prislúchajúcich k základnej verzii hyperbolického modelu s VRS pri veľkých dátových súboroch

Názov dátového súboru	Data 1	Data 2	Data 3	Data 4
Čas na 1 DMU - Metóda I	1,3797	1,6644	1,9003	2,2514
Čas na 1 DMU - Metóda II	1,3583	1,6635	1,8828	2,2486
Čas na 1 DMU - Metóda III	0,4349	0,5224	0,5803	0,6503

V nasledovnej Tabuľke 14 uvádzame informácie o počte iterácií potrebných ku skonvergovaniu jednotlivých algoritmov ako aj počet neskonvergovaných útvarov pre jednotlivé dátové súbory. Z dôvodu prehľadnejšej demonštrácie rozdielov jednotlivých algoritmov má Tabuľka 14 mierne odlišnú štruktúru ako Tabuľka 8 týkajúca sa počtov iterácií v prípade menej rozmerných dátových súboroch. Tabuľka je horizontálne rozdelená na 4 hlavné časti prislúchajúce jednotlivým dátovým súborom, pričom jednotlivé riadky každej z týchto častí prináležia výsledkom jednotlivých metód.

Upozorňujeme však, že Metódu III v Tabuľke 14 neuvádzame, keďže jej základná

myšlienka nie je založená na iteratívnom algoritme, ale na vyriešení jednej úlohy SDP, ktorá ihneď dospeje k optimálnemu riešeniu. Pri Metóde III teda nepozorujeme žiadne problémy s konvergenciou algoritmu, či iné komplikácie s riešiteľnosťou úlohy. Zároveň ku každému dátovému súboru pridávame, v riadku označenom ako „článok [10]“, informáciu o výslednom počte iterácií Metódy I, ktoré vyšli autorom tejto metódy v článku [10] použitím softvéru GAMS.

Ako prvé si určite všimnime, že autorom článku [10] skonvergovali všetky sledované útvary pre všetky dátové súbory, čo sa však v našom prípade tvrdiť nedá. Pripomíname ešte veľmi dôležitý fakt a to, že v prípade všetkých dátových súborov riešime základný hyperbolický model bez nežiadúcich výstupov, ktorého metódy riešenia I a II pri malých dátových súboroch z časti 3.1 vždy skonvergovali, dokonca hneď po prvej iterácii. Odlišnosti v porovnaní s výsledkami autorov Metódy I neplynú len z počtu neskonvergovaných útvarov, ale aj z rozdielnej distribúcie zvyšných, skonvergovaných útvarov. Rozdiely sú však rádovo maximálne v desiatkach útvarov.

Celkovo pozorujeme, že pri porovnaní našich výsledkov Metódy I a II vychádza v menších dátových súboroch Data 1 a Data 2 efektívnejší náš algoritmus Metódy II. Tieto závery sú dôsledkom nulového počtu neskonvergovaných útvarov, ale aj celkovo nižšieho počtu iterácií potrebných ku skonvergovaniu algoritmu Metódy II. Hoci musíme podotknúť, že Metóda I pri týchto menších dátových súboroch Data 1 a Data 2 neskonvergovala len v jednom prípade pri dátovom súbore Data 2. V prípade rozsiahlejších dátových súborov však už dochádza k inej situácii. Celkovo vidíme, že počet iterácií potrebných ku skonvergovaniu algoritmu Metódy II je síce pri skonvergovaných útvaroch nižší, ale počet neskonvergovaných útvar je naopak vyšší a to najmä v prípade dátového súboru Data 4.

Tabuľka 14: Informácie o počte iterácií pre jednotlivé rozsiahle dátové súbory - základná verzia hyperbolického modelu s VRS

Počet iterácií		1 iterácia	2 iterácie	3 iterácie	> 3 iterácie	Neskonver.
Data 1	Metóda I	1383	1076	41	0	0
	Metóda II	1388	1083	29	0	0
	článok [10]	1385	1080	35	0	0
Data 2	Metóda I	2341	2566	92	0	1
	Metóda II	2402	2546	50	2	0
	článok [10]	2369	2562	69	0	0
Data 3	Metóda I	2835	4477	183	4	1
	Metóda II	2875	4499	114	6	6
	článok [10]	2856	4501	143	0	0
Data 4	Metóda I	2341	7230	320	35	4
	Metóda II	2442	7339	146	30	43
	článok [10]	2344	7363	293	0	0

V ďalšej Tabuľke 15 uvádzame prehľad základných štatistík absolútnych rozdielov výsledných mier efektívít skonvergovaných útvarov medzi jednotlivými metódami. Všimnime si, že v tomto prípade vyšli rozdiely o niečo väčšie ako v prípade menších dátových súboroch s nežiadúcimi výstupmi z Tabuľky 6. Napriek tomu však nepozorujeme markantné rozdiely v hodnotách efektívít vypočítaných jednotlivými metódami pre skonvergované útvary, o čom hovorí napríklad priemer absolútnych rozdielov medzi jednotlivými metódami v stĺpci „Avg“. Ešte doplníme, že vo väčšine prípadov najmenšie absolútne rozdiely vo výsledných mierach efektívít pozorujeme medzi Metódami I a II.

Tabuľka 15: Analýza rozdielov výsledkov pre jednotlivé rozsiahle dátové súbory - uvažujúc VRS

Porovnávané metódy		Max	Min	Avg	Sum
Data 1	I a III	$1,36 \times 10^{-6}$	$3,3 \times 10^{-10}$	$9,49 \times 10^{-9}$	$2,37 \times 10^{-5}$
	II a III	$3,54 \times 10^{-5}$	$3,08 \times 10^{-10}$	$9,01 \times 10^{-8}$	$2,25 \times 10^{-4}$
	I a II	$3,54 \times 10^{-5}$	0	$8,08 \times 10^{-8}$	$2,02 \times 10^{-4}$
Data 2	I a III	$3,94 \times 10^{-3}$	$2,78 \times 10^{-10}$	$1,58 \times 10^{-6}$	$7,89 \times 10^{-3}$
	II a III	$3,94 \times 10^{-3}$	$2,78 \times 10^{-10}$	$1,99 \times 10^{-6}$	$9,95 \times 10^{-3}$
	I a II	$6,26 \times 10^{-5}$	0	$4,12 \times 10^{-7}$	$2,06 \times 10^{-3}$
Data 3	I a III	$6,10 \times 10^{-3}$	$3,03 \times 10^{-10}$	$4,37 \times 10^{-6}$	$3,28 \times 10^{-2}$
	II a III	$6,10 \times 10^{-3}$	$2,88 \times 10^{-10}$	$4,65 \times 10^{-6}$	$3,49 \times 10^{-2}$
	I a II	$5,57 \times 10^{-5}$	0	$2,73 \times 10^{-7}$	$2,05 \times 10^{-3}$
Data 4	I a III	$7,60 \times 10^{-3}$	$2,78 \times 10^{-10}$	$9,12 \times 10^{-6}$	$9,11 \times 10^{-2}$
	II a III	$7,60 \times 10^{-3}$	$2,74 \times 10^{-10}$	$9,65 \times 10^{-6}$	$9,61 \times 10^{-2}$
	I a II	$6,40 \times 10^{-5}$	0	$4,91 \times 10^{-7}$	$4,89 \times 10^{-3}$

Pre lepšiu predstavu ohľadom nájdených rozdielov uvádzame aj Tabuľku 16 poskytujúcu informácie o počte skonvergovaných útvarov, ktoré mali absolútny rozdiel výsledných hodnôt efektív vypočítaných jednotlivými metódami väčší ako 10^{-5} , 10^{-4} a 10^{-3} . Môžeme ľahko nahliadnuť, že sa v skutočnosti nejedná o veľký počet útvarov relatívne k ich celkovému počtu v pozorovaných dátových súboroch. Upriamime pozornosť ešte aj na fakt, že v týchto dátových súboroch sa absolútny rozdiel hodnôt efektív väčší ako 10^{-2} vôbec nevyskytoval, a preto už tieto hodnoty v tabuľke neuvádzame. Upozorňujeme ešte na pozorovanie, že v prípade Metód I a II sa nevyskytovali dokonca ani také útvary, ktorých absolútny rozdiel výsledných efektív z jednotlivých metód by prevyšoval 10^{-4} .

Tabuľka 16: Analýza počtu útvarov s rozdielnymi výsledkami pre skonvergované útvary a jednotlivé rozsiahle dátové súbory - uvažujúc VRS

Pozorovaný rozdiel väčší ako		10^{-5}	10^{-4}	10^{-3}
Data 1	I a III	0	0	0
	II a III	8	0	0
	I a II	8	0	0
Data 2	I a III	9	7	3
	II a III	73	7	3
	I a II	68	0	0
Data 3	I a III	13	13	12
	II a III	84	13	12
	I a II	72	0	0
Data 4	I a III	46	41	27
	II a III	211	41	27
	I a II	167	0	0

Ak by sme sa dívali na počet všetkých útvarov s rozdielnymi výsledkami, t.j. do pozorovania zaradíme aj neskonvergované útvary, dostávame Tabuľku 17. Vzhľadom k tomu, že niektoré útvary pre nejakú z metód neskonvergovali, počty rozdielných výsledkov pre daný dátový súbor budú, pochopiteľne, väčšie, a preto sme do tabuľky zaradili aj počet útvarov, ktorých rozdiel výsledných mier efektívít je väčší ako 10^{-2} . Môžeme si všimnúť, že spravidla rozdiel týchto počtov s počtami z Tabuľky 16 korešponduje s počtom neskonvergovaných útvarov. V prípade neskonvergovaných útvarov pri Metóde I a II sa tieto útvary väčšinou nezhodovali, čo vidieť aj na počte útvarov s rozdielom výsledkov Metód I a II väčších ako 10^{-2} .

Tabuľka 17: Analýza počtu útvarov s rozdielnymi výsledkami pre všetky útvary a jednotlivé rozsiahle dátové súbory - uvažujúc VRS

Pozorovaný rozdiel väčší ako		10^{-5}	10^{-4}	10^{-3}	10^{-2}
Data 1	I a III	0	0	0	0
	II a III	8	0	0	0
	I a II	8	0	0	0
Data 2	I a III	10	8	4	1
	II a III	73	7	3	0
	I a II	69	1	1	1
Data 3	I a III	14	14	13	1
	II a III	90	19	18	6
	I a II	77	5	5	5
Data 4	I a III	50	45	31	4
	II a III	254	84	70	43
	I a II	214	47	47	47

Pri celkovom pohľade na výsledky uvedených metód môžeme jednoznačne tvrdiť, že Metóda III je nielen približne 3-krát rýchlejšia a efektívnejšia, ale aj spoľahlivejšia v samotnej konvergencii algoritmu, keďže pri všetkých pozorovaných dátových súboroch sme na rozdiel od zvyšných dvoch metód dokázali zistiť hodnotu efektivity pre všetky sledované útvary. Navyše máme vďaka dôkazu Tvrdenie 2.1 istotu, že optimálna hodnota efektivity θ^* je rovná hodnote miery efektivity hyperbolického modelu daného vzťahmi (3)-(4).

Záver

Postupné rozširovanie znalostí z oblasti optimalizácie a nelineárneho programovania so sebou prináša aj nové metódy riešenia problémov, na ktoré sme dosiaľ nepoznali odpoveď. Prirodzene sa tak natískajú otázky týkajúce sa ich predností alebo nedostatkov oproti ostatným. V našej práci sme sa konkrétne zamerali na porovnanie troch vybraných metód riešenia jedného z nelineárnych modelov, konkrétne hyperbolického modelu.

Najprv však bolo nutné predstaviť základy DEA, potrebné k ďalšej analýze vybraných algoritmov riešenia. V kapitole 1.1 preto objasňujeme kľúčové pojmy a zákonitosti z teórie obálkovej analýzy dát, ktoré sme čerpali napríklad z [13] a [2]. V podkapitole 1.2 sme usmernili našu pozornosť na vybraný, nelineárny DEA model, ktorý je stredobodom záujmu našej práce, a teda na hyperbolický model. V časti 1.3 sme čitateľa oboznámili s DDF modelom, ktorý je kľúčový v dvoch porovnávaných metódach, konkrétne v Metóde I a II. Ako sme v práci spomínali, hyperbolický model má značné využitie aj v environmentálnej problematike, a preto sme venovali časť 1.4 environmentálnej modifikácii hyperbolického a DDF modelu. Keďže nás primárne zaujímali predovšetkým možnosti riešenia prvého uvedeného, na začiatku kapitoly 2 sme predstavili efektívny spôsob riešenia v prípade CRS, ktorý bol popísaný v článku [9]. V nasledujúcich častiach 2.1, 2.2 a 2.3 sme sa zamerali na metódy riešenia hyperbolického modelu pre prípad VRS, presnejšie na dva vybrané algoritmy predstavené v článkoch [10] a [16] a na našu vlastnú modifikáciu metódy z článku [10], ktorú sme uviedli s cieľom zrýchliť a zefektívniť algoritmus. Kapitulu sme tiež doplnili o mierne modifikované metódy riešenia v prípade nežiadúcich výstupov nachádzajúce sa v častiach 2.4.1, 2.4.2 a 2.4.3.

Kľúčovou je z hľadiska naplnenia cieľa našej práce kapitola 3, v ktorej sme empiricky porovnali predstavené metódy riešenia. Na začiatku podkapitol 3.1, 3.2 a 3.3 sme zhrnuli a objasnili základné informácie o dátových súboroch, na ktorých sme vykonávali analýzu a v ďalšom sme uviedli konkrétne výsledky našich pozorovaní pre rôzne veľké dátové súbory. Za najvýznamnejší prínos našej práce považujeme výsledky týkajúce sa rýchlosti uvedených metód vzhľadom ku skutočnosti, že Metóda III vychádzala približne trikrát rýchlejšia ako Metóda I či II. Dôležitým pozorovaním taktiež bolo,

že v príklade o elektrárnach, ale aj v príkladoch s väčším množstvom útvarov sa vyskytli objekty, pre ktoré algoritmy Metódy I a II neskonvergovali, čo robí z Metódy III spoľahlivejší spôsob nájdenia optimálneho riešenia hyperbolického modelu.

Možným rozšírením tejto práce by mohlo byť nájdenie konkrétneho dôvodu, resp. pravidla, prečo v niektorých prípadoch algoritmus Metódy I a II neskonvergoval aj v prípade ich základných verzií, prípadne bližšie rozanalyzovanie výsledkov neskonvergovaných útvarov.

Zoznam použitej literatúry

- [1] Aparicio, J., Ruiz, J.L., Sirvent I.: *Closest targets and minimum distance to the Pareto-efficient frontier in DEA*, Journal of Productivity Analysis 28 (2007), 209-218
- [2] Cooper, W.W., Seiford, L. M., Tone, K.: *Data Envelopment Analysis. A Comprehensive Text with Models, Applications, References and DEA-Solver Software*, Second Edition, Springer, New York, 2007
- [3] Chambers, R.G., Chung Y., Färe, R.: *Profit, Directional Distance Function, and Nerlovian Efficiency*, Journal of Optimization Theory nad Applications 98 (1998), 351-364
- [4] Dulá, J.H.: *José Dulá's Home page*, dostupné na internete:
<http://www.people.vcu.edu/~jdula/>
- [5] Emrouznejad, A.: *Data Envelopment Analysis Homepage*, dostupné na internete (1.1.2018):
<http://deazone.com>
- [6] Farrell, M.J.: *The Measurement of Productive Efficiency*, Journal of the Royal Statistical Society 120 (1957), 253-290
- [7] Färe, R., Grosskopf, S., Lovell, C.A.K.: *The Measurement of Efficiency of Production*, Springer Science+Business Media, New York, 1985
- [8] Färe, R., Grosskopf, S., Pasurka, C.A.: *Environmental production functions and environmental directional distance functions*, Energy 32 (2007), 1055-1066
- [9] Färe, R., Grosskopf, S., Zaim, O.: *Hyperbolic efficiency and return to dollar*, European Journal of Operational Research 136 (2002), 671-679
- [10] Färe, R. et al.: *Estimating the hyperbolic distance function: A directional distance function approach*, European Journal of Operational Research 254 (2016), 312-319

- [11] Färe, R. et al.: *Multilateral Productivity Comparisons When Some Outputs are Undesirable: A Nonparametric Approach*, The Review of Economics and Statistics 71 (1989), 90-98
- [12] Ganley, J.A., Cubbin, J. S.: *Public Sector Efficiency Measurement: Applications of Data Envelopment Analysis*, Elsevier, New York, 1992
- [13] Halická, M.: *DEA modely*, učebné texty, FMFI UK, Bratislava, 2015, dostupné na internete (28.12.2017):
<http://www.iam.fmph.uniba.sk/institute/halicka/teach/DEAskripta.pdf>
- [14] Halická M., Trnovská M.: *Duality and Profit Efficiency for the Hyperbolic Measure Model*, working paper, FMFI UK, Bratislava, 2017
- [15] Halická M., Trnovská M.: *Negative features of hyperbolic and directional distance models for technologies with undesirable outputs*, working paper, FMFI UK, Bratislava, 2017
- [16] Halická M., Trnovská M.: *Russell Measure Model: computational aspects, duality and profit efficiency*, European Journal of Operational Research (2018), doi: 10.1016/j.ejor.2018.01.012
- [17] Mehdiloozad, M., Sahoo, B.K., Roshdi, I.: *Generalized Multiplicative Directional Distance Function for Efficiency Measurement in DEA*, European Journal of Operational Research 232 (2014), 679-688
- [18] Mehdiloozad, M. et al.: *On the identification of the global reference set in data envelopment analysis*, European Journal of Operational Research 245 (2015), 779-788
- [19] Ševčovič, D., Halická, M., Brunovský, P.: *DEA analysis for a large structured bank branch network*, Central European Journal of Operational Research 9 (2001), 329-342, dostupné na internete (3.1.2018):
<http://www.iam.fmph.uniba.sk/institute/sevcovic/papers/cl19.pdf>
- [20] Trnovská, M.: *Konverzná optimalizácia*, prednášky, FMFI UK, Bratislava, 2017

- [21] Zofio, J.L., Lovell C.A.K.: *Graph efficiency and productivity measures: an application to US agriculture*, Applied Economics 33 (2001), 1433-1442
- [22] Zofio, J.L., Prieto A.M.: *Environmental efficiency and regulatory standards: The case of CO₂ emission from OECD industries*, Resource and Energy Economics 23 (2001), 63-83

Príloha A

A Zdrojové kódy programov pre MATLAB

A.1 Metóda I

Základná Metóda I

```
1 function [Theta,t]=Hyperbolic()
2
3 %Zadefinovanie matic vstupov a vystupov, pricom 1 riadok zodpoveda 1 DMU
4
5 %Data
6 % %Mehdiloo2015
7 filename = 'Mehdiloo2015.xlsx';
8 sheet = 1;
9 xRange = 'I7:M76';
10 X=xlsread(filename,sheet,xRange);
11
12 sheet = 2;
13 yRange = 'G5:I74';
14 Y=xlsread(filename,sheet,yRange);
15
16 n=length(Y(:,1)); %pocet DMU
17 m=length(X(1,:)); %pocet vstupov
18 s=length(Y(1,:)); %pocet vystupov
19
20 %Definicia premennych
21 Theta = zeros(1,n);
22 NumIter = zeros(1,n);
23
24 tic
25 for o = 1:n
26     AddIter = 0; %pomocna premenna pre zacatie dalsej iteracie; 0 - unsolved, 1 - so
27     xo = X(o,:)';
28     yo = Y(o,:)';
29
30     gx = xo;
```

```

31     gy = yo;
32
33     %%% 1. faza - cieľ: najst priesečník
34
35     % 1. krok - najdenie riešenia primárnej úlohy
36     cvx_begin quiet
37         cvx_precision default
38         variable lambda(n) nonnegative;
39         variable B;
40         maximize (B);
41         subject to
42             X'*lambda <= xo - B*gx;
43             -Y'*lambda <= -yo - B*gy;
44             sum(lambda) == 1;
45     cvx_end
46
47     % ak je beta = 0, útvar je efektívny, koncime algoritmus
48     if abs(cvx_optval) <= 10^(-6)
49         Theta(o) = 1;
50         NumIter(o) = NumIter(o) + 1;
51     else
52
53     % 1.krok - najdenie riešenia dualnej úlohy
54     cvx_begin quiet
55         cvx_precision default
56         variables p(s) w(m) z;
57         maximize (yo' * p - xo' * w - z)
58         subject to
59             Y * p - X * w - ones(n,1)*z <= zeros(n,1);
60             gy' * p + gx' * w == 1;
61             p >= 0;
62             w >= 0;
63     cvx_end
64
65     while NumIter(o) == 0 || AddIter == 0
66
67     % 2. krok - vypočítat theta a najst priesečník
68     if w==0

```

```
69     theta = p' * yo / z;
70 elseif p==0
71     theta = -z / (w' * xo);
72 else
73     theta = (-z + sqrt(z^2 + 4 * p'*yo * w' * xo)) / (2 * w' * xo);
74 end
75
76 % 3. krok - zistit, ci priesečník leži v množine M
77
78 % množina  $R^I$ 
79 Bndr = Y * p - X * w - ones(n,1)*z;
80 j = find( abs(Bndr) < 10^(-6) );
81
82 % Úloha prípustnosti
83 cvx_begin quiet
84     cvx_precision low
85     variable l(size(j,1));
86     subject to
87         X(j,:)'*l <= theta*xo;
88         Y(j,:)'*l >= yo/theta;
89         sum(l) == 1;
90         l >= 0;
91 cvx_end
92
93 if cvx_optval <= 10^(-6) % je prípustný
94     Theta(o) = theta;
95     AddIter = 1;
96 elseif cvx_optval == Inf % nie je prípustný
97     % Skonstruovanie nového smeru
98     gx = xo - theta * xo;
99     gy = yo/theta - yo;
100
101     % Vyriešenie primárnej úlohy
102     cvx_begin quiet
103         cvx_precision default
104         variable lambda(n) nonnegative;
105         variable B;
106         maximize (B);
```

```

107         subject to
108             X'*lambda <= xo - B*gx;
109             -Y'*lambda <= -yo - B*gy;
110             sum(lambda) == 1;
111         cvx_end
112
113         % Vyriesenie dualnej ulohy
114         cvx_begin quiet
115             cvx_precision default
116             variables p(s) w(m) z;
117             maximize (yo' * p - xo' * w - z)
118             subject to
119                 Y * p - X * w - ones(n,1)*z <= zeros(n,1);
120                 gy' * p + gx' * w == 1;
121                 p >= 0;
122                 w >= 0;
123             cvx_end
124
125         end
126
127         NumIter(o) = NumIter(o) + 1;
128
129     end
130 end
131
132 end;
133 t=toc;
134
135 %vytvorenie excel suboru
136 filename = 'Vysledky.xlsx';
137 xlswrite(filename,Theta',1);
138 xlswrite(filename,NumIter',2);
139 xlswrite(filename,t,3);
140
141 end

```

Metóda I v prípade nežiadúcich výstupov

```
1 function [Theta,tEnviro]=HyperbolicEnviro()
2
3 %Zadefinovanie matic vstupov a vystupov, pricom 1 riadok zodpoveda 1 DMU
4
5 %Data
6 %elektrarne
7 filename= 'elektrarne_data.xlsx';
8 sheet = 2;
9 xRange = 'B4:D95';
10 yRange = 'E4:E95';
11 yBadRange = 'F4:G95';
12
13 X=xlsread(filename,sheet,xRange);
14 Y=xlsread(filename,sheet,yRange);
15 Ybad=xlsread(filename,sheet,yBadRange);
16
17 n=length(Y(:,1)); %pocet DMU
18 m=length(X(1,:)); %pocet vstupov
19 s=length(Y(1,:)); %pocet ziaducich vystupov
20 sBad=length(Ybad(1,:)); %pocet neziaducich vystupov
21
22 %Definicia premennych
23 Theta = zeros(1,n);
24 NumIter = zeros(1,n);
25
26 tic
27
28 for o = 1:n
29
30     AddIter = 0; %pomocna premenna pre zacatie dalsej iteracie; 0 - unsolved, 1 - so
31     xo = X(o,:)';
32     yo = Y(o,:)';
33     yoBad = Ybad(o,:)';
34
35     gy = yo;
36     gyBad = yoBad;
```



```
37
38 %%% 1. faza - cieľ: najst priesečník
39
40 % 1. krok - najdenie riešenia primárnej úlohy
41 cvx_begin quiet
42     cvx_precision default
43     variable lambda(n) nonnegative;
44     variable B;
45     maximize (B);
46     subject to
47         X'*lambda <= xo;
48         -Y'*lambda <= -yo - B*gy;
49         %Ybad'*lambda <= yoBad - B*gyBad;
50         Ybad'*lambda == yoBad - B*gyBad;
51         sum(lambda) == 1;
52 cvx_end
53
54 % ak je beta = 0, útvar je efektívny, koncime algoritmus
55 if abs(cvx_optval) <= 10^(-6)
56     Theta(o) = 1;
57     NumIter(o) = NumIter(o) + 1;
58 else
59
60 % 1.krok - najdenie riešenia dualnej úlohy
61 cvx_begin quiet
62     cvx_precision default
63     variables p(s) pBad(sBad) w(m) z;
64     maximize (yo' * p - xo' * w - yoBad' * pBad - z)
65     subject to
66         Y * p - X * w - Ybad * pBad - ones(n,1)*z <= zeros(n,1);
67         gy' * p + gyBad' * pBad == 1;
68         p >= 0;
69         %pBad >= 0;
70         w >= 0;
71 cvx_end
72
73 while NumIter(o) == 0 || AddIter == 0
74
```

```

75 % 2. krok - vypocitat theta a najst priesečník
76 if pBad==0
77     theta = p' * yo / (w' * xo + z);
78 elseif p==0
79     theta = -(z + w' * xo) / (pBad' * yoBad);
80 else
81     theta = (- z - w'*xo + sqrt((z+w'*xo)^2 + 4 *p'*yo * pBad' * yoBad)) / (2 * pBad' * yoBad);
82 end
83
84 % 3. krok - zistit, ci priesečník lezi v mnozine M
85
86 % mnozina R^I
87 Bndr = Y * p - X * w - Ybad * pBad - ones(n,1)*z;
88 j = find( abs(Bndr) < 10^(-6) );
89
90 % Uloha pripustnosti
91 cvx_begin quiet
92     cvx_precision low
93     variable l(size(j,1));
94     %minimize 0
95     subject to
96         X(j,:)'*l <= xo;
97         Y(j,:)'*l >= yo/theta;
98         %Ybad(j,:)'*l <= theta*yoBad;
99         Ybad(j,:)'*l == theta*yoBad;
100     sum(l) == 1;
101     l >= 0;
102 cvx_end
103
104 if cvx_optval <= 10^(-6) % je pripustny
105     Theta(o) = theta;
106     AddIter = 1;
107 elseif cvx_optval == Inf % nie je pripustny
108     % construct new direction
109     gyBad = yoBad - theta * yoBad;
110     gy = yo/theta - yo;
111
112 % Vyriesenie primarnej ulohy

```

```

113     cvx_begin quiet
114         cvx_precision default
115         variable lambda(n) nonnegative;
116         variable B;
117         maximize (B);
118         subject to
119             X'*lambda <= xo;
120             -Y'*lambda <= -yo - B*gy;
121             %Ybad'*lambda <= yoBad - B*gyBad;
122             Ybad'*lambda == yoBad - B*gyBad;
123             sum(lambda) == 1;
124     cvx_end
125
126     % Vyhľadanie dualnej úlohy
127     cvx_begin quiet
128         cvx_precision default
129         variables p(s) pBad(sBad) w(m) z;
130         maximize (yo' * p - xo' * w - yoBad' * pBad - z)
131         subject to
132             Y * p - X * w - Ybad * pBad - ones(n,1)*z <= zeros(n,1);
133             gy' * p + gyBad' * pBad == 1;
134             p >= 0;
135             %pBad >= 0;
136             w >= 0;
137     cvx_end
138
139     end
140
141     NumIter(o) = NumIter(o) + 1;
142
143     end
144 end
145
146 end
147 tEnviro = toc;
148
149 %vytvorenie excel suboru
150 filename = 'VysledkyEnviro.xlsx';

```

```
151 xlswrite(filename,Theta',1);
152 xlswrite(filename,NumIter',2);
153 xlswrite(filename,tEnviro,3);
154 end
```

A.2 Metóda II

Základná Metóda II

```
1  %Zadefinovanie matic vstupov a vystupov, pricom 1 riadok zodpoveda 1 DMU
2
3  %Data
4  %Mehdiloo2015
5  filename = 'Mehdiloo2015.xlsx';
6  sheet = 1;
7  xRange = 'I7:M76';
8  X=xlsread(filename,sheet,xRange);
9
10 sheet = 2;
11 yRange = 'G5:I74';
12 Y=xlsread(filename,sheet,yRange);
13
14 n=length(Y(:,1));    %pocet DMU
15 m=length(X(1,:));    %pocet vstupov
16 s=length(Y(1,:));    %pocet vystupov
17
18 %Definicia premennych
19 Theta = zeros(1,n);
20 NumIter = zeros(1,n);
21
22 tic
23 for o = 1:n
24
25     AddIter = 0;    %pomocna premenna pre zacatie dalsej iteracie; 0 - unsolved, 1 - so
26     xo = X(o,:)';
27     yo = Y(o,:)';
28     gx = xo;
29     gy = yo;
```

```
30
31 %%% 1. faza - cieľ: najst priesečník
32
33 % 1. krok - najdenie riešenia primárnej úlohy
34 cvx_begin quiet
35     variable lambda(n) nonnegative;
36     variable B;
37     maximize (B);
38     subject to
39         X'*lambda <= x0 - B*gx;
40         -Y'*lambda <= -y0 - B*gy;
41         sum(lambda) == 1;
42     cvx_end
43
44 % ak je beta = 0, útvar je efektívny, koncime algoritmus
45 if abs(cvx_optval) <= 10^(-6)
46     Theta(o) = 1;
47     NumIter(o) = NumIter(o) + 1;
48
49 else
50
51 % 1.krok - najdenie riešenia dualnej úlohy
52 cvx_begin quiet
53     variables p(s) w(m) z;
54     maximize (y0' * p - x0' * w - z)
55     subject to
56         Y * p - X * w - ones(n,1)*z <= zeros(n,1);
57         gy' * p + gx' * w == 1;
58         p >= 0;
59         w >= 0;
60     cvx_end
61
62 %%% 2. faza - cieľ: najst projekciu priesečníka späť na hranicu množiny M
63
64 while NumIter(o) == 0 || AddIter == 0
65
66 % 2. krok - vypočítat theta a najst priesečník
67 if w==0
```

```

68     theta = p' * yo / z;
69 elseif p==0
70     theta = -z / (w' * xo);
71 else
72     theta = (-z + sqrt(z^2 + 4 * p'*yo * w' * xo)) / (2 * w' * xo);
73 end
74
75 % Skonstruovanie noveho smeru
76 gx =    xo - theta * xo;
77 gy =    yo/theta - yo;
78
79 % Vyriesenie primarnej ulohy
80 cvx_begin quiet
81     variable lambda(n) nonnegative;
82     variable B;
83     maximize (B);
84     subject to
85         X'*lambda <= theta * xo - B*gx;
86         -Y'*lambda <= -1/theta * yo - B*gy;
87         sum(lambda) == 1;
88 cvx_end
89
90 if abs(cvx_optval) <= 10^(-6)
91     Theta(o) = theta;
92     AddIter = 1;
93 else
94     % Vyriesenie dualnej ulohy
95     cvx_begin quiet
96         variables p(s) w(m) z;
97         maximize (1/theta * yo' * p - theta * xo' * w - z)
98         subject to
99             Y * p - X * w - ones(n,1)*z <= zeros(n,1);
100             gy' * p + gx' * w == 1;
101             p >= 0;
102             w >= 0;
103     cvx_end
104
105 end

```

```
106
107     NumIter(o) = NumIter(o) + 1;
108
109     end
110 end
111 end;
112 t=toc;
113
114 %vytvorenie excel suboru
115 filename = 'VysledkyModif.xlsx';
116 xlswrite(filename,Theta',1);
117 xlswrite(filename,NumIter',2);
118 xlswrite(filename,t,3);
```

Metóda II v prípade nežiadúcich výstupov

```
1 function [Theta,tEnviro]=HyperbolicEnviroModif()
2
3 %Zadefinovanie matic vstupov a vystupov, pricom 1 riadok zodpoveda 1 DMU
4
5 %elektrarne
6 % filename= 'elektrarne_data.xlsx';
7 % sheet = 2;
8 % xRange = 'B4:D95';
9 % yRange = 'E4:E95';
10 % yBadRange = 'F4:G95';
11
12 X=xlsread(filename,sheet,xRange);
13 Y=xlsread(filename,sheet,yRange);
14 Ybad=xlsread(filename,sheet,yBadRange);
15
16 n=length(Y(:,1)); %pocet DMU
17 m=length(X(1,:)); %pocet vstupov
18 s=length(Y(1,:)); %pocet ziaducich vystupov
19 sBad=length(Ybad(1,:)); %pocet neziaducich vystupov
20
21 %Definicia premennych
22 Theta = zeros(1,n);
23 NumIter = zeros(1,n);
```

```

24
25 tic
26 for o = 1:n
27     AddIter = 0;    %pomocna premenna pre zacatie dalsej iteracie; 0 - unsolved, 1 - so
28     xo = X(o,:)';
29     yo = Y(o,:)';
30     yoBad = Ybad(o,:)';
31
32     gy = yo;
33     gyBad = yoBad;
34
35     %% 1. faza - ciel: najst priesečník
36
37     % 1. krok - najdenie riesenia primarnej ulohy
38     cvx_begin quiet
39         cvx_precision default
40         variable lambda(n) nonnegative;
41         variable B;
42         maximize (B);
43         subject to
44             X'*lambda <= xo;
45             -Y'*lambda <= -yo - B*gy;
46             Ybad'*lambda <= yoBad - B*gyBad;
47             %Ybad'*lambda == yoBad - B*gyBad;
48             sum(lambda) == 1;
49     cvx_end
50
51     % ak je beta = 0, utvar je efektívny, koncime algoritmus
52     if abs(cvx_optval) <= 10^(-6)
53         Theta(o) = 1;
54         NumIter(o) = NumIter(o) + 1;
55
56     else
57
58         % 1.krok - najdenie riesenia dualnej ulohy
59         cvx_begin quiet
60             cvx_precision default
61             variables p(s) pBad(sBad) w(m) z;

```



```

62     maximize (yo' * p - xo' * w - yoBad' * pBad - z)
63     subject to
64         Y * p - X * w - Ybad * pBad - ones(n,1)*z <= zeros(n,1);
65         gy' * p + gyBad' * pBad == 1;
66         p >= 0;
67         pBad >= 0;
68         w >= 0;
69     cvx_end
70
71     %% 2. f za - ciel: najst projekciu priesečníka spät na hranicu množiny M
72     while NumIter(o) == 0 || AddIter == 0
73
74         % 2. krok - vypocitat theta a najst priesečník
75         if pBad==0
76             theta = p' * yo / (w' * xo + z);
77         elseif p==0
78             theta = -(z + w' * xo) / (pBad' * yoBad);
79         else
80             theta = (- z - w'*xo + sqrt((z+w'*xo)^2 + 4 *p'*yo * pBad' * yoBad)) / (2 * pBad' * yoBad);
81         end
82
83         % Skonstruovanie noveho smeru
84         gyBad = yoBad - theta * yoBad;
85         gy = yo/theta - yo;
86
87         % Vyriesenie primarnej ulohy
88         cvx_begin quiet
89             cvx_precision default
90             variable lambda(n) nonnegative;
91             variable B;
92             maximize (B);
93             subject to
94                 X'*lambda <= xo;
95                 -Y'*lambda <= -1/theta * yo - B*gy;
96                 Ybad'*lambda <= theta * yoBad - B*gyBad;
97                 %Ybad'*lambda == theta * yoBad - B*gyBad;
98                 sum(lambda) == 1;
99         cvx_end

```

```
100
101     if abs(cvx_optval) <= 10^(-4)
102         Theta(o) = theta;
103         AddIter = 1;
104     else
105         % Vyriesenie dualnej ulohy
106         cvx_begin quiet
107             cvx_precision default
108             variables p(s) pBad(sBad) w(m) z;
109             maximize (1/theta * yo' * p - xo' * w - theta * yoBad' * pBad - z)
110             subject to
111                 Y * p - X * w - Ybad * pBad - ones(n,1)*z <= zeros(n,1);
112                 gy' * p + gyBad' * pBad == 1;
113                 p >= 0;
114                 pBad >= 0;
115                 w >= 0;
116             cvx_end
117
118     end
119
120     NumIter(o) = NumIter(o) + 1;
121
122     end
123 end
124 end;
125 tEnviro = toc;
126
127 %vytvorenie excel suboru
128 filename = 'VysledkyEnviroModif.xlsx';
129 xlswrite(filename,Theta',1);
130 xlswrite(filename,NumIter',2);
131 xlswrite(filename,tEnviro,3);
132 end
```

A.3 Metóda III

Metóda III

```
1 function [ThetaSDP]=HyperbolicSDP()  
2 %Zadefinovanie matic vstupov a vystupov, pricom 1 riadok zodpoveda 1 DMU  
3  
4 %Mehdiloo2015  
5 filename = 'Mehdiloo2015.xlsx';  
6 sheet = 1;  
7 xRange = 'I7:M76';  
8 X=xlsread(filename, sheet, xRange);  
9  
10 sheet = 2;  
11 yRange = 'G5:I74';  
12 Y=xlsread(filename, sheet, yRange);  
13  
14 X=xlsread(filename, sheet, xRange);  
15 Y=xlsread(filename, sheet, yRange);  
16  
17 n=length(Y(:,1)); %pocet DMU  
18 m=length(X(1,:)); %pocet vstupov  
19 s=length(Y(1,:)); %pocet vystupov  
20  
21 %Definicia premennych  
22 ThetaSDP = zeros(1,n);  
23  
24 tic  
25 for o = 1:n  
26     xo = X(o,:)';  
27     yo = Y(o,:)';  
28  
29     cvx_begin quiet sdp  
30         variable lambda(n) nonnegative;  
31         variables theta phi;  
32         minimize theta;  
33         subject to  
34             X'*lambda <= theta*xo;
```

```

35     -Y'*lambda <= -phi*yo;
36     [theta 1; 1 phi] >=0;
37     sum(lambda) == 1;
38     cvx_end
39
40     ThetaSDP(o) = theta;
41
42 end;
43 tSDP = toc;
44
45 %vytvorenie excel suboru
46 filename = 'VysledkySDP.xlsx';
47 xlswrite(filename,ThetaSDP',1);
48 xlswrite(filename,tSDP,2);
49 end

```

Metóda III v prípade nežiadúcich výstupov

```

1 function [ThetaSDP,tSDP]=HyperbolicSDPenviro()
2
3 %Zadefinovanie matic vstupov a vystupov, pricom 1 riadok zodpoveda 1 DMU
4
5 filename= 'elektrarne_data.xlsx';
6 sheet = 1;
7 xRange = 'B4:D95';
8 yRange = 'E4:E95';
9 yBadRange = 'F4:G95';
10
11 X=xlsread(filename,sheet,xRange);
12 Y=xlsread(filename,sheet,yRange);
13 Ybad=xlsread(filename,sheet,yBadRange);
14
15 n=length(Y(:,1)); %pocet DMU
16 m=length(X(1,:)); %pocet vstupov
17 s=length(Y(1,:)); %pocet vystupov
18 sBad=length(Ybad(1,:)); %pocet neziaducich vystupov
19
20 %Definicia premennych
21 ThetaSDP = zeros(1,n);

```

```
22
23 tic
24 for o = 1:n
25     xo = X(o,:)';
26     yo = Y(o,:)';
27     yoBad = Ybad(o,:)';
28
29     cvx_begin quiet sdp
30         variable lambda(n) nonnegative;
31         variables theta phi;
32         minimize theta;
33         subject to
34             X'*lambda <= xo;
35             %Ybad'*lambda == theta*yoBad;
36             Ybad'*lambda <= theta*yoBad;
37             -Y'*lambda <= -phi*yo;
38             [theta 1; 1 phi] >=0;
39             sum(lambda) == 1;
40     cvx_end
41
42     ThetaSDP(o) = theta;
43
44 end;
45 tSDP = toc;
46
47 %vytvorenie excel suboru
48 filename = 'VysledkySDPenviro.xlsx';
49 xlswrite(filename,ThetaSDP',1);
50 xlswrite(filename,tSDP,2);
51
52 end
```

Príloha B

B Vybrané výsledky příkladov

Tabuľka 18: Výsledná postupnosť hodnôt θ^I , β^I a $\tilde{\theta}^I$ neskonvergovaného útvaru *DMU* 14 z dátového súboru Elektrárne pre Metódy I a II - pre model (11)

Typ metódy	Metóda I			Metóda II		
Premenná	θ^I	β^I	$\tilde{\theta}^I$	θ^I	β^I	$\tilde{\theta}^I$
1	0,633	0,4493	0,6381	0,633	0,4493	0,6381
2	0,6968	0	1	0,6893	-1	2,7183
3	0,7628	0	1	0,7585	-1	2,7183
4	0,8442	0	1	0,8247	-1	2,7183
5	0,8809	0	1	0,8767	-1	2,7183
6	0,8844	0	1	0,8848	-1	2,7183
7	0,885	0	1	0,8852	-1	2,7183
8	0,8852	0	1	0,8271	2,6718	0,0691
9	0,8773	3,6716	0,0254	0,8727	-1	2,7183
10	0,8841	0	1	0,8841	-1	2,7183
11	0,8851	0	1	0,8851	-1	2,7183
12	0,8848	3,6712	0,0254	0,885	2,671	0,0692
13	0,8852	0	1	0,8852	-1	2,7183
14	0,8777	3,6716	0,0254	0,8182	2,6718	0,0691
15	0,8843	0	1	0,8712	-1	2,7183
16	0,8851	0	1	0,8836	-1	2,7183
17	0,8849	3,6711	0,0254	0,8851	-1	2,7183
18	0,8852	0	1	0,8852	-1	2,7183
19	0,8804	3,6715	0,0254	0,884	2,6713	0,0692
20	0,8846	0	1	0,8851	-1	2,7183
21	0,8852	0	1	0,8852	2,6705	0,0692
22	0,8324	3,6718	0,0254	0,8208	2,6718	0,0691
23	0,8799	0	1	0,8773	-1	2,7183

Tabuľka 19: Výsledná postupnosť hodnôt θ^I , β^I a $\tilde{\theta}^I$ neskonvergovaného útvaru *DMU* 14 z dátového súboru Elektrárne pre Metódy I a II - pre model (11)

Typ metódy	Metóda I			Metóda II		
Premenná	θ^I	β^I	$\tilde{\theta}^I$	θ^I	β^I	$\tilde{\theta}^I$
24	0,885	0	1	0,8839	-1	2,7183
25	0,8852	0	1	0,885	-1	2,7183
26	0,8181	3,6718	0,0254	0,8852	-1	2,7183
27	0,8674	0	1	0,8549	2,6717	0,0691
28	0,884	0	1	0,8812	-1	2,7183
29	0,8849	0	1	0,8846	-1	2,7183
30	0,8839	0	1	0,8852	-1	2,7183
31	0,885	0	1	0,8612	2,6717	0,0691
32	0,8852	0	1	0,882	-1	2,7183
33	0,8463	3,6717	0,0254	0,885	-1	2,7183
34	0,8666	0	1	0,8852	-1	2,7183
35	0,8803	0	1	0,8729	2,6716	0,0691
36	0,8847	0	1	0,8847	-1	2,7183
37	0,8852	0	1	0,8852	-1	2,7183
38	0,8588	3,6717	0,0254	0,8756	2,6716	0,0691
39	0,8808	0	1	0,8848	-1	2,7183
40	0,8845	0	1	0,8852	-1	2,7183
41	0,8852	0	1	0,8345	2,6718	0,0691
42	0,8723	3,6716	0,0254	0,8744	-1	2,7183
43	0,8837	0	1	0,8843	-1	2,7183
44	0,885	0	1	0,8851	-1	2,7183
45	0,8852	0	1	0,8851	2,6708	0,0692
46	0,8101	3,6718	0,0254	0,8849	-1	2,7183
47	0,8672	0	1	0,8852	-1	2,7183
48	0,884	0	1	0,8058	2,6718	0,0691
49	0,8851	0	1	0,8634	-1	2,7183
50	0,8851	3,6515	0,026	0,883	-1	2,7183